DATASTAX

OpsCenter 5.1 User Guide

Documentation

${ds.localized.time}

Contents

About OpsCenter

DataStax OpsCenter is a visual management and monitoring solution for Apache Cassandra and DataStax Enterprise.

DataStax OpsCenter is a visual management and monitoring solution for Apache Cassandra and 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, and monitoring the health of your clusters on the dashboard.

OpsCenter is available for both open source Cassandra and DataStax Enterprise. Different features are available depending on licensing.

Key features

DataStax OpsCenter offers features to more easily manage both DataStax Enterprise and Apache Cassandra clusters.

The key features of OpsCenter include:

Dashboard	An Overview that shows any alerts and condenses the dashboards of multiple clusters. A Dashboard that displays an overview of commonly monitored performance metrics. Ability to add and edit graphs in the dashboard.
Configuration and administration	Basic cluster configuration. Administration tasks, such as adding a cluster, using simple point-and-click actions. Visual creation of clusters. Multiple cluster management from a single OpsCenter instance using agents. Multiple node management. Downloadable PDF cluster report.
Enterprise-only functionality	Enterprise functionality in OpsCenter is only enabled on DataStax Enterprise clusters. Monitoring capabilities of DSE In-Memory tables. View the Spark console. Automatic failover from the primary OpsCenter to a backup OpsCenter instance. Security, with the ability to define user roles. DSE Management Services • Backup Service - allows automatic or manual backup and restore of data in clusters. • Repair Service - continuously runs and performs repair operations across a DSE cluster. • Capacity Service - understand cluster performance trends at a glance and plan for future capacity with forecasting.

- Best Practice Service - schedule pre-defined best practice rules that check various properties of clusters and environments.

Alerts

- Built-in external notification capabilities.
- Alert warnings of impending issues.
- Metrics are collected every minute from Cassandra, Analytics, and Search nodes, and stored in a keyspace created by OpsCenter.

Manage multiple nodes simultaneously for certain bulk operations.

Rebalance data across a cluster when new nodes are added or removed.

Generate a diagnostics tarball to send to support for further troubleshooting.

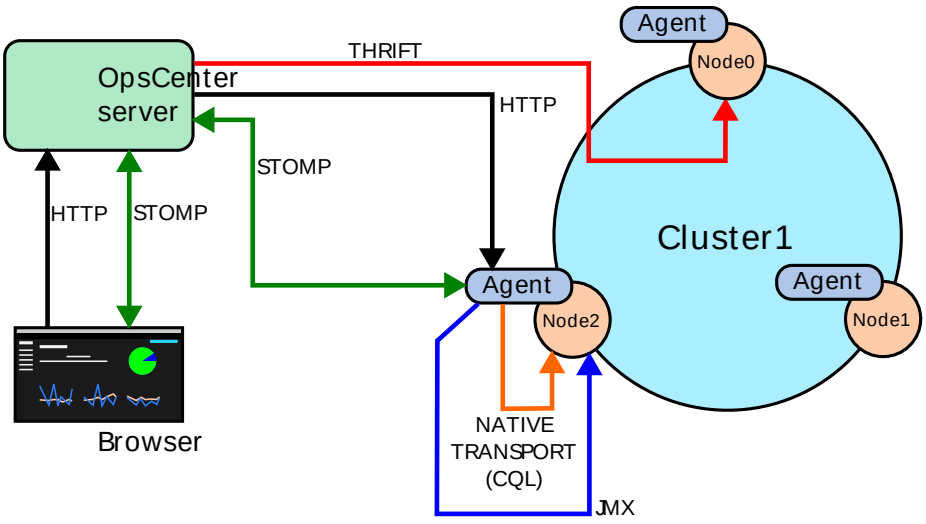
Hadoop Job Tracker integration.

View historical metrics more than one week in the past.

OpsCenter Architecture Overview

Basic architecture diagram and explanation.

The DataStax agents are installed on the Cassandra and DSE nodes. The agents use Java Management Extensions (JMX) to monitor and manage each node. Cassandra 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.



Installation

We recommend the following minimum hardware requirements for the machine on which OpsCenter will run:

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

You can access the web interface of OpsCenter from any computer with network access to the OpsCenter machine. We recommend using a recent version of one of the major web browsers.

Installing OpsCenter

Installation options include instructions for standalone installers, packages, and tarball installs.

Installing OpsCenter with the standalone installer

Installs the DataStax OpsCenter web application with a standalone GUI installer on Mac OS X, or on any Linux distribution with a standalone command line installer.

About this task

The standalone installer installs OpsCenter using either a GUI installer on Mac OS X, or a command line installer on any Linux distribution. The DataStax agent is not installed with the standalone installer. The agent is installed when DSE or Cassandra is installed, or you can manually install the agents. For more information, see Installing DataStax agents.

Before you begin

- Root or sudo access when installing as a system service, and if installing missing system dependencies.
- Latest version of Oracle Java SE Runtime Environment 7, **not** OpenJDK. See Installing the Oracle JRE.

Procedure

1. Download the installer for your operating system from the DataStax OpsCenter download page:
   - Mac OS X
   - Linux
2. Launch the installer:
   - (Mac OSX) Double-click the downloaded .dmg file.
   - (Linux) Follow these steps:
     1. From the directory where you downloaded the install file, change the permission to executable:

        ```
        chmod +x DataStaxOpsCenter-5.1.0.version number-linux-64-installer.run
        ```
     2. Start the installation:

        ```
        ./DataStaxOpsCenter-5.1.0.version number-64-linux-installer.run
        ```
3. Follow the instructions in the install wizard.
   The installation completes. If you installed as a service and selected **Launch OpsCenter Web Interface** in the wizard, OpsCenter opens in your default browser. Otherwise, continue with the next step.
4. If you chose not to run as a service, start OpsCenter to run in the background:

```
$ bin/opscenter
```

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

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

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

Uninstalling OpsCenter

Uninstalls OpsCenter if the install was done with the standalone installer.

About this task

Follow these steps to uninstall the OpsCenter application that was installed using the standalone installer on either Mac OS X or Linux operating systems.

Procedure

1. Go to the OpsCenter installation directory.
2. Launch the uninstaller:
   - **Linux:** `$ ./uninstall ## Run the uninstaller as root or sudo if needed`
   - **Mac OS X:** Double-click **uninstaller**.

   A dialog prompts you to confirm whether you want to do a full uninstall including config files and data.
3. Confirm the type of uninstall:
   - To confirm fully uninstalling the application, click **Yes**.
   - To uninstall only the OpsCenter application and retain data and config files, click **No**.

   The uninstaller removes the OpsCenter application.

Other install methods

Alternative install options to the standalone installer include RPM and Debian packages or a tarball for Linux distributions.

Install OpsCenter using RPM or Debian packages (YUM or APT repositories) or a binary tarball.

Installing the OpsCenter RPM package

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

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 DataStax OpsCenter – Supported Platforms.

Before you begin

- Yum package management utility.
- For CentOS or RHEL 5.x, EPEL.
- Python 2.6+

About this task

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

Procedure

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

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

2. In this file, add the repository for OpsCenter.

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

3. Install the OpsCenter package.

```
$ sudo yum install opscenter
```

For most users, the out-of-box configuration should work just fine, but if you need to you can configure OpsCenter differently.

4. Start OpsCenter:

```
$ sudo service opscenterd start
```

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

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

6. Next you can add an existing cluster or provision a new one.

Installing the OpsCenter deb package

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

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 DataStax OpsCenter – Supported Platforms.

Before you begin

- APT Package Manager is installed.
- Python 2.6+

About this task

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 machine reboots, OpsCenter restarts automatically.

Procedure

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

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

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

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

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

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

For most users, the out-of-box configuration should work just fine, but if you need to you can configure OpsCenter differently.

4. Start OpsCenter:

```
$ sudo service opscenterd start
```

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

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

6. Next you can add an existing cluster or provision a new one.

Installing OpsCenter on Mac OS X or any Linux distributions

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

About this task

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 DataStax OpsCenter – Supported Platforms.

Before you begin

- Python 2.6+

Procedure

1. Download the tarball distribution of OpsCenter.

```
$ curl -L http://downloads.datastax.com/community/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/
```

5. Next you can add an existing cluster or provision a new one.

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.

About this task

After you've added the cluster to OpsCenter, you'll see the status of the agents in the Dashboard.

OpsCenter attempts to automatically install the agents on the nodes in the newly added cluster. If this fails, you might need to manually install and configure the agents on each node.

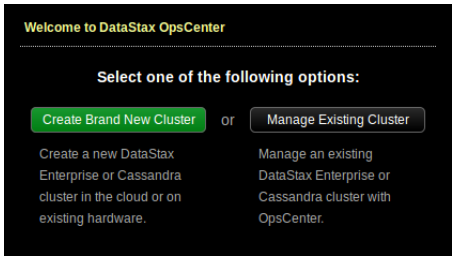
Before you begin

- 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, or accepted the default SSH port (22) for node-agent communication.

Procedure

1. Open a browser window and go to the OpsCenter URL at `http://<opscenter_host>:8888` where <*opscenter_host*> is the IP or hostname of the OpsCenter machine.

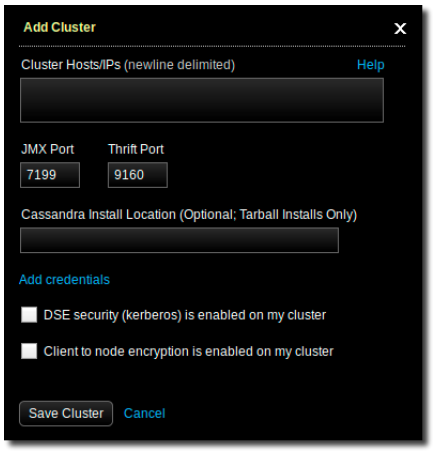
   When you start OpsCenter for the first time, you are prompted to connect to a cluster:



2. Open a browser window and go to the OpsCenter URL at `http://opscenter_host:8888/` where *opscenter_host* is the IP or hostname of the OpsCenter machine.

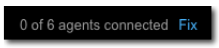
   ```
   http://1.2.3.4:8888/
   ```

3. In **Add Cluster**, enter the Hostnames or IP addresses of two or three nodes in the cluster, set the JMX and Thrift ports, and click **Save Cluster**.

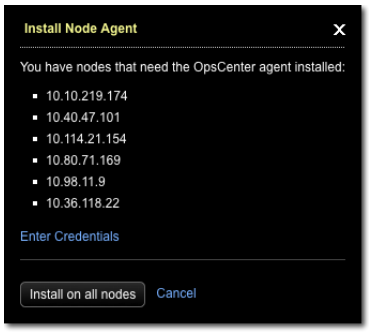


   After OpsCenter connects to the cluster, a **Fix** link appears near the top of the Dashboard.

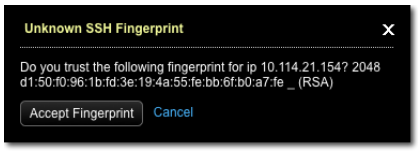
4. Click the **Fix** link to start installing the agents.



5. In **Install Node Agent**, click **Enter Credentials**.



6. In **Node SSH Credentials**, enter a `username` that has root privileges or sudo access to all of the nodes in your cluster, plus any other required credentials, and click **Done**.
7. In the **Install Node Agent** dialog, click **Install on all nodes**.
8. If prompted, click **Accept Fingerprint** to add a node to the known hosts for OpsCenter.



Results

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

If you are unable to install the agents through the OpsCenter UI, follow the instructions to manually install the agents.

Manually deploying agents - tarball

Install agents on nodes running Cassandra or DataStax Enterprise clusters.

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:

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

Before you begin

- The Cassandra or 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).

Procedure

1. Download the DataStax agent tarball, expand and unarchive it.

   ```
   $ curl -L http://downloads.datastax.com/community/datastax-agent-<version-number>.tar.gz | tar xz
   ```
2. Change into the agent directory.

   ```
   $ cd datastax-agent-<version-number>
   ```
3. In `address.yaml` set stomp_interface to the IP address that OpsCenter is using. (You may have to create the file.)

   ```
   $ echo "stomp_interface: <reachable_opscenterd_ip>" >> ./conf/address.yaml
   ```
4. If SSL communication is enabled in opscenterd.conf, use SSL in address.yaml.

   ```
   $ 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 - rpm

Installs agents on Linux nodes using Yum packages.

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:

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

Before you begin

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

Procedure

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.

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

2. Install the DataStax agent.

```
# yum install datastax-agent
```

3. In `address.yaml` set stomp_interface to the IP address that OpsCenter is using. You might have to create the file.

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

4. 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
```

5. Start the DataStax agent.

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

Manually deploying agents - deb

Installs agents on Linux nodes using APT packages.

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:

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

Before you begin

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

Procedure

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

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

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

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

   Note: If you have any difficulty adding the key, use HTTP instead of HTTPS.

3. Install the DataStax agent.

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

4. In `address.yaml` set stomp_interface to the IP address that OpsCenter is using. You might have to create the 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
   ```

Automatic installation of DataStax agents

When you install DataStax Enterprise 4.0 or later, the DataStax agent is automatically installed on the nodes of a cluster.

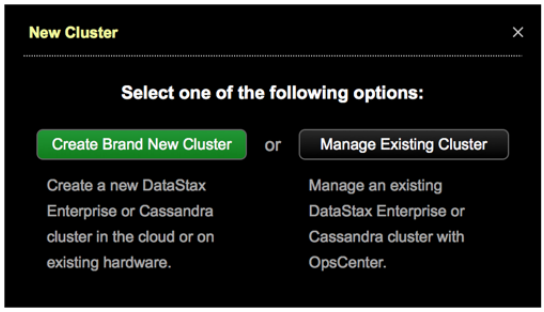
About this task

After installing, configuring, and running a DSE cluster and OpsCenter, you connect to OpsCenter in a web browser and are automatically asked whether to provision a new cluster or connect to an existing cluster. In either case, OpsCenter connects to nodes with agents already deployed (DSE 4.0 and greater), or deploys agents to nodes and then connects.

Procedure

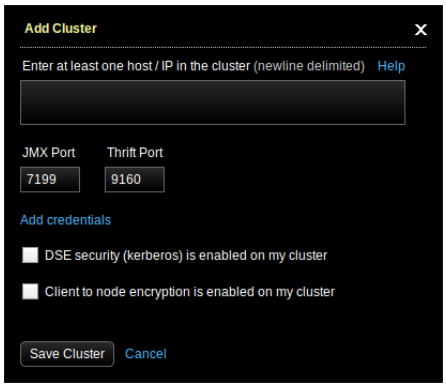
1. Install DSE 4.0 or greater.
2. Install OpsCenter.
3. Start your cluster and the OpsCenter daemon.
4. Open the URL for OpsCenter in a web browser.
   `http://localhost:8888/`

   The New Cluster dialog prompts you to select a cluster option.



5. Select **Manage Existing Cluster**.

   The Add Cluster dialog is displayed.



6. Add the hostnames or IP addresses of the nodes in the cluster. For best results, use private IP addresses.
7. Click **Save Cluster**.

Setting permissions to run the agent as a different user

Describes necessary directory and file permissions if you need to run the agent as a different user than the default user for DSE or DSC, which is cassandra.

About this task

Running the agent as the same user running DSE or DSC is highly recommended because directory and file permissions do not need to be set manually. By default, the DataStax agent when installed by deb or rpm packages runs as the same user as DSE and DSC, which is cassandra. Those who install the agent from a tarball are responsible for manually configuring running the agent and DSE or DSC as different users.

Before you begin

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

Table 1: Directory and File Permissions

Feature functionality	Permissions required
General agent functionality	Read permission to `cassandra.yaml`
Configuring a cluster	Read/write permissions to configuration directories and files
Backup and restore	• Read/write permissions to configuration directories and files • Read/write permissions to Cassandra data directories

Feature functionality	Permissions required
	Note: A umask must also be set to accommodate group permissions for new tables and data. • If commitlog archiving is enabled, the DSE process must also have permissions to run the agent's archive script and write permissions to the configured backup directory.

Table 2: Directory and File Locations

Directory/File	Location
`cassandra.yaml`	See Configuration directories and files below.
Configuration directories and files	• DSE package: `/etc/dse` • DSE tarball: `<install location>/conf` • Cassandra package: `/etc/cassandra` • Cassandra tarball: `<install location>/conf`
Data directories	Default: `/var/lib/cassandra` **Note:** Location is user-configurable; set in `cassandra.yaml`.
Commitlog archiving script	• Agent package install: `/usr/share/datastax-agent/bin/archive_commitlog.sh` • Agent tarball install: `<install location>/bin/archive_commitlog.sh`

About this task

To set up the umask:

Procedure

1. Open a terminal.
2. To give read/write permissions for new tables and data, edit the appropriate shell file for the DSE or DSC environment:

File	Location
`dse-env.sh`	• `/etc/dse/` • *`<install location>`*`/conf/`
`cassandra-env.sh`	• `/etc/cassandra` • *`<install location>`*`/conf`

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

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

```
umask 002
```

Configuring JAVA_HOME for DataStax agents

DataStax agents do not pick up the environment variables of the currently logged-in user by default.

About this task

For example, if Java is not in the machine's PATH, you may notice errors in the agent log on start-up:

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

Procedure

- 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 may need. For example:

  ```
  JAVA_HOME = /usr/lib/jvm/java-7-oracle
  ```

OpsCenter and DataStax agent ports

Default port numbers used by OpsCenter and the DataStax Agents.

Port	Description
OpsCenter ports	
8888	OpsCenter website. The `opscenterd` daemon listens on this port for HTTP requests coming directly from the browser. Configurable in opscenterd.conf.
50031	OpsCenter HTTP proxy for Job Tracker. The `opscenterd` daemon listens on this port for incoming HTTP requests from the browser when viewing the Hadoop Job Tracker page directly. Configured in opscenterd.conf. (DataStax Enterprise only)
61620	OpsCenter monitoring port. The `opscenterd` daemon listens on this port for TCP traffic coming from the agent. Configured in opscenterd.conf.
DataStax agents ports (on the monitored nodes)	
7199	JMX monitoring port. Each agent opens a JMX connection to its local node (the Cassandra or 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. This port is set in the cluster-specific configuration file.
8012	Hadoop Job Tracker client port. The Job Tracker listens on this port for job submissions and communications from task trackers; allows traffic from each Analytics node in a DataStax Enterprise cluster.
8012	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. (DataStax Enterprise only)
8012	Hadoop Task Tracker website port. Each Task Tracker listens on this port for HTTP requests coming directly from the browser and not proxied by the opscenterd daemon. (DataStax Enterprise only)
9042	The native transport port for the cluster configured in `native_transport_port` in cassandra.yaml. Set in `cql_port`.
61621	DataStax agent port. The agents listen on this port for SSL traffic initiated by OpsCenter. cluster-specific configuration file.
22	SSH port. Configurable in opscenterd.conf.

Port	Description
Solr Port and Demo applications port	
8983	Solr Port and Demo applications port.

cassandra.yaml

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/cassandra/cassandra.yaml`	X	X		
`install_location``/conf/cassandra.yaml`			X	X

Installation and configuration locations

File locations can vary based on the type of install.

Debian and Ubuntu Package install locations

File locations for Debian and Ubuntu package installs.

Directory	Description
`/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 start-up script

CentOS, OEL, and RHEL Package install locations

File locations for RHEL-based package installs.

Directory	Location
`/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

Binary Tarball distribution install locations

File locations for binary-based installs.

Directory	Location
`/agent`	Agent installation files
`/bin`	Startup and configuration binaries
`/content`	Web application files
`/conf`	Configuration files
`/doc`	License files
`/lib` and `/src`	Library files
`/log`	OpsCenter log files
`/ssl`	SSL files for OpsCenter to agent communications

Starting, stopping, and restarting OpsCenter

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

About this task

Commands are available for starting, stopping, and restarting OpsCenter for each type of installation. Packaged installations include startup scripts for running OpsCenter as a service. The available `service opscenterd` options are:

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

Procedure

The following list shows start, stop, and restart instructions for the supported platforms:

- Start OpsCenter:
  - Package installations: `sudo service opscenterd start`
  - Tarball installations: *`install_location`*`/bin/opscenter` (Use -f to start in the foreground.)
  - Windows installations: Start the OpsCenter Service from the Control Panel.

    Note: By default, DataStax Enterprise services on Windows start automatically.
- Stop OpsCenter:
  - Package installations: `sudo service opscenterd stop`
  - Tarball installations: Find the OpsCenter Java process ID (PID) and kill the process using its PID number:

    ```
    $ ps -ef | grep opscenter
    $ sudo kill pid
    ```
  - Windows installations: Stop the OpsCenter Service from the Control Panel.
- Restart OpsCenter:
  - Package installations:

    ```
    $ sudo service opscenterd restart
    ```
  - Tarball installations:

    Find the OpsCenter process ID (*pid*), kill the process using its PID number, and start OpsCenter:

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

install_location`/bin/opscenter` (Use -f to start in the foreground.)

- Windows installations:

  Restart the OpsCenter Service from the Control Panel.

Starting and restarting DataStax agents

Commands for starting and restarting DataStax agents for each type of installation.

Procedure

- To start the DataStax agent:
  - Packaged installs: The DataStax agent starts automatically.
  - Tarball installs: `$` *install_location* `bin/datastax-agent` (Use `-f` to start in the foreground.)
  - Windows installs: Start the DataStax Agent Service from the Control Panel.
- To restart the DataStax agent:
  - Packaged installs: `$ sudo service datastax-agent restart`
  - Tarball installs: Find the DataStax agent Java process ID (PID), kill the process using its PID number, and start the DataStax agent:

    ```
    $ ps -ef | grep datastax-agent
    $ sudo kill <pid>
    ```

    `$ bin/datastax-agent` (Use `-f` to start in the foreground.)
  - Windows installs: Restart the DataStax Agent Service from the Control Panel.

Upgrading OpsCenter

See the Upgrade Guide for detailed instructions on upgrading OpsCenter.

Configuration

Configuring role-based security

OpsCenter allows enabling user authentication, adding users, and defining custom roles.

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

About user access roles

OpsCenter provides the ability to define custom, fine-grained access roles.

DataStax Enterprise customers have the ability to define custom, fine-grained access roles for these users. OpsCenter can be configured to require users to log in.

Note: If your organization is using DataStax Community, all users who log in to OpsCenter have admin rights.

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 when authentication is enabled. Only users with the admin role can manage users and roles, create new clusters, or manually update definition files.

User-defined role privileges

User-defined roles can only be defined by admin users. The permissions of the user-defined roles are per cluster. Any functionality in OpsCenter that the user does not have permission for appear disabled when that user is logged in.

Creating a new cluster does not automatically add permissions to any existing user-defined roles.

Enabling authentication in OpsCenter

Instructions for enabling role-based authentication in OpsCenter.

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

About this task

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 recommend the first time you log in.

If you enable authentication, we strongly recommend enabling SSL communication between OpsCenter and the agents.

Procedure

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

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

   ```
   [authentication]
   enabled=True
   ```
2. Restart OpsCenter:
   - Package installations:

     ```
     $ sudo service opscenterd restart
     ```
   - Tarball installations:

     Find the OpsCenter process ID (*pid*), kill the process using its PID number, and start OpsCenter:

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

     install_location`/bin/opscenter` (Use -f to start in the foreground.)
   - Windows installations:

     Restart the OpsCenter Service from the Control Panel.
3. Open the OpsCenter UI in a browser.

   `http://localhost:8888`
4. Enter the default username of `admin` and the password `admin`.

Managing users and roles

Manage users and role permissions visually through the OpsCenter UI.

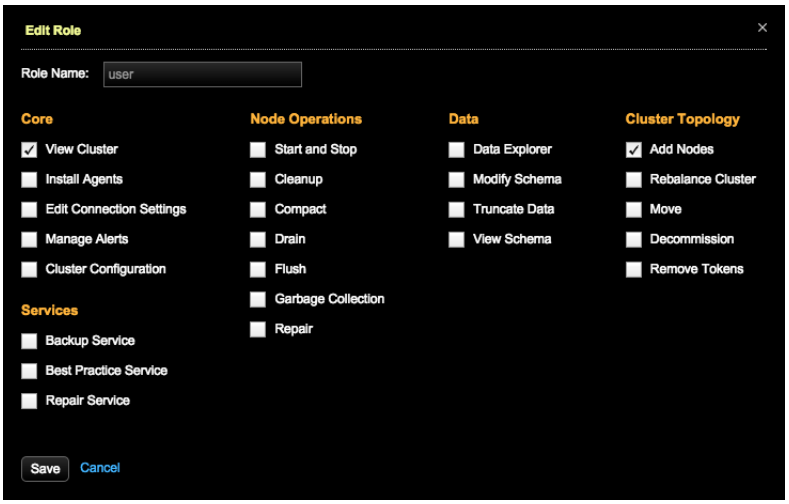
About this task

Follow these instructions to manage users and roles in OpsCenter.

Procedure

1. Log in to OpsCenter as an admin. Click **Settings** > **Users & Roles**.
   The Users and Roles dialog appears.
2. To create a new user:
   a) Click **Create User**.
   b) Enter the username, password, and role for the user, and 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 **Submit**.
   c) To change the user's role, select the new role from the **Roles** list and click **Submit**.
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.
   b) To edit an already-existing role, click the **Edit** icon.

The Edit Role dialog appears.



c) Select the options the user role has permissions for and click **Save**.

6. To create a role:
   a) Click **Create Role**.
   b) Enter the name of the role under **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 **Submit**.
7. To delete a role:
   a) Select the role you want to delete and click the **Delete** icon.

Migrating users to the new password database

Migrate user authentication in versions of OpsCenter prior to 5.0 to 5.0+ using the provided script.

About this task

Use the provided migration script when migrating user authentication between versions of OpsCenter prior to 5.1.

Note: The migration script is only required when migrating from OpsCenter versions pre-5.0 to 5.0+. For upgrading user authentication with a tarball install of OpsCenter 5.1 and higher, copy the `passwd.db` file as instructed in the Upgrade Guide.

The `migrate_password_file.py` script reads the users and passwords from the old password file and inserts them into the new password database.

When OpsCenter first starts up with authentication enabled, it creates an `admin` role. Before you migrate, you must create a `user` role. Existing users in the old password file are copied to one of these roles based on their previous role.

Before you begin

To migrate your users and passwords, you must have file system access to the old password file.

Procedure

1. Enable authentication in OpsCenter.
2. Start OpsCenter so that the new password database is created.

3. Create a role with the name `user` in OpsCenter.
4. In a terminal, run the `migrate_password_file.py` script. The script takes two arguments: the path to the old password file, and the path to the new password database.

   The default location of the `migrate_password_file.py` script is `/usr/share/opscenter/bin/` for package installs and *`install_location`*`/bin` for tarball installs.

   The default location of the old password file is `/etc/opscenter/.passwd` for package installs and *`install_location`*`/conf/.passwd` for tarball installs.

   The default location of the new password database is `/etc/opscenter/passwd.db` for package installs and *`install_location`*`/passwd.db` for tarball installs.

   For package installs:

   ```
   $ cd /usr/share/opscenter/bin
   $ sudo ./migrate_password_file.py /etc/opscenter/.passwd /etc/opscenter/
   passwd.db
   ```

   For tarball installs:

   ```
   $ cd install_location/bin
   $ sudo ./migrate_password_file.py install_location/
   conf/.passwd install_location/passwd.db
   ```
5. Restart OpsCenter:

   - Package installations:

     ```
     $ sudo service opscenterd restart
     ```
   - Tarball installations:

     Find the OpsCenter process ID (*pid*), kill the process using its PID number, and start OpsCenter:

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

     `install_location``/bin/opscenter` (Use -f to start in the foreground.)
   - Windows installations:

     Restart the OpsCenter Service from the Control Panel.

Changing the location of the password database

Change the default location of the password database used for authentication.

About this task

The password database is created when authentication is enabled. The default location of the password database is `/etc/opscenter/passwd.db` for package installs and *`install_location`*`/passwd.db` for tarball installs. You can change the location of the password database in the opscenterd.conf file.

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
install_location/conf/opscenterd.conf			X	X

Procedure

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

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

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

   Note: 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:
   - Package installations:

     ```
     $ sudo service opscenterd restart
     ```
   - Tarball installations:

     Find the OpsCenter process ID (*pid*), kill the process using its PID number, and start OpsCenter:

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

     install_location`/bin/opscenter` (Use -f to start in the foreground.)
   - Windows installations:

     Restart the OpsCenter Service from the Control Panel.

Configuring SSL

OpsCenter uses Secure Socket Layer (SSL) to encrypt the communication protocol and authenticate traffic between DataStax agents and the main OpsCenter daemon. SSL is disabled by default. Enabling SSL is recommended.

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 - package installations

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

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

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`

Before you begin

- The Python interface for the OpenSSL library (`pyOpenSSL`). With package installs (rpm or deb) of OpsCenter, the `python-openssl` package is installed as a dependency. However, this is not the case with CentOS 5.x installs.

Procedure

1. Ensuring that a version of `pyOpenSSL` compatible with the installed version of `libssl` is a requirement for any secure communications in OpsCenter.
   - If you are using `libssl` 1.x, ensure that `pyOpenSSL` 0.10+ is installed and compiled properly.

   a) Optional: Determine the version of `pyOpenSSL` installed.

   ```
   $ python -c "import OpenSSL; print OpenSSL.__version__"
   ```

   b) Optional: Manually install `pyOpenSSL`.

   ```
   $ sudo easy_install pyOpenSSL
   ```

2. Run the OpsCenter `setup.py` script:

   ```
   $ sudo /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:

   `/var/lib/opscenter`

3. Open opscenterd.conf in an editor and add the following lines to enable SSL:

   ```
   $ sudo vi /etc/opscenter/opscenterd.conf
   ```

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

4. Restart the OpsCenter daemon.

If you want 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 (see steps below).

5. Reconfigure the agents on all nodes.

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

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

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

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

```
$ ssh user@node
```

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

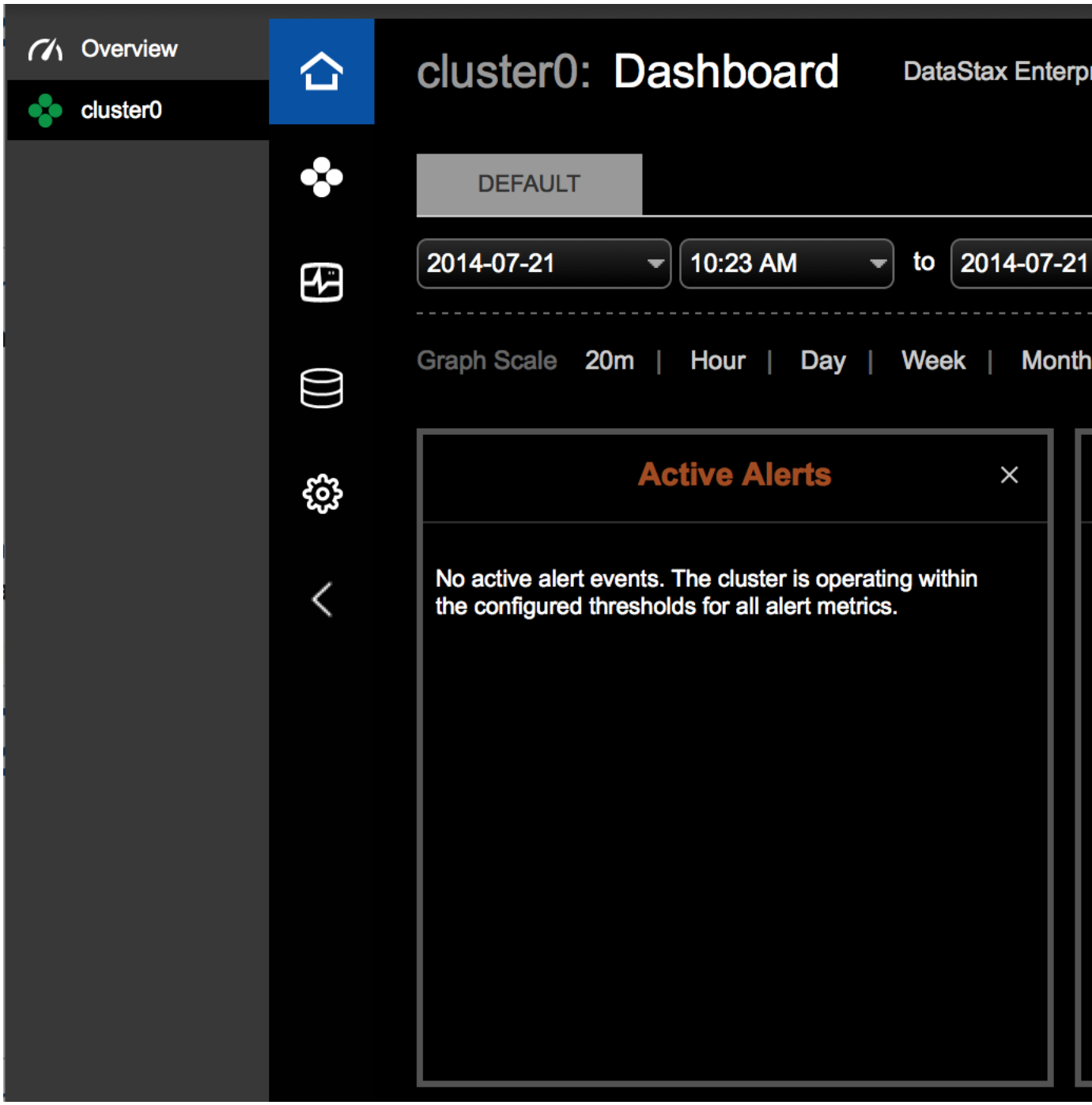
```
$ sudo vi /var/lib/datastax-agent/conf/address.yaml

 use_ssl: 1
```

d) Restart the agent.

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

If you do not want to manually edit all of the node configuration files, follow the agent installation procedure.

6. After `opscenterd` and all agents have been configured and restarted, verify proper connection through the dashboard.



What to do next

If you are upgrading an existing cluster to SSL, see adding an existing cluster for instructions on generating an OpenSSL certificate for the cluster to be reconfigured for SSL communications with OpsCenter.

Enabling SSL - tarball installations

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

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

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`

Procedure

1. Ensuring that a version of pyOpenSSL compatible with the installed version of `libssl` is a requirement for any secure communications in OpsCenter.
   - If you are using `libssl` 1.x, ensure that `pyOpenSSL` 0.10+ is installed and compiled properly.
   a) Optional: Determine the version of `pyOpenSSL` installed.

   ```
   $ python -c "import OpenSSL; print OpenSSL.__version__"
   ```
   b) Optional: Manually install `pyOpenSSL`.

   ```
   $ sudo easy_install pyOpenSSL
   ```
2. Run the OpsCenter `setup.py` script:

   ```
   $ sudo 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`
3. Open opscenterd.conf in an editor and add two lines to enable SSL.

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

   ```
   [agents]
   use_ssl = true
   ```
4. Restart the OpsCenter daemon.

If you want to connect to a cluster in which agents have already been deployed, you can log in to each of the nodes and reconfigure the address.yaml file (see steps below).

5. Reconfigure the agents on all nodes.
   a) Copy *`install_location`*`/ssl/agentKeyStore` from the OpsCenter machine to `/var/lib/datastax-agent/ssl/agentKeyStore` for package installations, or *`agent_install_location`*`/ssl/agentKeyStore` on each node in the cluster.

   ```
   $ 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.

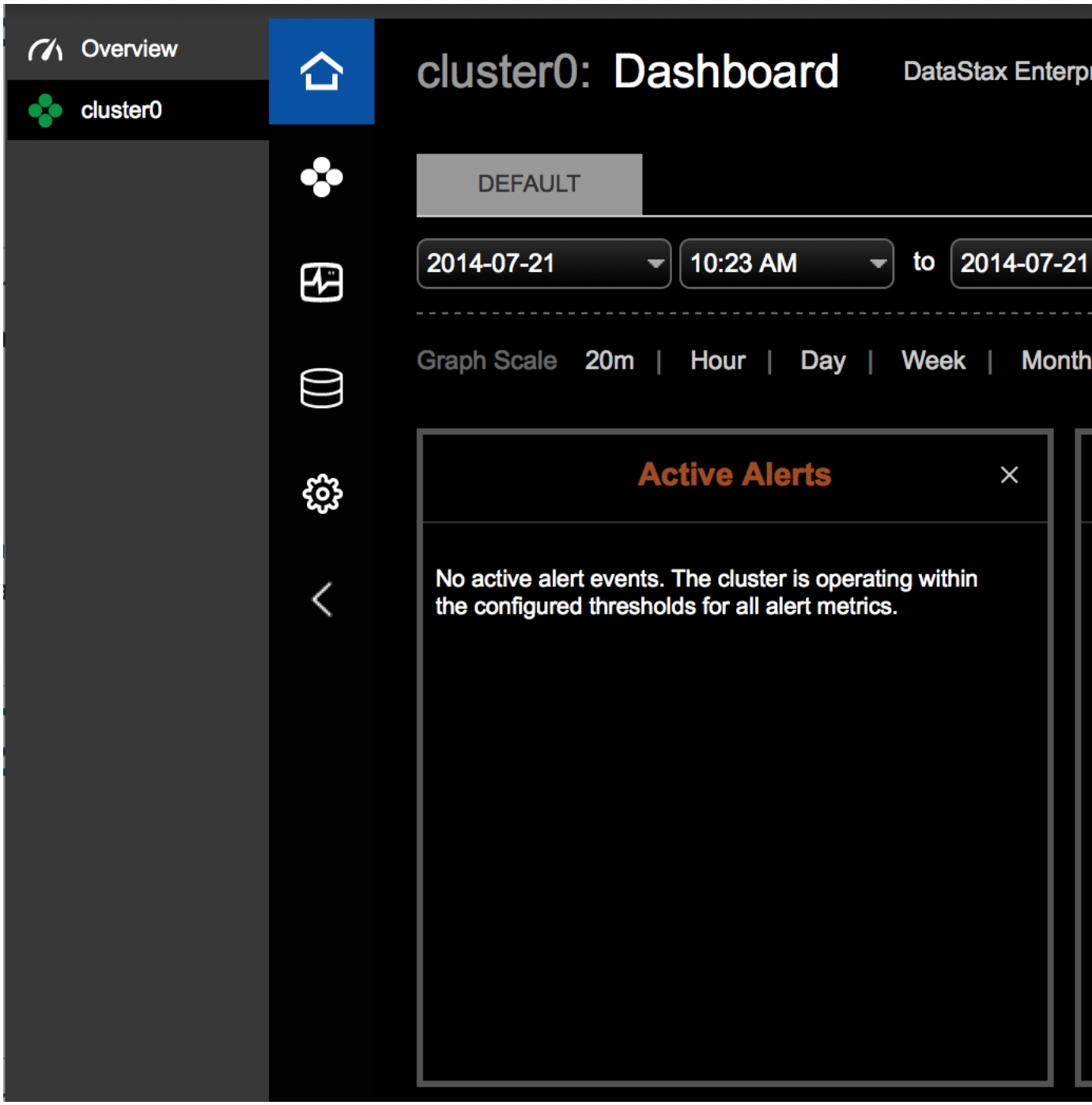
```
$ sudo vi install_location/conf/address.yaml

 use_ssl: 1
```

d) Restart the agent.

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

If you do not want to manually edit all of the node configuration files, follow the agent installation procedure.

6. After `opscenterd` and all agents have been configured and restarted, verify proper connection through the dashboard.



What to do next

If you are upgrading an existing cluster to SSL, see adding an existing cluster for instructions on generating an OpenSSL certificate for the cluster to be reconfigured for SSL communications with OpsCenter.

Disabling SSL - package installations

To disable SSL for package installations, modify the OpsCenter configuration file and restart OpsCenter.

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

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`

About this task

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

Procedure

1. Open opscenterd.conf in an editor and add two lines to enable SSL.

```
$ sudo vi /etc/opscenter/opscenterd.conf
```

```
[agents]
use_ssl = false
```

2. Restart the OpsCenter daemon.
3. Reconfigure the agents.
   a) Log into each node in the cluster using `ssh`.

```
$ ssh user@node
```

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

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

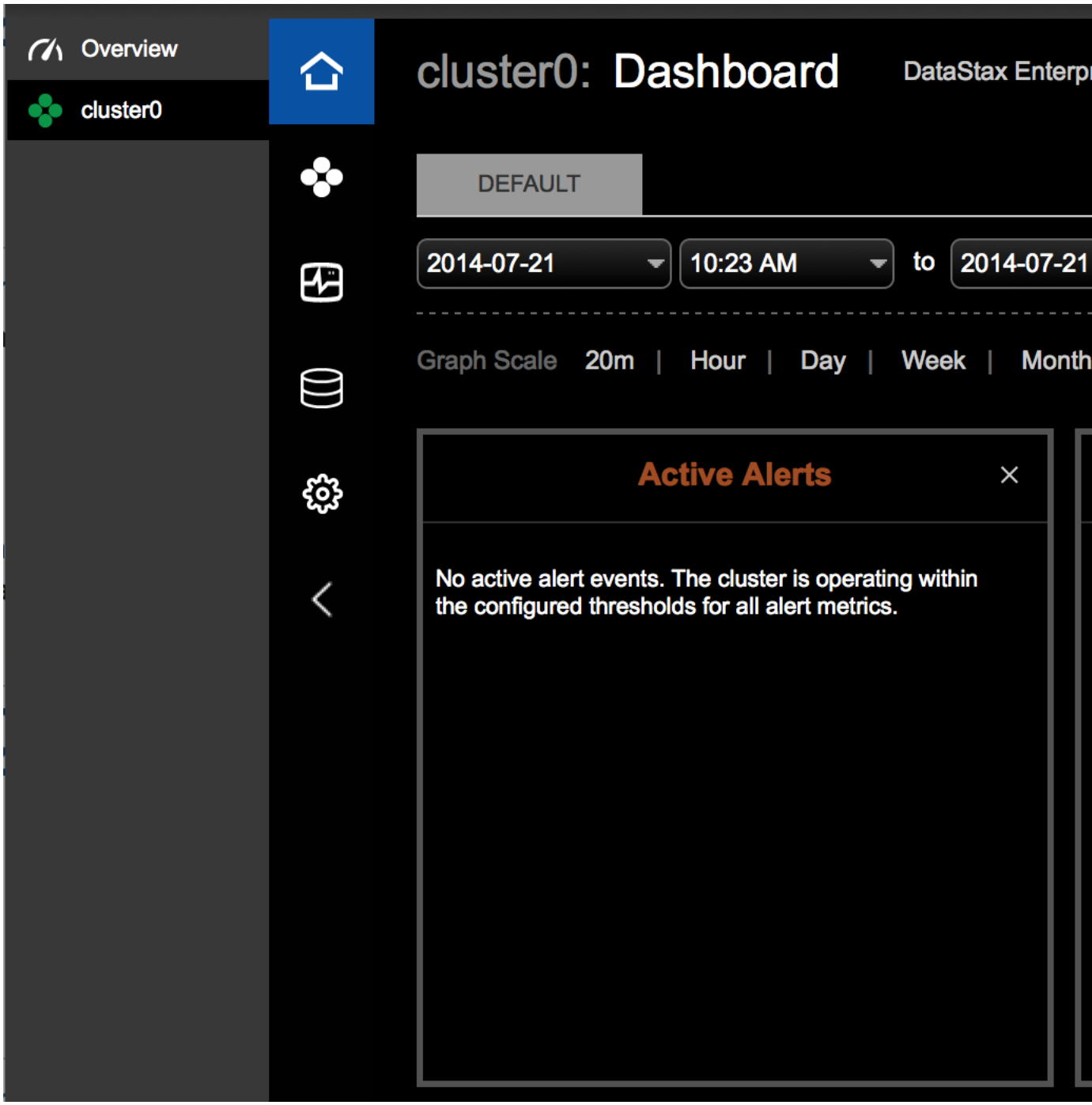
```
use_ssl: 0
```

   c) Restart the agent.

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

If you do not want to manually edit all of the node configuration files, follow the agent installation procedure.

4. After `opscenterd` and all agents have been configured and restarted, verify proper connection through the dashboard.



Disabling SSL - tarball installations

To disable SSL for tarball installations, modify the OpsCenter configuration file and restart OpsCenter.

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

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`

About this task

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

Procedure

1. Open opscenterd.conf in an editor and add two lines to enable SSL.

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

   ```
   [agents]
   use_ssl = false
   ```

2. Restart the OpsCenter daemon.
3. Reconfigure the agents.
   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.

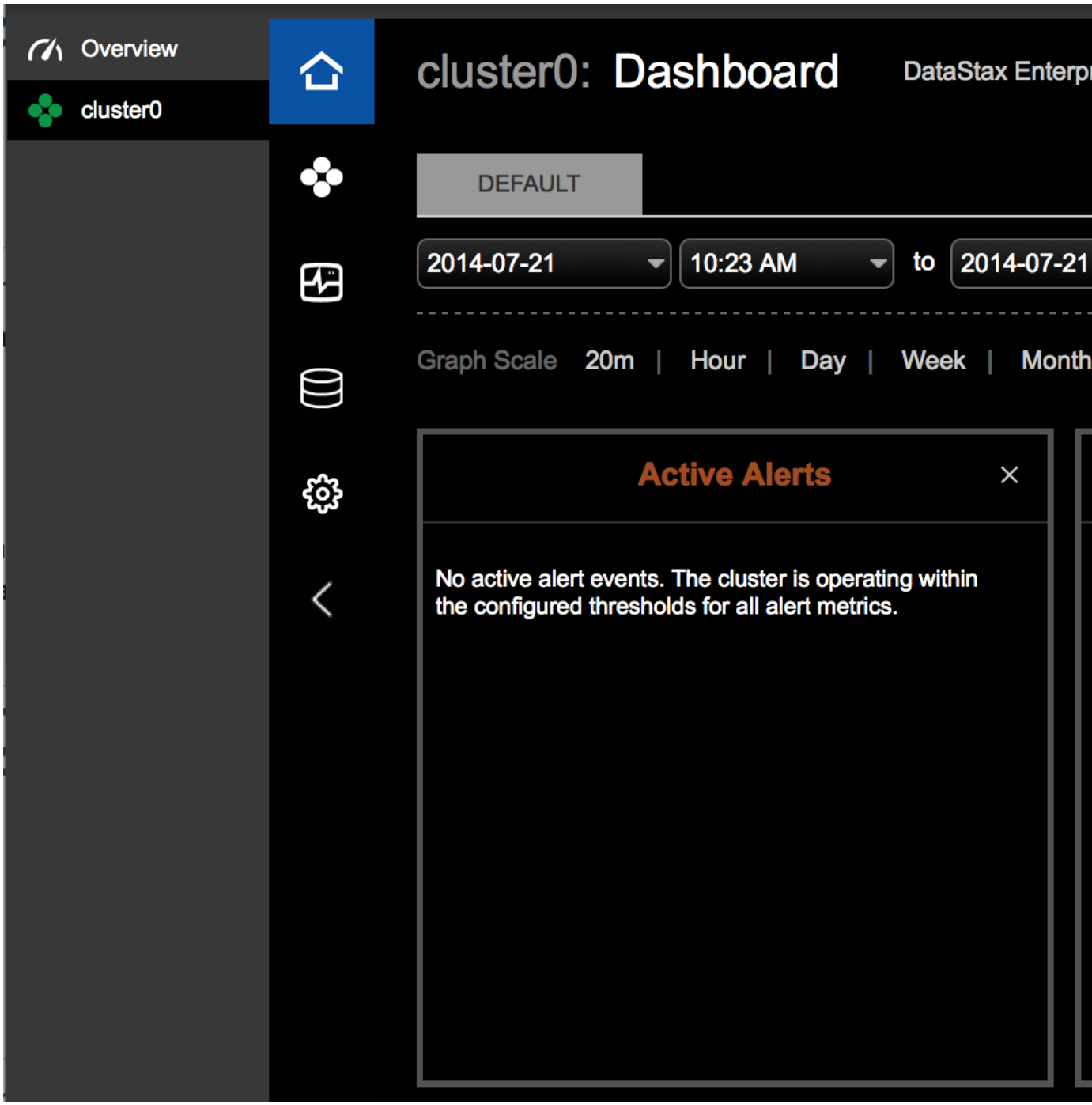
      ```
      $ sudo vi install_location/conf/address.yaml
      ```

      ```
      use_ssl: 0
      ```

   c) Restart the agent.

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

   If you do not want to manually edit all of the node configuration files, follow the agent installation procedure.
4. After `opscenterd` and all agents have been configured and restarted, verify proper connection through the dashboard.



Enabling HTTPS

Enable Hypertext Transfer Protocol Secure (HTTPS) support in OpsCenter and specify SSL information for better security.

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

About this task

You can enable or disable HTTPS support for OpsCenter. To enable HTTPS, follow the steps below. To disable HTTPS, comment out the SSL entries again.

Procedure

1. Open the OpsCenter configuration file, opscenterd.conf.
2. Scroll to the `[webserver]` section.

   This snippet from `opscenterd.conf` shows the `[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.
4. Save `opscenterd.conf` and restart OpsCenter.

Using Kerberos authentication with OpsCenter

If a cluster uses Kerberos authentication, you need to create and configure the OpsCenter principles before adding the cluster to OpsCenter.

Procedure

1. Create an `opscenterd` principal and register it with Cassandra/DSE.

```
$ 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`.

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

2. Manually `kinit` the `opscenterd` user on the same account that runs the OpsCenter daemon.

There is a limitation on the Kerberos drivers used by OpsCenter that prevents OpsCenter from using a keytab.

3. Create service principals for the OpsCenter agent user running on each node and register them with Cassandra/DSE. The default user name is `opscenter-agent`.

```
$ cqlsh
cqlsh> create user 'opscenter-agent/Kerberos host@Kerberos domain';
```

4. Create keytabs for the `opscenter-agent` principals at `/usr/share/datastax-agent/krb5.keytab` on each node.
5. Set the owner of these keytabs and the `/usr/share/datastax-agent` directory to the `opscenter-agent` user.

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

6. When adding the cluster as described in Adding an existing cluster, check DSE Security and enter the service principal name for DSE.

Configuring events and alerts

The OpsCenter Event Log page in the Activities section displays a continuously updated list of events and alerts.

The OpsCenter Event Log page in the **Activities** section displays a continuously updated list of events and alerts. The following list reflects the most detailed logging level available for Cassandra, DataStax Enterprise, and OpsCenter events:

- `DEBUG` (0)
- `INFO` (1)
- `WARN` (2)
- `ERROR` (3)
- `CRITICAL` (4)
- `ALERT` (5)

Alerts

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. Alerts are disabled by default.

Alerts are triggered only by events from the OpsCenter API or UI. For example, a `nodetool` move operation submitted from the command line does not trigger an alert. However, a move operation launched using **Nodes** > **List View** > **Other Actions** > **Move** controls in the OpsCenter does trigger an alert.

All alerts contain the following information about each event captured:

Field	Description	Example
`api_source_ip`	IP that originally sent the request.	67.169.50.240
`target_node`	Destination of a STREAMING action.	10.1.1.11
`event_source`	Component that caused the event.	OpsCenter (i.e., restart, start)
`user`	OpsCenter user that caused the event.	opscenter_user
`time`	Normal timestamp for the event.	1311025650414527
`action`	Type of event (see Event action types)	20

Field	Description	Example
`subject`	Customizable subject line of the email alert.	[WARN] OpsCenter Event - Node reported as being down: 127.0.0.1
`message`	Description of the event.	Garbage Collecting node 10.1.1.13
`level`	Numerical code for the log level.	1
`source_node`	Node where the event originated.	10.1.1.13
`level_str`	Logging level of the event.	INFO

Enabling email alerts

OpsCenter can post alerts to multiple email addresses. To send alerts to multiple email addresses, create a different email configuration file with settings for each email address.

About this task

To enable email alerts, edit the *<config_location>*`/event-plugins/email.conf` file and provide valid SMTP server host and port information.

Before you begin

Make sure that you have valid SMTP mail accounts to send and receive alerts.

Procedure

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, user, and password.
4. (Recommended) 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.
6. Optional: Set the specific levels of alerts to send. The default is to listen for all levels.
7. Optional: Customize the subject line as desired. Available options are described in the config file comments.
8. Save *<config_location>*`/event-plugins/email.conf` and restart the OpsCenter daemon.

   To send alerts to multiple email addresses, create a different email configuration file with settings for each email address. All configuration files are loaded so name them as `email1.conf`, `email2.conf`, and so on.

Example

For a configuration with email alerts enabled for a warning level and SSL enabled, the *<config_location>*`/event-plugins/email.conf` looks like:

```
[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
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=cassandra_admin@acme.com
from_addr=mercury@gmail.com

# Customizable subject for email.  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
subject=[{level_str}] OpsCenter Event on {cluster} - {message}
```

Example

The email subject could appear as:

```
[WARN] OpsCenter Event - Node reported as being down: 127.0.0.1
```

Enabling alerts posted to a URL

OpsCenter can post alerts to a URL if you provide a correctly formatted POST script.

About this task

OpsCenter can be configured to send alerts within an HTTP POST request to a specified URL. For example, a simple PHP script containing `print_r($_POST);` will echo the received POST request. An example POST request is:

```
POST / HTTP/1.0
Host: localhost
User-Agent: Twisted PageGetter
Content-Length: 184
Content-type: application/x-www-form-urlencoded
connection: close

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

The request body contains fields described in Alerts.

To enable URL posting on the OpsCenter side:

Procedure

1. Edit the `posturl.conf` file and provide a path to your script.
   - Package installations: `/etc/opscenter/event-plugins`
   - Binary tarball installations (Linux and Mac OSX): *`install_location`*`/opscenter/conf/event-plugins`
   - Windows installations: `Program Files (x86)\DataStax Community\opscenter\conf\event-plugins`
2. Make sure your web server and posting script are configured to receive alerts.

3. On the OpsCenter daemon host, open `posturl.conf` for editing.
4. Set enabled to 1. For `url`, provide a valid path to your posting script.

```
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.

Example

In a system with posting enabled for critical and alert-level events, `posturl.conf` looks like:

```
[posturl]
enabled = 1
url =http://10.1.1.11/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 =CRITICAL,ALERT
```

Verifying that events are posting correctly

Set preferences to specify handling URL events posting on the receiving side.

About this task

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

Procedure

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
  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:

```
Array
( [api_source_ip ] => 67.169.50.240
   [target_node ] => None
   [event_source ] => OpsCenter
   [user ] => None
   [ time ] => 1311025598851602
   [action ] => 20
   [message ] => Garbage Collecting node 50.1.1.24
       [level ] => 1
       [source_node ] => 50.1.1.24
       [level_str ] => INFO
```

)

Event action types

The event action type can be one of a number of operations performed by OpsCenter. OpsCenter stores the event data it receives in the events and events_timeline column families in the OpsCenter keyspace.

Events

Data for events is stored in the `events` and `events_timeline` column families in the `OpsCenter` keyspace. The `action` value for an event is one of the events listed in this table:

Event	Code	Description
COMPACTION	0	Major compaction has occurred.
CLEANUP	1	Unused keys have been removed or cleaned up.
REPAIR	2	A repair operation has been initiated.
FLUSH	3	Memtables have been flushed to disk.
DRAIN	4	The commit log has been emptied, or drained.
DECOMMISSION	5	A leaving node has streamed its data to another node.
MOVE	6	Like NODE_MOVE; a new token range has been assigned.
NODE_DOWN	13	A node has stopped responding.
NODE_UP	14	An unresponsive node has recovered.
NODE_LEFT	15	A node has left, or been removed from, the ring.
NODE_JOIN	16	A node has joined the ring.
NODE_MOVE	17	A node has been assigned a new token range (the token has moved).
OPSC_UP	18	OpsCenter has been started and is operating.
OPSC_DOWN	19	OpsCenter was stopped or stopped running.
GC	20	Java garbage collection has been initiated.
KMIP_ERROR	45	KMIP server error conditions.

Configuring an alert for KMIP errors

Configure an alert to monitor KMIP error status. Alerting for KMIP error status is a DataStax Enterprise feature only.

About this task

Configure an alert to monitor KMIP server status. Alerting for KMIP server status is a DataStax Enterprise feature only. For more information, see configuring KMIP to use off-server encryption keys 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.

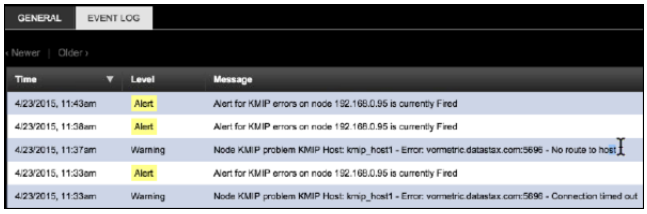
Procedure

1. Click the **Alerts** menu.
2. In the Active Alerts dialog, click **Manage Alerts**.

The Add Alert dialog appears.

3. In the **Notify me when** menu, choose **KMIP Error**.

4. Indicate the duration of the condition before alerting.
5. Select the notification frequency of the alert and click **Save Alert**.
   Any KMIP errors are displayed in the Event Log.



Configuring data collection and expiration

OpsCenter collects system and column family metrics data for each node in a cluster.

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 called OpsCenter. The column families containing metric data continue to grow. You can configure how long you want to keep historical metrics. Data expires after configurable time periods.

Estimating the amount of metrics data generated

Provides a guideline for the amount of column families monitored, the number of days monitored, and the MB per node of metrics data generated.

The following table provides guidance for estimating the amount of metrics data generated:

Number of days	Number of column families monitored	MB per node
31	5	200
31	10	300
31	20	500
365	5	250
365	10	380
365	20	630

The default upper limit of data collected is 365 days.

Controlling data collection

Discusses how OpsCenter helps control consumption of disk space when collecting performance data.

To help control consumption of disk space, OpsCenter limits the growth of OpsCenter performance data by:

- Excluding specified keyspaces and column families from performance data collection
- Shortening the time period after which performance data automatically expires

Excluding keyspaces and column families 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 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, uncomment and edit the following values in the OpsCenter cluster configuration file (cluster_name.conf):

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

Column families are specified in the format:

```
<keyspace_name>.<column_family_name>.
```

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
- Windows installations: Program Files (x86)\DataStax Community\opscenter\conf \clusters*cluster_name.conf*

Changing performance data expiration times

Performance data stored in OpsCenter expires after configurable time periods.

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
- Windows installations: Program Files (x86)\DataStax Community\opscenter\conf \clusters*cluster_name.conf*

About this task

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, and two-hour roll-ups are set to be kept indefinitely (expiration is disabled).

Procedure

1. Edit the cluster_name.conf file.
2. Add the following time-to-live (ttl) values in seconds under a [cassandra_metrics] section:

```
[cassandra_metrics]

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

3. Restart OpsCenter.

   Data collected after restarting OpsCenter expires according to the new setting. The data collected before restarting OpsCenter expires according to the setting in effect when it was collected.

Storing collection data on a different cluster

Store collection data on a separate DSE cluster as an alternative to OpsCenter storing data in an OpsCenter keyspace on the same DSE cluster being managed. Storing collection data in a separate storage cluster is an enterprise-only feature.

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
- Windows installations: Program Files (x86)\DataStax Community\opscenter\conf \clusters*cluster_name.conf*

If you do not want OpsCenter to store data in an OpsCenter keyspace on the DSE cluster being managed, you can store the data on a separate DSE cluster. Storing collection data in a separate storage cluster is an enterprise-only feature. OpsCenter supports connecting to a DSE storage cluster when SSL is enabled.

Before you begin

- The seed nodes must be accessible without Kerberos security.
- A unique keyspace must be used for each DSE cluster being managed by OpsCenter. If you are storing data for multiple clusters, we recommend 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
```

About this task

Procedure

1. Open the cluster configuration file cluster_name.conf for editing.
2. Add a [storage_cassandra] section with the applicable storage configuration options for your environment:

```
[storage_cassandra]
username = opsusr
```

```
password = opscenter
seed_hosts = host1, host2
api_port = 9160
cql_port = 9042
keyspace = OpsCenter_Cluster1
```

Available storage configuration options:

[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 Cassandra or DataStax Enterprise cluster by the seeds property in the `cassandra.yaml` configuration file. The default value is localhost.

[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] auto_node_discovery

Configure when using a different cluster for OpsCenter storage. Enables or disables auto-discovery of nodes. When disabled, OpsCenter only attempts to contact nodes in the seed list, and will not auto-discover nodes. By default this is `True`.

[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 keyspace used for OpsCenter storage details, this probably doesn't work since OpsCenter is frequently hardcoded.

[storage_cassandra] ssl_ca_certs

Configure when using a different cluster for OpsCenter storage. The server certificate to use to validate SSL for thrift connections.

[storage_cassandra] ssl_ca_certs

Configure when using a different cluster for OpsCenter storage. The server certificate to use to validate SSL for thrift connections.

[storage_cassandra] ssl_validate

Configure when using a different cluster for OpsCenter storage. Specifies whether the SSL thrift connection should be validated. The default value is True.

[storage_cassandra] ssl_client_pem

Configure when using a different cluster for OpsCenter storage. Specifies the client-side SSL PEM file to use if using two-way auth

[storage_cassandra] ssl_client_key

Configure when using a different cluster for OpsCenter storage. Specifies the client-side SSL key file to use if using two-way auth

The SSL configuration options are applicable to OpsCenter version 5.1.1+ only.

Configuring automatic updates of OpsCenter definition files

OpsCenter uses definition files to enable support for different versions of DataStax Enterprise, DataStax Community, and Cassandra without the need to upgrade the currently installed version of OpsCenter itself.

OpsCenter ships with a set of files called *definition files* that can be updated independently of OpsCenter itself. OpsCenter uses definition files to enable support for newer versions of DataStax Enterprise, DataStax Community, and Cassandra without the need to upgrade the currently installed version of OpsCenter itself. Definition files are updated independently of OpsCenter by automatically downloading new definitions at regular intervals from a central server. You can modify the default interval or disable the downloads altogether.

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. Definition file locations:

- `/etc/opscenter/conf/definitions` (package installs)
- *`install_location`*`/conf/definitions` (tarball installs)

The definition files are updated every hour by default. The default interval can be modified by setting the `sleep` option in the `[definitions]` section of opscenterd.conf. The `sleep` option interval should be specified in seconds.

Setting the update interval to 7200 seconds (every 2 hours):

```
[definitions]
sleep = 7200
```

Disabling definition file updates

The auto-update process can be disabled by setting `auto_update` to `False` in the `[definitions]` section of opscenterd.conf.

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
install_location/conf/opscenterd.conf			X	X

Automatic failover

Automatic failover from the primary DataStax OpsCenter instance to the backup OpsCenter instance provides high availability without any manual intervention or downtime.

Automatic failover provides continuous high availability of OpsCenter for managing mission-critical data on DataStax Enterprise clusters without manual intervention or downtime. Automatic failover is a DataStax Enterprise feature.

Currently, OpsCenter allows one backup instance to a primary instance in an active-passive configuration. The **OpsCenter Failover Enabled** Best Practice Rule recommends enabling failover. When no backup is configured, the rule fails and sends an alert. After enabling failover, 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-DSE 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 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` 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. 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.

Failover aftereffects

After an automatic failover, minimal manual intervention if any is required for recovery, depending on the root cause of the failover and what processes were in progress at that time. Generally, the effects of failing over are similar to restarting OpsCenter, with a few notable exceptions:

- **Alerts** - Trigger as normal. An exception is an alert firing and unfiring within the failover window; in which case the alert is never triggered.
- **Authentication** - Logs out existing user sessions. User sessions do not persist. Users must log in again.
- **Backup** - Skips a scheduled backup if it falls within the failover window. Backup does not occur until the next scheduled time.
- **Restore** - Continues the restore operation if failover occurred mid-restore; however, the result of the restore cannot be communicated because the backup OpsCenter was unaware the restore transpired.
- **Repair Service** - Resumes from the last saved state.
- **Provisioning** - Requires manual clean up under the following circumstances:
  - If failover occurs during launching EC2 instances, the instances launch but are not provisioned by OpsCenter.
  - If failover occurs during provisioning and installing on machines, the provision process ceases and leaves the machines in a partially-provisioned state.

Enabling automatic failover

Configure automatic OpsCenter failover from the primary OpsCenter instance to the designated backup OpsCenter instance.

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

About this task

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.

To enable automatic OpsCenter failover:

Procedure

1. Optional: Set up a hostname/IP that can switch between primary and backup OpsCenter instances to avoid changing the browser URL for OpsCenter if a failover occurs.

   If you do not set up a hostname or IP for seamless URL switching post-failover, inform your OpsCenter users of any alternate URL to access OpsCenter.
2. Mirror the configuration directories stored on the OpsCenter primary to the OpsCenter backup using the method you prefer, such as NFS mount or rysnc.
   a) If SSL is enabled, mirror the contents of the SSL configuration directory on the primary OpsCenter machine to the backup OpsCenter machine.
      - `/var/lib/opscenter/ssl` (package installs)

- *install_location*/ssl (tarball installs)

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

b) Mirror the contents of the main configuration directory on the primary OpsCenter machine to the backup OpsCenter machine.

- /etc/opscenter (package installs)
- *install_location*/conf (tarball installs)

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

Note: The failover_configuration_directory should *not* be mirrored across OpsCenter installs when configuring OpsCenter to support failover.

c) Create and run an automated script to keep the mirrored directories in sync.

The following example cron scripts run rsync to synchronize the configuration directories every 5 minutes for a package install:

```
*/5 * * * * /usr/bin/rsync -az /etc/opscenter <user>@<backup_host>:/etc/
opscenter
```

```
*/5 * * * * /usr/bin/rsync -az /var/lib/opscenter/ssl
 <user>@<backup_host>:/var/lib/opscenter/ssl
```

The following example cron scripts run rsync to synchronize the configuration directories every 5 minutes for a tarball install:

```
*/5 * * * * /usr/bin/rsync -az install_location/conf
 <user>@<backup_host>:install_location/conf
```

```
*/5 * * * * /usr/bin/rsync -az install_location/ssl
 <user>@<backup_host>:install_location/ssl
```

Note:

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

3. Optional: If you want to override the default values, edit the [failover] section of the OpsCenter configuration file opscenterd.conf.

Note: Making any changes to the opscenterd.conf file requires restarting OpsCenter.

Table 3: OpsCenter daemon failover default configuration parameters

Option	Description	Default
heartbeat_period	Frequency in seconds with which the primary OpsCenter sends a heartbeat to the backup OpsCenter.	10
heartbeat_reply_period	Frequency in seconds with which the OpsCenter backup sends a heartbeat to the primary OpsCenter.	300
heartbeat_fail_window	Amount of time in seconds that must elapse before the lack of a heartbeat triggers a failover.	60

Option	Description	Default
failover_configuration_directory	Directory location where failover-specific configuration is stored. **Note:** The failover configuration directory should *not* be mirrored or replicated across OpsCenter installs when configuring OpsCenter to support failover.	• `/var/lib/opscenter/failover/` (package installs) • `/opscenterd/failover/` (tarball installs)

4. On the backup OpsCenter in the failover directory, create a `primary_opscenter_location` configuration file that indicates the IP address of the primary OpsCenter daemon to monitor:
   - `/var/lib/opscenter/failover/primary_opscenter_location` (package installs)
   - `/opscenterd/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
   ```

   Before the backup OpsCenter can take over as the primary OpsCenter, the backup OpsCenter deletes the `primary_opscenter_location` file in the event of a failover. After a failover, recreate the `primary_opscenter_location` file on the newly designated backup OpsCenter.

Configuration files

Configure capabilities by manually modifying the opscenterd.conf, cluster_name.conf, and address.yaml 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 settings.

- 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 when you add a cluster to OpsCenter.
- address.yaml: configure the properties for the DataStax agent. You can set most of these properties in the `[agent_config]` section of cluster_name.conf on the opscenterd machine and the properties propagate automatically to all agents.

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`
- Windows installations: `Program Files (x86)\DataStax Community\opscenter\conf\clusters\`*`cluster_name.conf`*

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

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 configuration properties

Configure OpsCenter properties in the opscenterd.conf file and restart OpsCenter.

These properties are configured in the opscenterd.conf file.

Note: After changing properties in this file, restart OpsCenter 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.

[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] log_path

The location where HTTP traffic to the service is logged. The default is `/var/log/opscenter/http.log` for package installations and *`install_location`*`/log/http.log` 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] 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] tarball_process_timeout

The timeout, in seconds, to deliver the agent tarball to a node during agent installation. The default value is 600 seconds.

[logging] level

The logging level for OpsCenter. Available levels are (from most to least verbose): TRACE, DEBUG, INFO, WARN, or ERROR. The default level is INFO.

The OpsCenter log file is located in `/var/log/opscenter/opscenterd.log`.

[logging] resource_usage_interval

The interval, in seconds, in which OpsCenter logs the system resource usage. The default value is 60.

[logging] log_path

The log file for OpsCenter. The default location is `/var/log/opscenter/opscenter.log` for package installations and *`install_location`*`/log/opscenterd.log` for tarball installations.

[logging] log_length

Logs will rotate after the specified number of bytes. The default is 10000000 (10MB).

[logging] max_rotate

The maximum number of logs to retain. The default value is 10.

[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.

[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 defintion 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).

[definitions] ssl_certfile

The SSL certificate file used for SSL communication with the definition file `download_host`. The default file is `/var/lib/opscenter/ssl/definitions.pem` for package installations and *`install_location`*`/ssl/definitions.pem` for tarball installations.

[authentication] passwd_db

Full path to the file for configuring password authentication 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.

[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 defalt port is 22. Add an [agents] section, if one doesn't already exist, to the opscenterd.conf. In this section, add the ssh_port option and a value for the port number:

```
ssh_port = 2222
```

[agents] incoming_port

The port used by OpsCenter for incoming stomp communication. The default port is 61620.

[agents] incoming_interface

The interface used by OpsCenter for incoming stomp traffic from the 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] install_throttle

The maximum number of concurrent agent installs OpsCenter will attempt. The default value is 20. Keeping this value low prevents high CPU usage during agent installs but increasing it may make agent installs complete faster.

[agents] fingerprint_throttle

The maximum number of concurrent SSH fingerprints OpsCenter will process when provisioning or installing agents. The default value is 50. Keeping this value low prevents high CPU usage during agent installs but increasing it may make agent provisioning and installs complete faster.

[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.pem` for package installations and *`install_location`*`/ssl/opscenter.pem` 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.pem` for package installations and *`install_location`*`/ssl/agentKeyStore.pem` for tarball installations.

[agents] ssh_executable

The location of the `ssh` executable binary. The default locations is `/usr/bin/ssh`.

[agents] scp_executable

The location of the `scp` executable binary. The default location is `/usr/bin/scp`.

[agents] ssh_keygen_executable

The location of the `ssh-keygen` executable binary. The default location is `/usr/bin/ssh-keygen`.

[agents] ssh_keyscan_executable

The location of the `ssh-keyscan` executable binary. The default location is `/usr/bin/ssh-keyscan`.

[agents] ssh_user_known_hosts_file

The location of the OpsCenter user's `known_hosts` file that will be used by OpsCenter during SSH communications. The default location is `~/.ssh/known_hosts`.

[agents] ssh_sys_known_hosts_file

The location of the system wide `known_hosts` file that will be used by OpscCnter during SSH communications. The default location is `/etc/ssh/ssh_known_hosts`.

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

[agents] path_to_find_java

The location of the `find-java` shell script, used to find the location of Java on the agent machine. The default is `/usr/share/opscenter/agent/bin/find-java` for package installations and *`install_location`*`/agent/bin/find-java` 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] 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 time in seconds after an agent request has been received after which the agent is considered down.

[agents] reported_interface

The interface that OpsCenter tells agents to connect to for STOMP communication. It is not set by default and OpsCenter will try to automatically detect the interface.

[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.

[stat_reporter] initial_sleep

The delay in seconds before the cluster stats reporter starts to run. The default value is 300 (5 minutes).

[stat_reporter] interval

The interval in seconds between usage metric reports to DataStax Support. By default, OpsCenter sends usage metrics about the cluster to DataStax Support every day.

To disable the phone-home functionality, add the following lines to your opscenterd.conf file:

```
interval = 0
```

Additional configuration metric collection properties are available in Metrics Collection Properties.

[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.

[stat_reporter] report_file

The location where generated PDF reports on the cluster are stored. The default location is `/usr/share/opscenter/cluster-report.pdf` for package installations and *`install_location`*`/cluster-report.pdf` 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 so following the Hadoop model, we start incrementing from there.

[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.

[provisioning] private_key_dir

The folder containing private SSL key files used when provisioning new clusters. The default location is `/var/lib/opscenter/ssl` for package installations and *`install_location`*`/conf` for tarball installations. You may alternately specify the key file using the OpsCenter API when provisioning.

[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.pem`

[labs] disable_data_explorer

This option will disable the data explorder in opscenter

[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] disable_backup_service_java7_check

Disables the java7 check for backup service in the UI in case our parsing is broken somehow

[repair_service] log_directory

The location in which to store repair service logs. The default location is `/var/log/opscenter/repair_service/` for package installations and *`install_location`*`/log/repair_service` for tarball installations.

[repair_service] log_length

Logs will rotate after the specified number of bytes. Defaults to 10485760 (10MB).

[repair_service] max_rotate

The maximum number of logs to retain. The default is 10.

[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. The default value is 0.

[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] repair_estimation_factor

Estimated reduced efficiency due to other issues like 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

Whether incremental repairs should do subrange repair or full repair of a node's entire range.

[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.

[ui] default_api_timeout

The default timeout value in seconds for an API call from the OpsCenter UI to the OpsCenter API. The default value is 10.

[ui] max_metrics_requests

The maximum concurrent metrics requests from the OpsCenter UI to `opscenterd`. The default value is 16.

[ui] node_detail_refresh_delay

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

[ui] storagemap_ttl

How often, in seconds, the data in the storage capacity chart is updated in the OpsCenter UI. It is set to 300 seconds (5 minutes) by default so changes to storage capacity on individual nodes may not be reflected in the UI for up to 5 minutes.

[request_tracker] queue_size

The maximum number of requests that can be tracked. The default value is 10,000.

[clusters] add_cluster_timeout

How long, in seconds, OpsCenter will wait when adding a cluster before reporting an error. The default value is 30 seconds. Adding a cluster includes things like connecting to THRIFT, getting a node list, 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).

[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.

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

Statistics reporter properties

A complete breakdown of the data OpsCenter communicates back to DataStax. The data is sent in a key-value JSON format.

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`, `windows`, 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.

cassandra_versions
A list of the different Cassandra versions in the cluster.

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.

OpsCenter logging properties

Configure the location and logging behavior properties of OpsCenter log files.

The following properties configure the location and properties of OpsCenter log files:

[webserver] log_path

The location where HTTP traffic to the service is logged. The default is `/var/log/opscenter/http.log` for package installations and *`install_location`*`/log/http.log` for tarball installations.

[logging] level

The logging level for OpsCenter. Available levels are (from most to least verbose): TRACE, DEBUG, INFO, WARN, or ERROR. The default level is INFO.

The OpsCenter log file is located in `/var/log/opscenter/opscenterd.log`.

[logging] log_path

The log file for OpsCenter. The default location is `/var/log/opscenter/opscenter.log` for package installations and *`install_location`*`/log/opscenterd.log` for tarball installations.

[logging] log_length

Logs will rotate after the specified number of bytes. The default is 10000000 (10MB).

[logging] max_rotate

The maximum number of logs to retain. The default value is 10.

[repair_service] log_directory

The location in which to store repair service logs. The default location is `/var/log/opscenter/repair_service/` for package installations and *`install_location`*`/log/repair_service` for tarball installations.

[repair_service] log_length

Logs will rotate after the specified number of bytes. Defaults to 10485760 (10MB).

[repair_service] max_rotate

The maximum number of logs to retain. The default is 10.

OpsCenter updater properties for definition files

OpsCenter updater properties configure the updates for definition files that enable support for different releases of DataStax Enterprise, DataStax Community, and Cassandra.

These properties are for configuring the OpsCenter updater, which updates the definition files that enable support for different releases of DataStax Enterprise, DataStax Community, and Cassandra.

Configure the definition file properties in the opscenterd.conf file.

Note: After changing properties in this file, restart OpsCenter for the changes to take effect.

[definitions] auto_update

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

[definitions] sleep

The duration in seconds between checks to update the definition files. The default time is 3600 seconds (1 hour).

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

Cluster configuration properties

Cluster configuration properties inform OpsCenter about the Real-time (Cassandra), Analytics (Hadoop), and Search (Solr) nodes that it is monitoring.

Cassandra connection properties

Configure Cassandra connection properties in the cluster-specific configuration file cluster_name.conf.

These properties are configured in the cluster-specific configuration file, cluster_name.conf.

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.

[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 Cassandra or 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 location of the cassandra system logs on a node. The default location is `/var/log/cassandra/system.log`.

[cassandra] auto_node_discovery

Enables or disables auto-discovery of nodes. When disabled, OpsCenter only attempts to contact nodes in the seed list, and will not auto-discover nodes. By default this is `True`.

[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_ca_certs

The server certificate to use to validate SSL for thrift connections.

[cassandra] ssl_validate

Specifies whether the SSL thrift connection should be validated. The default value is True.

[cassandra] ssl_client_pem

Specifies the client-side SSL PEM file to use if using two-way auth.

[cassandra] ssl_client_key

Specifies the client-side SSL key file to use if using two-way auth.

[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.

[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] auto_node_discovery

Configure when using a different cluster for OpsCenter storage. Enables or disables auto-discovery of nodes. When disabled, OpsCenter only attempts to contact nodes in the seed list, and will not auto-discover nodes. By default this is `True`.

[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 keyspace used for OpsCenter storage details, this probably doesn't work since OpsCenter is frequently hardcoded.

[storage_cassandra] ssl_ca_certs

Configure when using a different cluster for OpsCenter storage. The server certificate to use to validate SSL for thrift connections.

[storage_cassandra] ssl_validate

Configure when using a different cluster for OpsCenter storage. Specifies whether the SSL thrift connection should be validated. The default value is True.

[storage_cassandra] ssl_client_pem

Configure when using a different cluster for OpsCenter storage. Specifies the client-side SSL PEM file to use if using two-way auth

[storage_cassandra] ssl_client_key

Configure when using a different cluster for OpsCenter storage. Specifies the client-side SSL key file to use if using two-way auth

[collection] basic_info_period

The frequency, in seconds, to check Cassandra for a Cassandra API update. The default value is 3600 (1 hour).

[collection] node_poll_period

This appears to be read but unused

[collection] nodelist_poll_period

The interval in seconds OpsCenter waits to poll the nodes in a cluster. The default value is 30.

[collection] job_poll_period

The frequency, in seconds, to poll the job tracker. The default value is 5.

[collection] cf_poll_period

The frequency, in seconds, to check for a schema update. The default value is 60.

[collection] push_throttle_period

The frequency, in seconds, to push node information to the UI. The default value is 60.

[metric_storage] plugin

I think this is unused

[metric_storage] metric_poll_period

I think this is unused

[metric_caching] num_data_points_cached

The number of data points to cache for cluster metrics. The default value is 50.

[metric_caching] num_metrics_cached

The number of metrics to cache for cluster metrics. The default value is 1000.

[agents] api_port

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.

[agents] ssl_keystore_password

The SSL truststore password for agents to use to connect to CQL.

[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 commitlogs are stored and retained on each node. This property must be set prior to enabling commitlog archiving."

[agents] restore_req_update_period

The frequency (in seconds) that a restore will report progress back to OpsCenter. The default value is automatically calculated based on cluster size. To optimize performance, larger clusters will have a longer threshold in which restore progress will be sent to the UI. Configure this property if you would like to see more/less frequent updates during a restore. Note: lower numbers may impact performance during a restore

[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 column families 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 valie 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 -1, 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 backup to. They shouldn't have spaces and should be delimited by comments.

[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.

[kerberos] default_service

The default Kerberos service name.

[kerberos] default_hostname

The default Kerberos hostname.

[kerberos] default_client_principal

The default Kerberos client principal.

[kerberos] default_client_user

The default Kerberos client user.

[kerberos] opscenterd_client_principal

The OpsCenter client principal in Kerberos.

[kerberos] job_tracker_client_principal

The job tracker client principal in Kerberos.

[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, only polyfit is an option currently

[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 can fail to respond before a remote destination restore action will fail

[agent_config] Empty Section

Empty Section

[dse] Empty Section

Empty Section

[repair_service] Empty Section

Empty Section

[repair_service] example_key

Anything in this section will be used to override opscenter level repair_service items

[kerberos_hostnames] Empty Section

Empty Section

[kerberos_hostnames] 192.168.1.101

Per-node specification for the Kerberos hostname of the service (DSE). A list of IP, hostname pairs. For example `192.168.1.101 = cassandra01.example.com`.

[kerberos_services] Empty Section

Empty Section

[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] Empty Section

Empty Section

[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] Empty Section

Empty Section

[cluster_display_options] display_name

Display name used by OpsCenter to signify this 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`
- Windows installations: `Program Files (x86)\DataStax Community\opscenter\conf \clusters\`*`cluster_name.conf`*

Metrics Collection Properties

Use these metrics properties to limit the keyspaces and column families for metrics collection and change default expiration periods (ttl) for performance data.

[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 column families to **not** collect metrics for, separated by commas. Each entry should be of the form "ks.cf". For example: `metrics_ignored_column_families = system.NodeInfo, system.Schema, Keyspace1.Standard1`

[cassandra_metrics] ignored_solr_cores

A list of solr cores to **not** collect metrics for, separated by commas. Each entry should be of the form "ks.cf". For example: `metrics_ignored_solr_cores = Keyspace1.Standard1, solr.wiki`.

These properties set the expiration time for data stored in the OpsCenter keyspace. Each time period for rolling up data points into summary views has a separate expiration threshold, or time-to-live (ttl) value expressed in seconds. By default, shorter time periods have lower values that result in more efficient expiration and compaction of the relatively larger volumes of data. Uncomment these properties to change the default expiration periods for performance data. Properties and default values are:

[cassandra_metrics] 1min_ttl

Sets the time in seconds to expire 1 minute data points. The default value is 604800 (7 days).

[cassandra_metrics] 5min_ttl

Sets the time in seconds to expire 5 minute data points. The default valie 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 -1, or never.

DataStax Agent configuration

Configure DataStax agents with options in the address.yaml file. Fix and troubleshoot agent connections.

Agent auto-connect

If you are adding an existing cluster to OpsCenter, and the nodes do not have the agent installed, OpsCenter displays a **Fix Agents** link next to the number of nodes. Clicking **Fix Agents** causes OpsCenter to attempt to install and start the agent on any nodes that do not have a running agent, as described in Automatic installation of DataStax agents and Installing DataStax agents.

If clicking **Fix Agents** did not work, there are a number of things that could prevent OpsCenter from successfully installing and starting the agent on the nodes. For example, if the agent was previously installed on the node and has an incorrect configuration, OpsCenter cannot connect to the agent. Some simple things to check if the agent fails to connect:

- Verify the agent is able to start up without errors on the node. Look through the agent logs for errors.
- Check the settings of the `stomp_interface` and `stomp_port` options for each agent to make sure they match and are pointing to the OpsCenter host and port, respectively.
- Test that the OpsCenter host and port are accessible from the node. A firewall might be blocking access to the OpsCenter port, for example.
- Verify that agents have the correct configuration settings for SSL communication if OpsCenter is configured to use SSL communication between itself and the agents.
- Test that the SSH credentials you have entered in OpsCenter work on each node.
- Verify JMX connectivity is enabled on the node.

In most environments, `stomp_interface` is the only property that will need to be explicitly configured, and this might happen automatically as previously mentioned. You can set most of these properties in the `[agent_config]` section of cluster_name.conf on the opscenterd machine and the properties propagate automatically to all agents. Some properties or some cases will require setting these properties directly in `address.yaml` on applicable agents.

The address.yaml configuration file

The address.yaml file contains configuration options for the DataStax Agent.

Note: As of version 5.1 of OpsCenter, the `hosts` option in `address.yaml` now determines which nodes the agent connects to. For further information on configuration changes and migration paths, see the Upgrade Guide.

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:

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

Options

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. 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"]`

alias

A name for the node to use as a label throughout OpsCenter.

restore_on_transfer_failure

Determines whether to allow a restore operation from S3 to continue after any file fails to download. Setting the value to true allows the restore operation to continue. The default setting is false.

max-number-queued-files

Sets the maximum number of files to queue for uploading to a remote backup destination. Increasing this number impacts memory. The default setting is 100000.

stomp_interface

Reachable IP address of the `opscenterd` machine. The connection made will be on `stomp_port`.

stomp_port

The `stomp_port` used by `opscenterd`. The default setting is 61620.

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.

agent_rpc_interface

The IP that the agent HTTP server listens on. In a multiple region deployment, this is typically a private IP.

agent_rpc_broadcast_address

The IP that the central OpsCenter process uses to connect to the DataStax agent.

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. The default setting is `0`.

cassandra_conf

The agent will attempt to auto-detect the location of the `cassandra.yaml` file via JMX, but if it cannot this needs to be set to the full path of `cassandra.yaml`. By default `/etc/cassandra/cassandra.yaml` on package installs or `<install_location>/conf/cassandra.yaml` on tarball installs.

cassandra_install_location

The location where Cassandra is installed for tarball installs if OpsCenter is unable to auto-detect the install location.

cassandra_log_location

The location of Cassandra's `system.log` file. This is only used for the diagnostics tarball, and should only be set if `system.log` is in a non-standard location.

tmp_dir

The location of the Backup Service staging directory for backups. The default location is `/var/lib/datastax-agent/tmp`.

Advanced options

metrics_enabled

Whether or not to collect and store metrics for the local node. Setting this option to `false` turns off metrics collection. The default setting is `true`.

cassandra_port

Port used to connect to Cassandra. The default setting is 9042. This information will be sent by `opscenterd` for convenience, but can be configured locally as needed.

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.

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.

api_port

Port the local HTTP server will bind to. The default setting is 61621. This option needs to be identical across all agents, and set explicitly in opscenterd.conf if changed.

runs_sudo

Sets whether the DataStax Agent will be run using sudo. 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.

The default setting is `true`.

Customize scripts for starting and stopping DataStax Enterprise and Cassandra

OpsCenter allows starting and stopping the DataStax Enterprise/Cassandra process on each node in a visual manner. Customize the startup or shutdown of a node using the provided example scripts.

About this task

OpsCenter allows starting and stopping the DataStax Enterprise/Cassandra 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).

Procedure

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
   ```

Example scenarios

Example scenarios of OpsCenter deployments include configuring for multiple regions, IP forwarding, or very large clusters.

Configuring for multiple regions

OpsCenter can operate in multiple regions or IP-forwarding deployments. Configure address.yaml to accommodate multiple regions, or to forward a public IP to a private IP address on the agent.

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`

About this task

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:

Procedure

1. Open the address.yaml file for editing.
2. Add the following options to the `address.yaml` file:
   - **local_interface**: (Optional) 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.

- `agent_rpc_interface`: The IP that the agent HTTP server listens on. In a multiple region deployment, this is typically a private IP.
- `agent_rpc_broadcast_address`: The IP that the central OpsCenter process uses to connect to the DataStax agent.

3. Repeat the above steps for each node.

Example

Here is the configuration for a three node cluster that spans two regions:

```
Region: us-west
Availability Zone: us-west-2

OpsCenter host
    public IP: 198.51.100.5
    private IP: 10.11.12.10

Node1
    public IP:  198.51.100.1
    private IP: 10.11.12.1
    Cassandra (cassandra.yaml)
        broadcast_address: 198.51.100.1
        listen_address:    10.11.12.1
    Agent  (address.yaml)
        local_interface:             198.51.100.1
        agent_rpc_interface:         10.11.12.1
        agent_rpc_broadcast_address: 198.51.100.1
        stomp_interface:             198.51.100.5

Node2
    public IP:  198.51.100.23
    private IP: 10.11.12.15
    Cassandra  (cassandra.yaml)
        broadcast_address: 198.51.100.23
        listen_address:    10.11.12.15
    Agent  (address.yaml)
        local_interface:             198.51.100.23
        agent_rpc_interface:         10.11.12.15
        agent_rpc_broadcast_address: 198.51.100.23
        stomp_interface:             198.51.100.5

Region: us-east
Availability Zone: us-east-1

Node1
    public IP:  203.0.113.20
    private IP: 10.11.13.28
    Cassandra  (cassandra.yaml)
        broadcast_address: 203.0.113.20
        listen_address:    10.11.13.28
    Agent  (address.yaml)
        local_interface:             203.0.113.20
        agent_rpc_interface:         10.11.13.28
        agent_rpc_broadcast_address: 203.0.113.20
        stomp_interface:             198.51.100.5
```

Configuring for very large clusters

OpsCenter can manage clusters containing multiple hundreds of nodes. When managing very large clusters with up to 1000 nodes, adjusting cluster configuration settings improves performance.

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
- Windows installations: Program Files (x86)\DataStax Community\opscenter\conf\clusters*cluster_name.conf*

About this task

OpsCenter can manage very large clusters up to 1000 nodes. When working with very large clusters, the performance of OpsCenter decreases with the default settings. To improve performance, adjust the cluster settings to turn off automatic node discovery and increase the time period between polls of a cluster's nodes and token lists.

After a very large cluster has been added to OpsCenter, change the following settings in the cluster_name.conf.

Procedure

1. Turn off auto-discovery of nodes by setting the auto_node_discovery option to False under [cassandra].

   ```
   [cassandra]
   auto_node_discovery = False
   ```

2. 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
   ```

Using OpsCenter

OpsCenter is a Web application for monitoring and administering all nodes in a Cassandra cluster from one centralized console. OpsCenter runs on the client-side in a web browser.

OpsCenter Workspace Overview

Describes the major areas of functionality available in the OpsCenter workspace.

At the top of every functional area of OpsCenter, you can access these functions:

- New Cluster - create a new cluster or add an existing cluster.
- Alerts - Configure alert thresholds for a number of Cassandra cluster-wide, column family, and operating system metrics. Available for DSE clusters only.
- Settings
  - Cluster Connections - Modify cluster settings or remove the cluster from OpsCenter.
  - Users & Roles - Enable, disable, and manage role-based authentication.
- Help: online help, download a tarball containing diagnostic information about the nodes in a cluster, get a PDF report with information on the managed clusters, send feedback, and view the OpsCenter version number.
- Username - If authentication is enabled, the username is displayed. You can change your password or log out by clicking the username.

The left navigation pane displays a link to the Overview section and a list of each cluster. The Overview section displays any active alerts and a summary box for each cluster managed by OpsCenter.

OpsCenter is divided into these main functional areas:

- Dashboard - View information about the clusters managed by OpsCenter and monitor a number of Cassandra cluster performance metrics. Real-time and historical performance metrics are available at different granularities: cluster-wide, per node, or per column family. Update notifications for upgrades to DSE or Cassandra are also displayed in the Dashboard.
- Nodes - See your cluster from different perspectives and perform certain maintenance operations on cluster nodes.
- Activities - Displays all running tasks in the cluster. The Activities icon displays a badge when tasks are currently running. View the most recent OpsCenter log events, such as OpsCenter startup and shutdown, in the Event Log. View the status of Hadoop analytics jobs under Hadoop Jobs.
- Data
  - Tables - Create and manage keyspaces and the tables (column families) within those keyspaces.
  - Backups - Visually take, schedule, and manage backups across all registered clusters. Restore to clusters from backups. Available for DSE clusters only.
  - Explorer - Browse through table (column family) data.
- Services - Enable DataStax Enterprise services, including the Backup, Repair, Capacity, and Best Practice services.
- Hide/Show Clusters - Hide or show the list of clusters on the left side of the UI.

Using OpsCenter authentication

Describes logging in and out of OpsCenter and changing the user password when authentication is enabled.

About this task

If authentication is enabled, follow these instructions to log in or log out of OpsCenter and change the user password.

Procedure

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 password is `admin`.
3. To change the user password:
   a) Click the username on the upper right and select **Change Password**.
   b) Enter the current password, enter the new password, confirm the new password, and click **Submit**.
4. Log out by clicking your username in the top navigation bar and clicking **Sign Out**.

Managing clusters

The New Cluster menu command in OpsCenter allows creating new clusters or managing existing clusters.

Creating a cluster

Provision a new Cassandra or DataStax Enterprise cluster locally or in the cloud using OpsCenter.

About this task

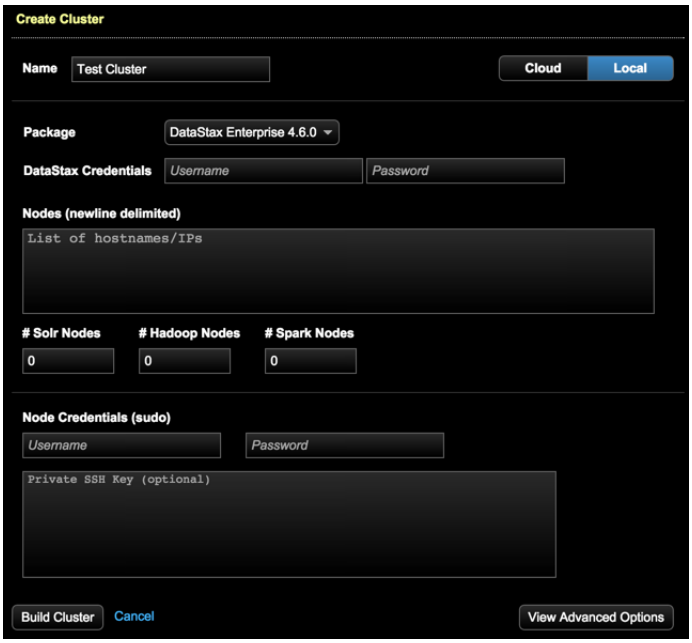
Follow these instructions to provision a new local or cloud cluster using OpsCenter. OpsCenter can provision new Cassandra and DataStax Enterprise clusters. Each node in the cluster must meet the following requirements:

- Oracle Java 7+ is installed.
- Cassandra or DataStax Enterprise is *not* installed.
- For local clusters, have a user capable of using `sudo`, unless OpsCenter has been configured not to use `sudo` (see the `runs_sudo` option described in OpsCenter configuration properties).
- Port access: When provisioning cloud clusters on EC2 nodes, the OpsCenter machine needs access to port 61621 on the managed nodes, and the nodes need access to port 61620 on the OpsCenter machine.

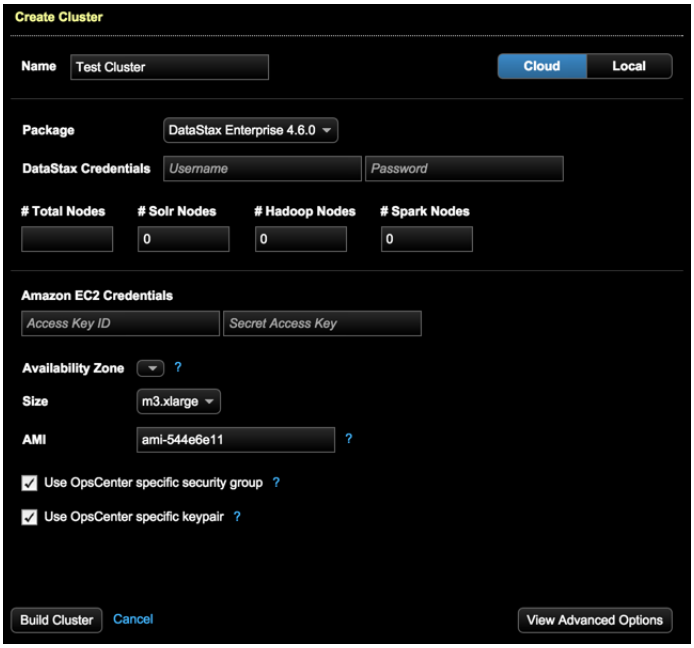
Procedure

1. Click **New Cluster**.
2. Click **Create Brand New Cluster**.
   The **Create Cluster** dialog appears.
3. Click **Cloud** or **Local**.

   The **Cloud** and **Local** buttons appear only if OpsCenter is running in the cloud.
4. Complete the fields as appropriate.
   - If you select **Local**, the Local pane appears.



- If you select **Cloud**, the Cloud pane appears. OpsCenter currently only supports the AWS EC2 cloud.



Consult the following table for assistance with completing the fields:

Table 4: New cluster fields

Field	Description
Name	Cluster name.
Package	The version of Cassandra or DataStax Enterprise to install on the nodes.

Field	Description
DataStax Credentials	<*userid*> and <*password*> that were in the email received from DataStax when registering to download DataStax Enterprise.
Nodes (Local only)	A list of existing machines on which to install the cluster. For best results, use private IP addresses.
Total Nodes (Cloud only)	Total number of Cassandra or DataStax Enterprise nodes for the cluster.
# Solr Nodes	Total number of Solr nodes for the cluster.
# Spark Nodes	For DataStax Enterprise 4.5 and higher clusters, the total number of Spark nodes for the cluster.
# Hadoop Nodes	Total number of Hadoop nodes for the cluster.
Node Credentials (Local only)	The <*userid*> and <*password*> of the user with sudo permission on the nodes.
Private SSH Key (Local only)	The private SSH key to use instead of Node Credentials.
Amazon EC2 Credentials (Cloud only)	The <*access-jey-id*> and <*secret-access-key*> to use to authenticate on AWS EC2.
Availability Zone (Cloud only)	Which availability zone to use to create the cluster. The list is only populated after entering your EC2 credentials.
Size (Cloud only)	Which size image to use. For more information about available options, see the DataStax Developer blog about AWS EC2 instance types.
AMI (Cloud only)	Which image to use.
Use OpsCenter-specific security group (Cloud only)	Determines whether OpsCenter creates its own specific security group or allows selecting one that is available using your EC2 credentials.
Use OpsCenter-specific keypair (Cloud only)	Determines whether OpsCenter creates its own specific keypair or allows you to select one that is available using your EC2 credentials.
View Advanced Options	Access `cassandra.yaml` for Configuring a cluster.

5. Click **Build Cluster**.

Results

The agent installs successfully and the new cluster is now available.

If the agent fails to install:

- On the affected nodes, check `opscenterd.log` and `/var/log/datastax-agent/installer.log`.
- Verify that the correct ports are open between machines as described in OpsCenter and DataStax agent ports.

If the agent installed successfully but there is an issue with the Cassandra or DataStax Enterprise setup process, check the following logs for any errors:

- `opscenterd.log`
- `/var/log/datastax-agent/agent.log`
- `/var/log/cassandra/output.log`
- `/var/log/cassandra/system.log`

Accessing Amazon EC2 instances created by OpsCenter

Instructions for logging in with SSH to Amazon EC2 instances created by OpsCenter.

About this task

If you are running OpsCenter on Amazon EC2, you can use `ssh` to log in to the instances created by OpsCenter. Using the default AMI, the username is `ubuntu`. The private key is located in `/var/lib/opscenter/ssl/` and is named after the region in which the `opscenterd` instance is running appended with `-OpsCenterProvisioningKeyPair.pem`. For example, the private key might be `US_West_(Northern_California)-OpsCenterProvisioningKeyPair.pem`.

Due to the way SSH handles permissions on the private key file, you must use `sudo` to call `ssh` unless you make a copy of the private key and move it to a location owned by a non-root user.

Procedure

Use SSH to log in to the EC2 instance.

```
$ sudo ssh -i "/var/lib/opscenter/ssl/US_West_(Northern_California)-
OpsCenterProvisioningKeyPair.pem" ubuntu@10.20.30.40
```

Adding an existing cluster

Add an existing cluster using OpsCenter.

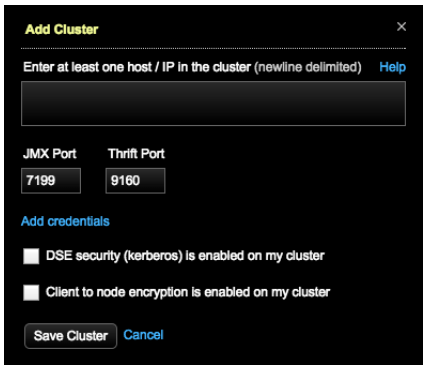
About this task

You can an existing cluster to manage within OpsCenter.

OpsCenter automatically enables commitlog archiving on the new nodes when adding new nodes to a cluster using OpsCenter if commitlog archiving is enabled on the cluster. If you manually 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.

Procedure

1. Click **New Cluster**.
2. The New Cluster prompt appears.
3. Click **Manage Existing Cluster**.
   The **Add Cluster** dialog appears.



4. 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
```

5. If you are not using the default JMX or Thrift ports, enter the appropriate port numbers.
6. If required, click **Add Credentials** and enter the username and password for JMX or Thrift ports.
7. Optional: Select the **DSE security (kerberos) is enabled on my cluster** check box and enter the service name.
8. Optional: Select the **Client node encryption is enabled on my cluster** check box and enter your PEM-encoded certificate in the **CA Certificate File Path** field.

   If you have a CER-encoded certificate, use the following command to convert it:

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

9. Optional: If client node encryption is enabled, select **Validate SSL Certificates** and enter the **Keystore File Path** and **Keystore Password**.

   For more information about enabling Kerberos, see Security in the DSE Documentation.
10. Click **Save Cluster**.

Retrying a failed install

Retry creating a cluster or adding nodes after resolving any issues encountered during a failed install.

If an error occurs while creating a cluster or adding nodes, click the **Retry** link to return to the Create Cluster dialog after the cause of the problem has been determined and resolved.

If you are installing DSE on existing machines (that is, the **Local** pane), ensure that all relevant software has been uninstalled beforehand.

If you are installing on new EC2 instances (that is, the **Cloud** pane), brand new instances are created when retrying; existing instances that were part of the failure must be terminated manually.

Modifying OpsCenter cluster connections

Cluster Connections settings define how OpsCenter connects to a cluster. Edit the settings if you have enabled authentication or encryption on a cluster.

About this task

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

Procedure

1. Select the cluster you want to edit from the **Cluster** menu.
2. Click **Settings** > **Cluster Connections**.
   The Edit Cluster dialog appears.
3. Change the IP addresses of cluster nodes if applicable.
4. Change JMX and Thrift listen port numbers if applicable.
5. Edit the user credentials if the JMX or Thrift ports require authentication.
6. Optional: Select the **DSE security (kerberos) is enabled on my cluster** check box and enter the service name.
7. Optional: Select the **Client node encryption is enabled on my cluster** check box and enter your PEM-encoded certificate in the **CA Certificate File Path** field.

   If you have a CER-encoded certificate, use the following command to convert it:

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

8. Optional: If client node encryption is enabled, select **Validate SSL Certificates** and enter the **Keystore File Path** and **Keystore Password**.

   For more information about enabling Kerberos, see Security in the DSE Documentation.
9. Click **Save Cluster**.

Node monitoring and administration

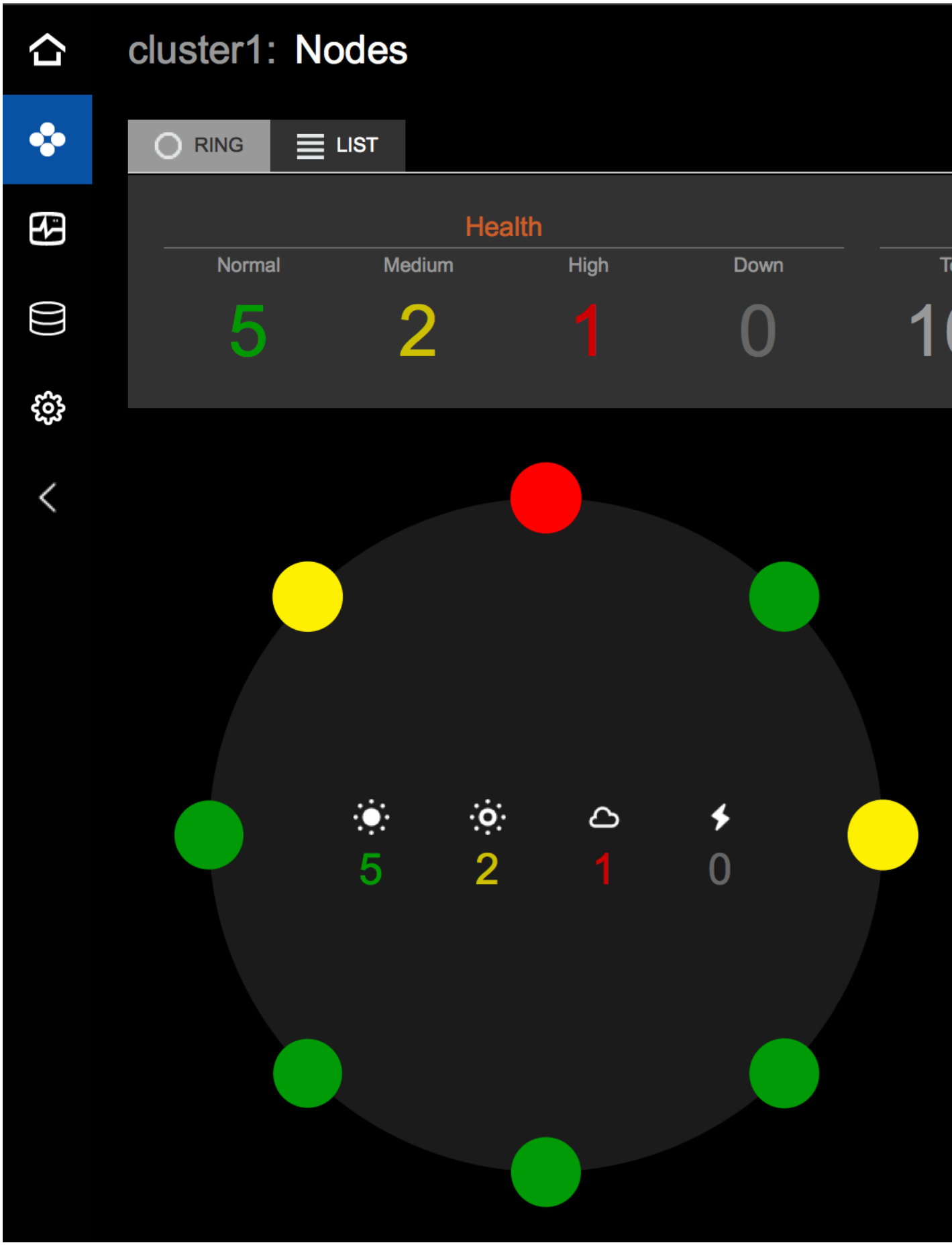
In the Cluster section of OpsCenter, select different views of the nodes comprising a Cassandra cluster and then perform node management.

Ring View

The Ring View displays a cluster as a ring of nodes from which node health, data distribution, and data center balance can be determined at a glance within a single visualization.

The Ring View displays a cluster as a ring of nodes from which node health, data distribution, and data center balance can be determined at a glance within a single visualization:

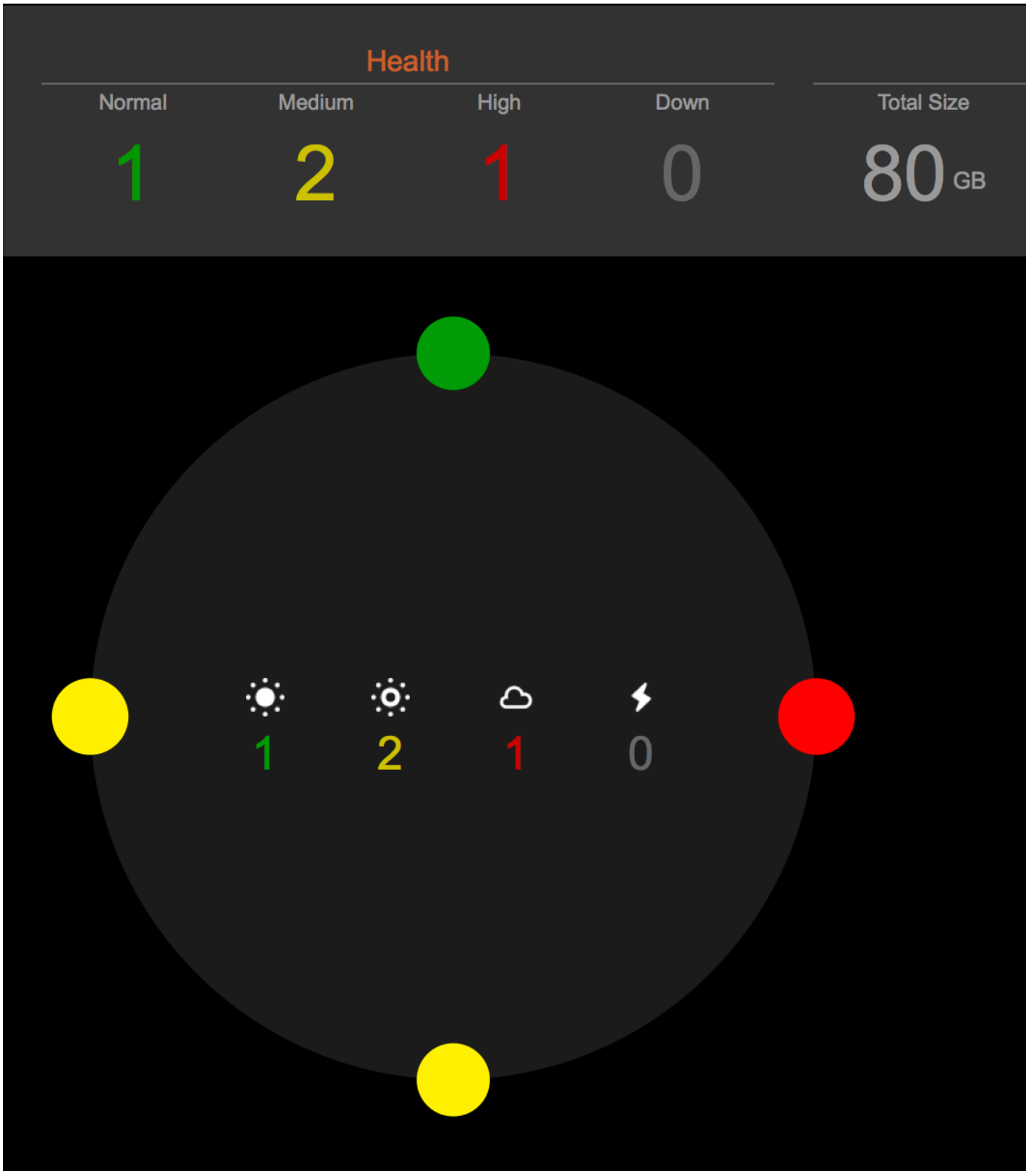
- Nodes are positioned around the ring according to their assigned token. In the case of `ByteOrderedPartioner` or vnodes, nodes are displayed as slices of the ring and sized based on the percentage of data they own.
- If the data center has more nodes than can be displayed on the screen, it will be represented as a condensed ring view. This typically occurs when the data center has hundreds of nodes.
- The color of each node 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). A summary of the cluster's health is located within the ring.
- The size of each node represents its data size relative to all other nodes in the cluster.


cluster1: Nodes
RING
LIST
Health
Normal
Medium
High
Down
5
2
1
0
5
2
1
0

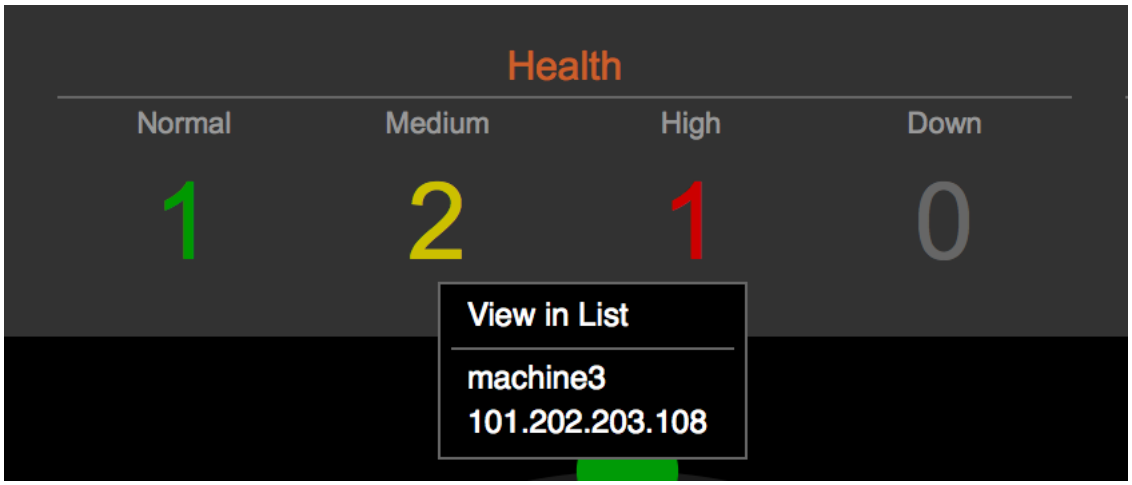
Health summary

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. You can easily identify potential problem nodes, as well as whether any multiple nodes within a single replica set are having issues.



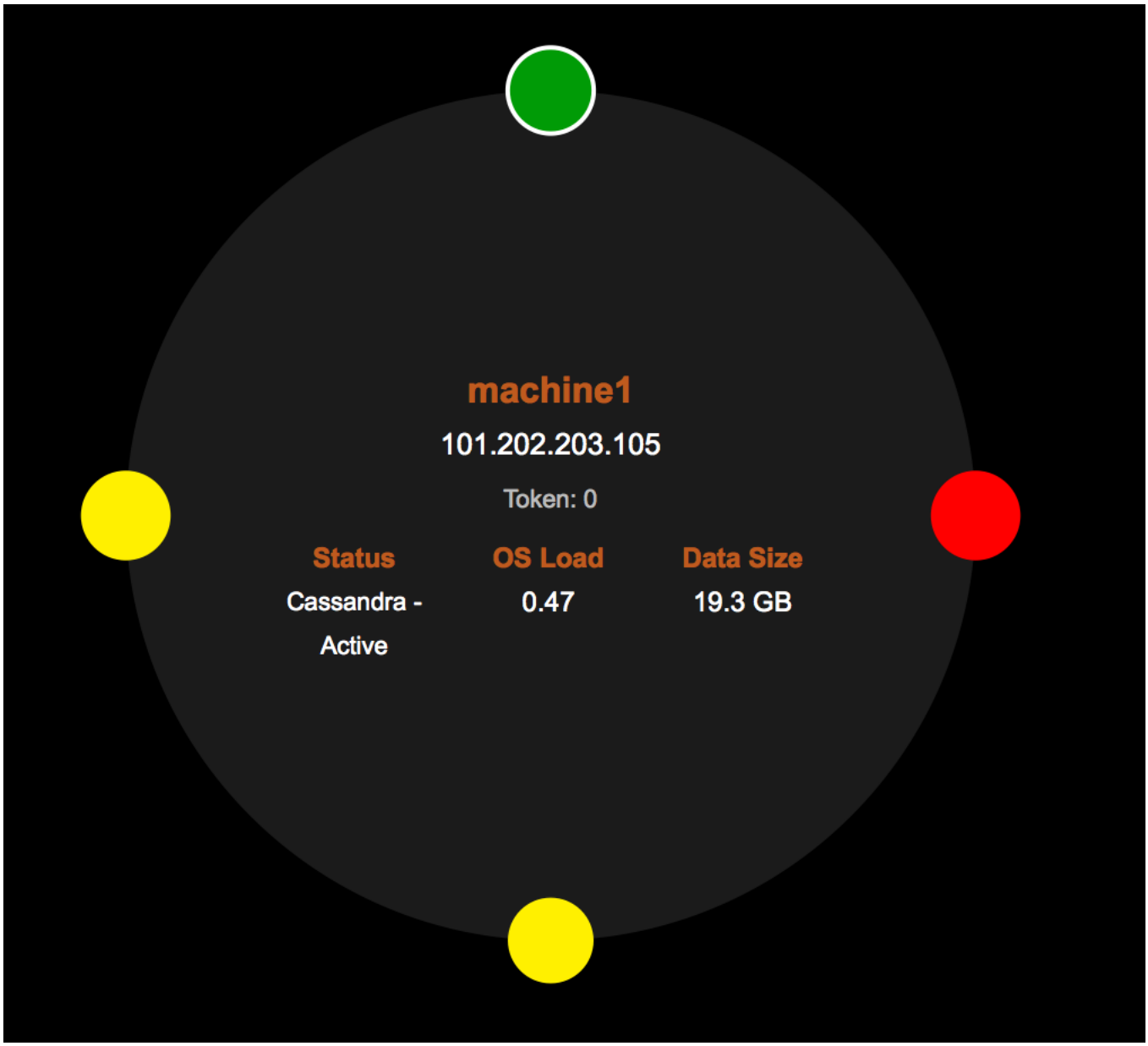
Clicking a total in the health summary presents a list of nodes included in the total.



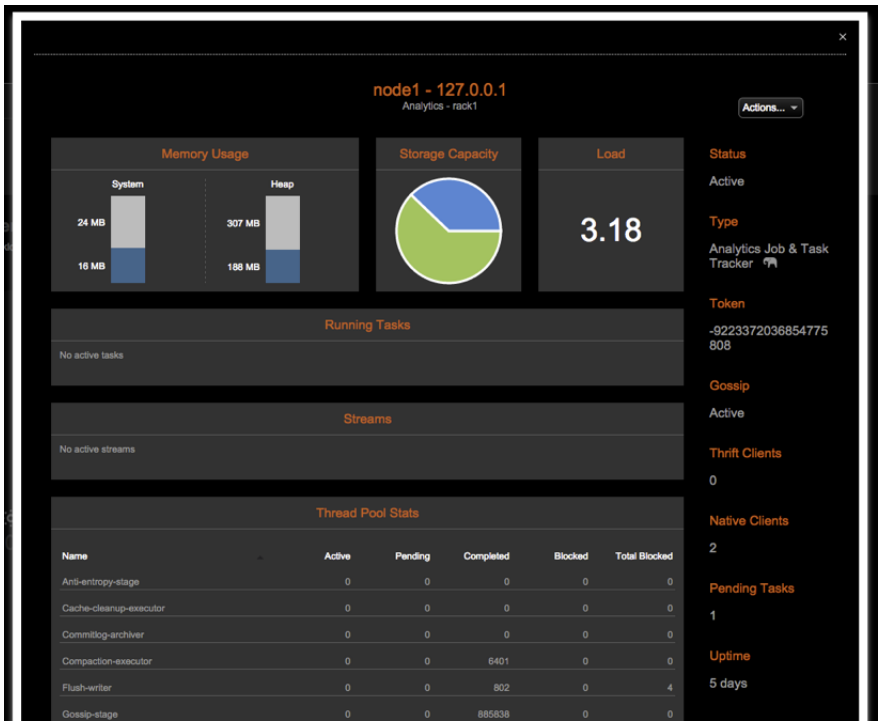
A health summary of each data center is located within the ring.

Node details

Hovering over a node displays some basic details about that node. These details are updated in realtime.

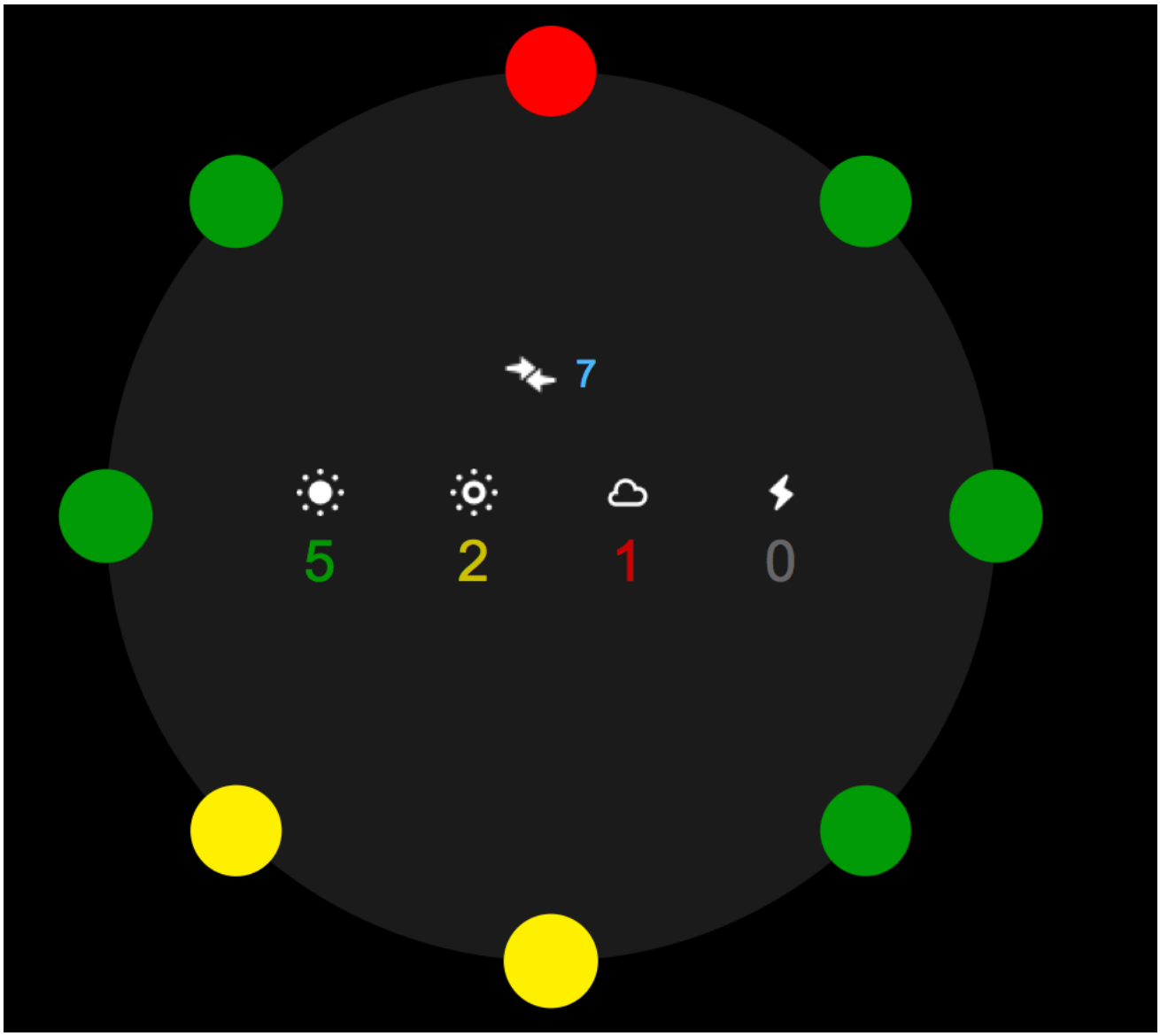


Clicking a node reveals a dialog displaying more information so you can run various operations on the node.

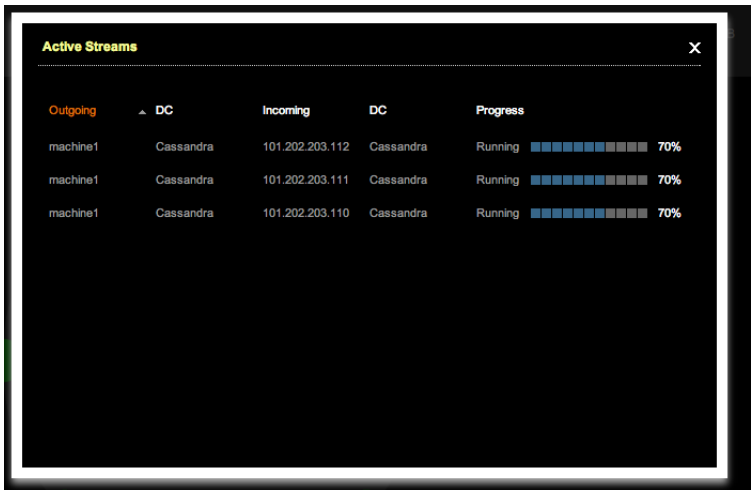


Streams

Whenever any nodes in that data center are streaming data to or from another node within the cluster, an icon (indicated by the arrows) is displayed inside of the ring. The icon appearance distinguishes between streams contained within that datacenter ("intra-dc streams") and streams between data centers ("inter-dc streams").



Clicking on the icon in any data center opens the Streams dialog, which gives details on all of the active streams in the cluster.



Node positioning

The goal of positioning nodes in a ring is to visually represent whether a datacenter is balanced or not (that is, data is more likely to be evenly distributed across nodes). In a healthy ring, nodes are spread out evenly around the ring.

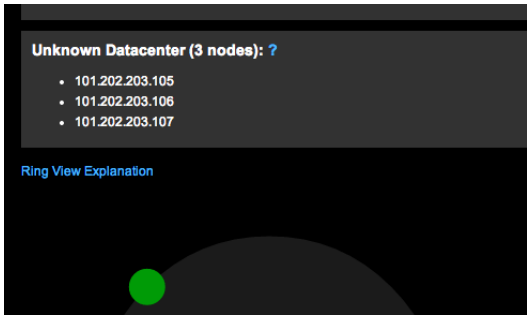
When a cluster uses `RandomPartitioner` or `Murmur3Partitioner` for its snitch, its nodes are positioned around the ring according to their assigned token, but there are some cases where positioning by token does not make sense:

- If vnodes are enabled, 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 doesn't 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 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

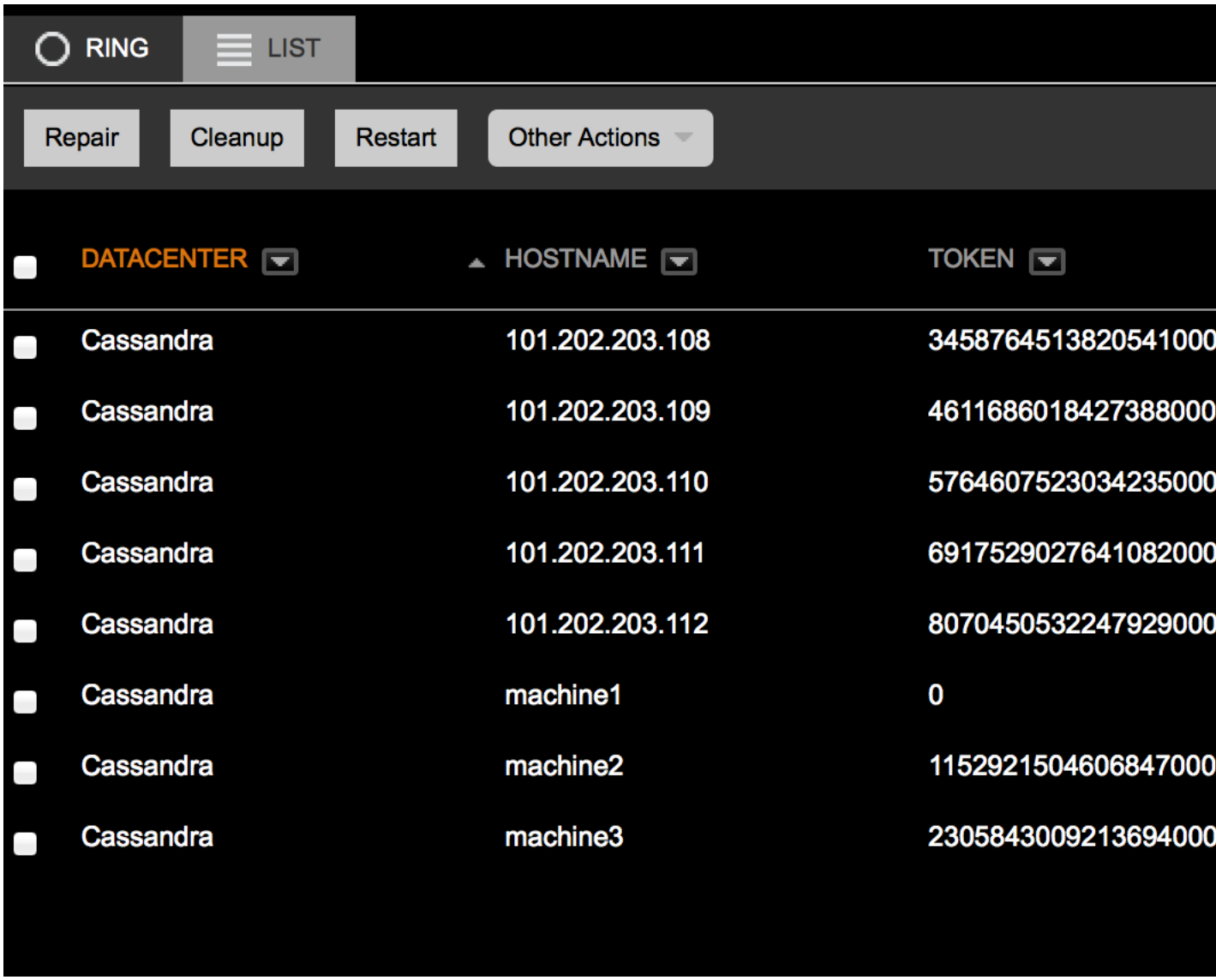
Rings are displayed by datacenter. This information is gathered from the agent that runs on each node. If a new data center has just been added, or if new nodes have been added to the cluster, it is displayed in a special list above all rings until OpsCenter fully processes the new cluster information.



List View

List View is an alternative to Ring View. List View provides faster access to data and more flexibility when viewing data. All data is updated in realtime.

List View is an alternative to Ring View. List View provides faster access to data and more flexibility when viewing data. All data is updated in realtime.

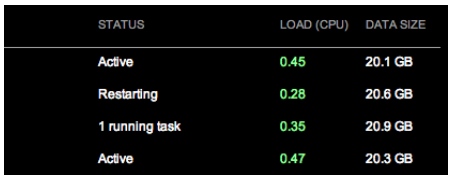


By selecting any of the check boxes next to each node, you can run any operation on any number of nodes in a cluster. For more information, see Node management operations.

Columns can be sorted in ascending or descending order by clicking on the column label to view which nodes have the most data, the highest CPU load, and so forth.

The Status column displays whether a node is:

- up or down
- in a special mode (for example, joining, draining, or moving)
- running any tasks, such as compactions



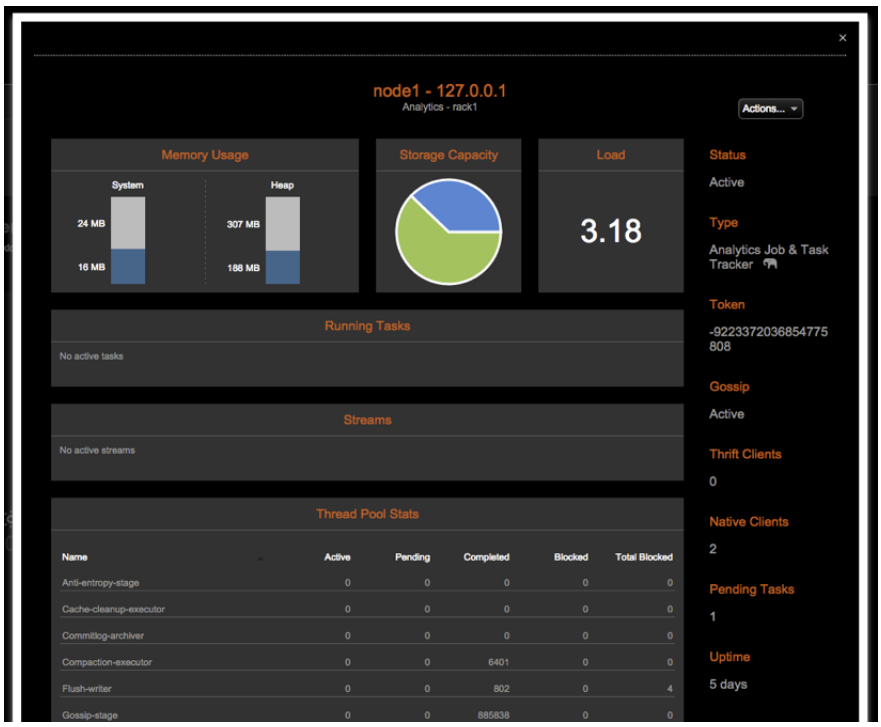
Filtering nodes

You can filter the list of nodes by data center, host name, token, status, and load columns. Filtering nodes reduces the number of nodes displayed in the list, which is useful when working with large clusters that contain hundreds of nodes.

Viewing node details

View node details such as Status, Capacity, Uptime, and Memory Usage including In Memory. To view details about a single node, click the row for the node in the List View.

A Node dialog appears.



Node management operations

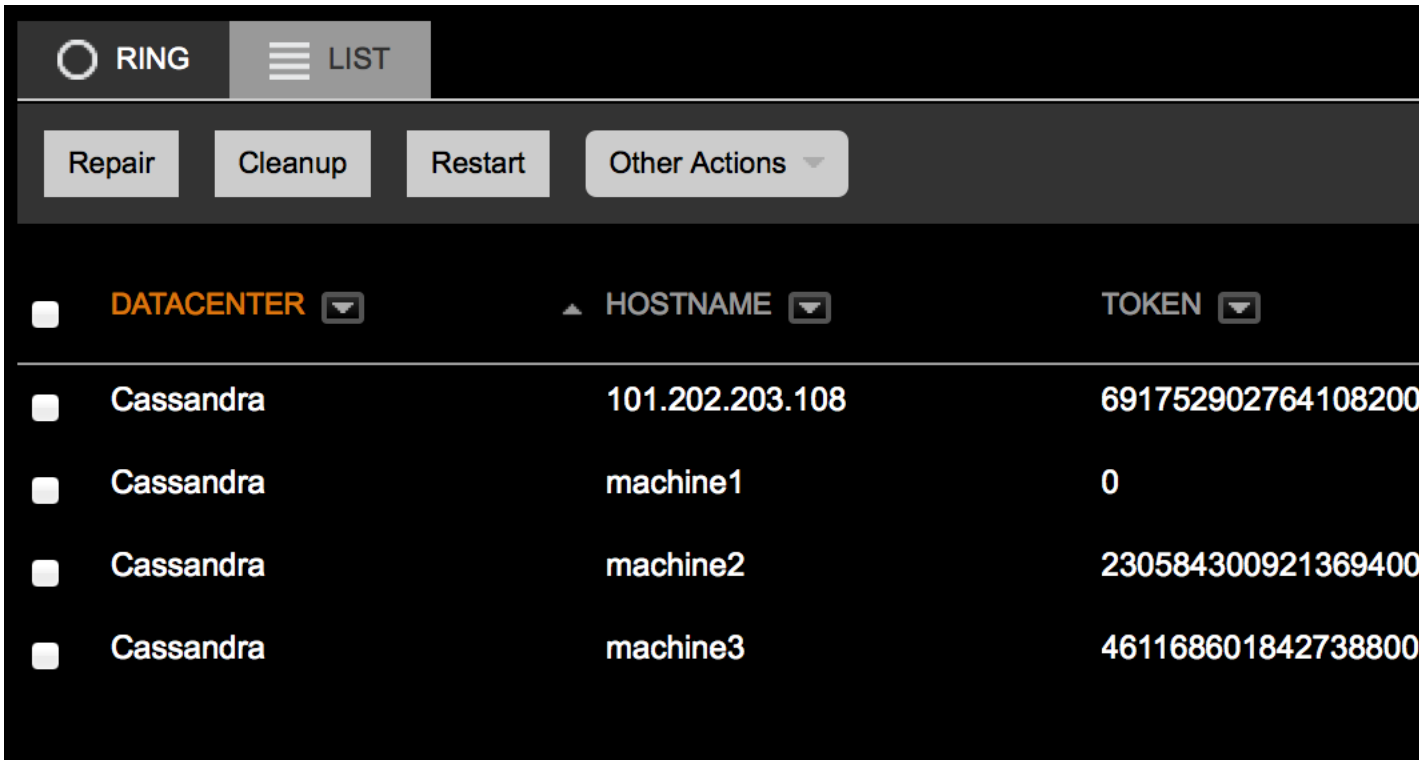
Use OpsCenter to run operations on nodes in an easy to use, visual way that takes the guesswork out of properly managing nodes in a cluster. Managing multiple nodes is an enterprise-only feature.

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.

Note: Managing multiple nodes is an enterprise-only feature.

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 or List View and choose an action from the **Actions** menu:

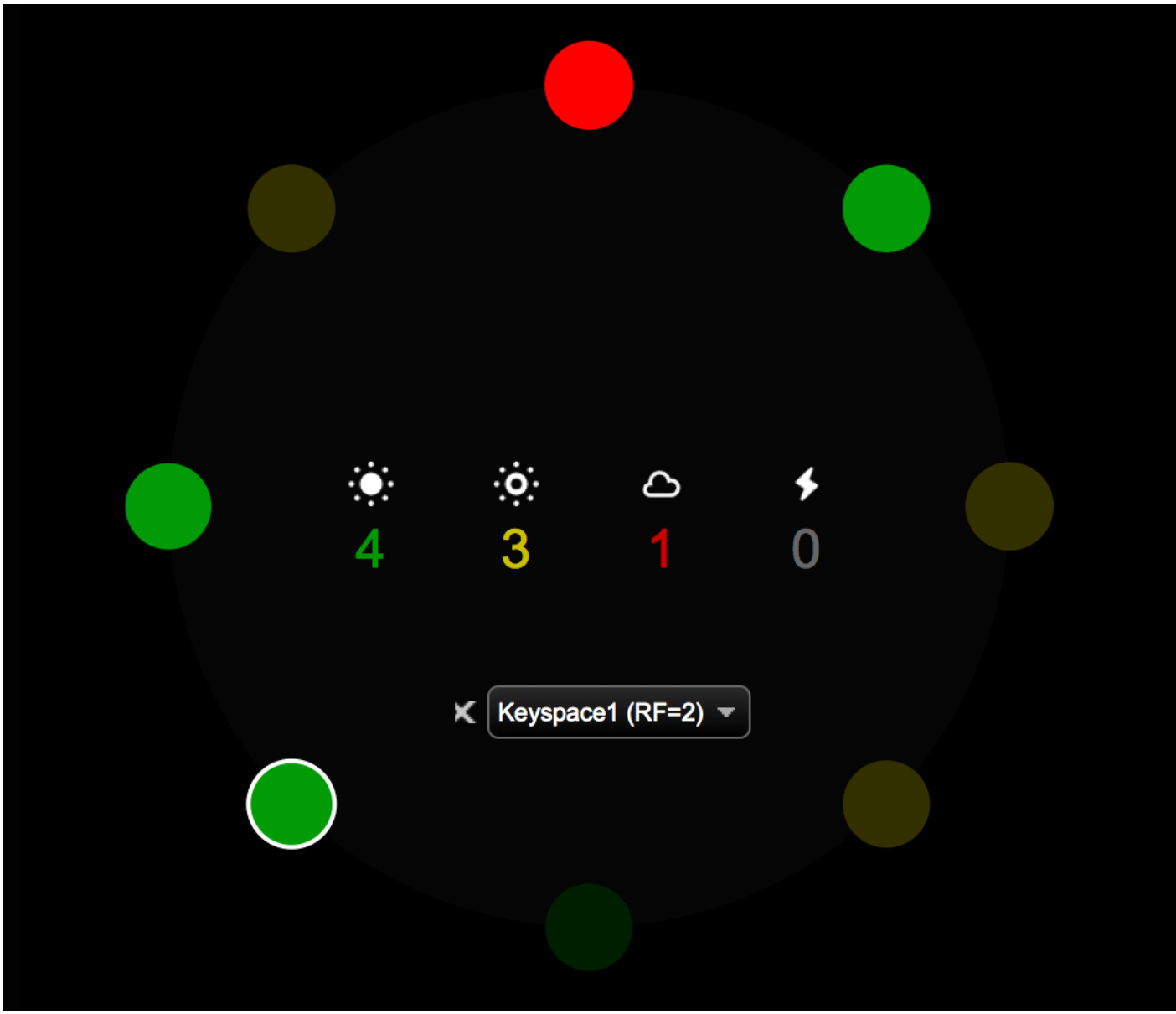
Operations details

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.



Configure (single node only)

Modify settings specified in `cassandra.yaml` for the selected node.

Start/Stop

Starts or stops the DSE or Cassandra process on the node.

Restart

Restarts the DSE or Cassandra 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

Performs a major compaction, which is not a recommended procedure in most Cassandra 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 column family 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.

Configuring a node

Manage cassandra.yaml for a single node.

About this task

Manage `cassandra.yaml` for a single node.

Procedure

1. In the left navigation pane, click **Cluster** > **Nodes** > **List View**.
2. Click the node to view its details.
   The node details dialog appears.
3. Click **Actions** > **Configure**.
   The Edit Node Config - *node name or alias* dialog appears.
4. Click the **Cassandra** tab.
5. Edit the values for any of the options. Click the scroll bars to access all of the available options.

   For option descriptions, see the documentation for the Cassandra version that the cluster or node is running, such as `cassandra.yaml`.
6. When you are done, click **Save Configuration**.
   The configuration is saved and you are prompted to restart the Cassandra for the node.
7. Indicate whether you want to drain the node before restarting the service and click restart.

Configure an alias for a node

Configure an alias for a node to display throughout OpsCenter.

About this task

Configure an alias to display for a node throughout OpsCenter. An alias replaces the IP address or hostname displayed for a node. Give each node a meaningful and memorable name for your environment.

Procedure

1. Open the address.yaml file for editing.
2. Add the following option to the file:

```
alias: nodeName1
```

3. Restart the agent.
4. Repeat the above steps for each node.

Monitoring in-memory usage

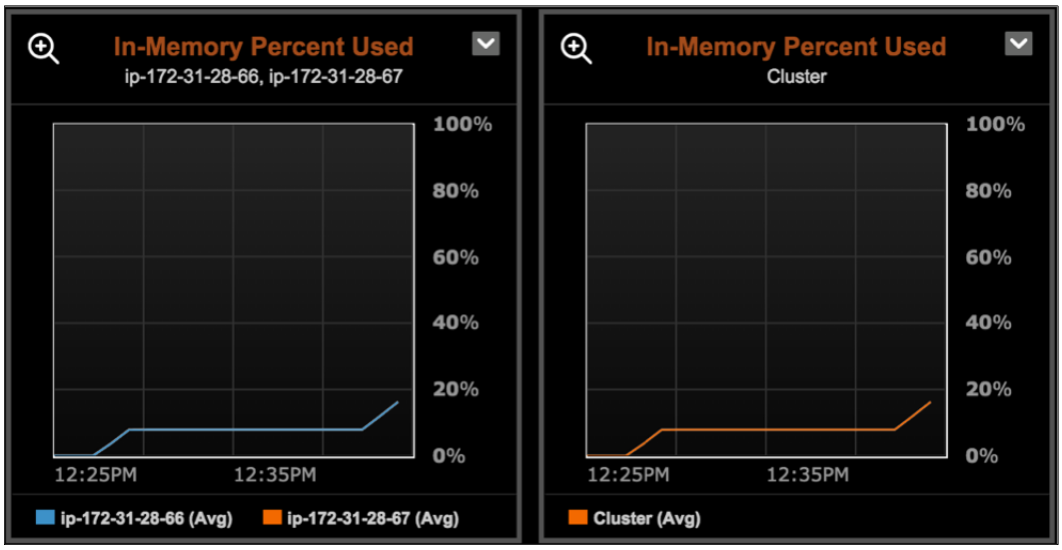
Monitor in-memory usage from within OpsCenter. The In-Memory option is a DataStax Enterprise feature only.

About this task

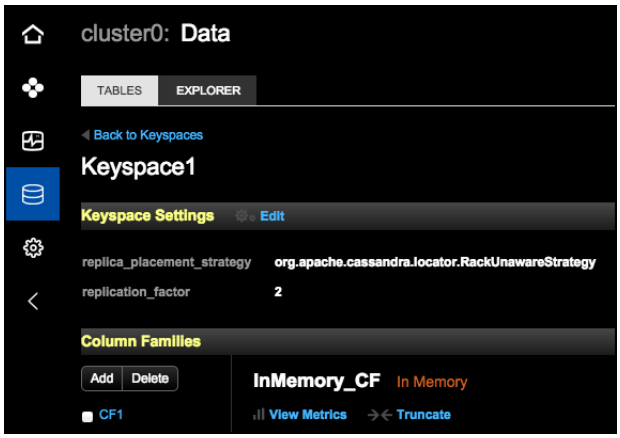
Monitor in-memory usage from within OpsCenter. The In-Memory option is a DataStax Enterprise feature only. More information about creating or altering a table to use in-memory is available in the DataStax Enterprise documentation DSE In-Memory section.

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

- The In-Memory Percent Used alert is available to configure for DSE 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 in the dashboard of OpsCenter version 5.1.2+.



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** > **Column Families (tables)**:

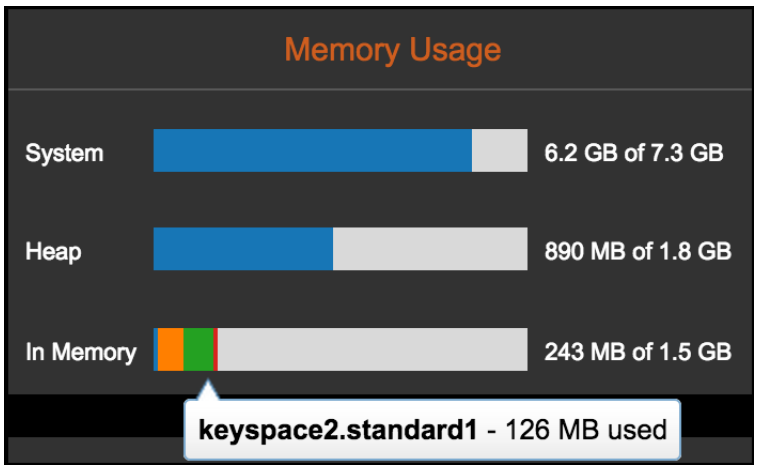


About this task

To view the in-memory usage of a node:

Procedure

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+):
   - For 4.7 and beyond, 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 prior to 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. Alerting for in-memory usage is a DataStax Enterprise feature only.

About this task

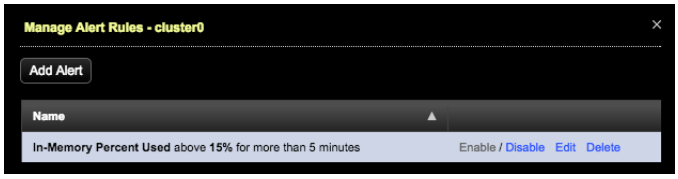
Configure an alert to monitor in-memory usage. Alerting for in-memory usage is a DataStax Enterprise feature only.

Procedure

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 and click **Save Alert**.



The configured In-Memory Percent Used alert appears in the Manage Alert Rules dialog.

Viewing the Spark Console

Access the Spark Console for a Spark Master node from within OpsCenter.

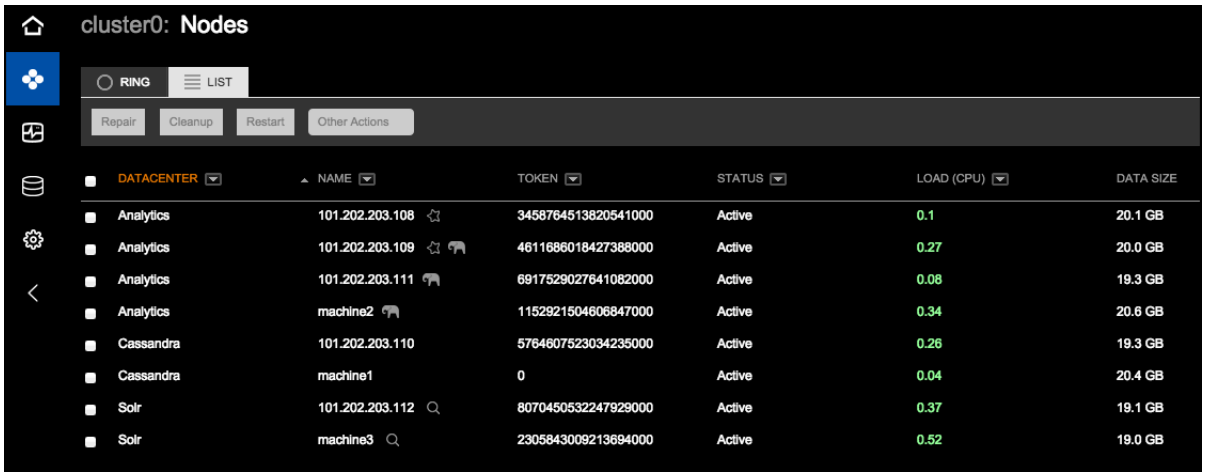
About this task

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.

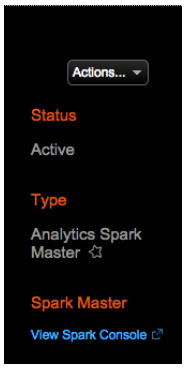
Procedure

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 for faster identification.
3. Click the **View Spark Console** link.



The Spark Console launches within another browser window.

Restarting a node

Restart the Cassandra or DataStax Enterprise service on any node.

About this task

Restart the Cassandra or DataStax Enterprise service on any node. Restart a node from the Nodes List or from the Node Details dialog using the Actions menu.

Procedure

1. Click **Nodes** in the left navigation pane.
2. Select the check boxes for the nodes you want to restart.
3. Click **Other Actions** > **Restart**.
   The Restart Cassandra or Restart DSE dialog prompts you to confirm the restart.
4. Optional: Indicate whether to drain the node before stopping and restarting.
5. Click **Restart Cassandra** or **Restart DSE** as approproriate.

Cluster administration

OpsCenter manages multiple DataStax Enterprise or Apache Cassandra clusters with a single install of the central opscenterd server.

OpsCenter manages multiple DataStax Enterprise or Apache Cassandra clusters with a single install of the central `opscenterd` server.

OpsCenter support of Cassandra / DSE versions

See the Upgrade Guide for information about the officially supported and tested versions.

Generating a PDF report

Generates a PDF report for a monitored cluster. The report shows the version of OpsCenter, number of clusters and nodes being monitored, gigabytes of storage used, name of the cluster, and information about nodes in the cluster.

To generate a PDF 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 monitored, 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 software version
- DataStax software version
- Memory usage
- Operating system running on the node

You can save or print the PDF report.

Collecting and downloading diagnostic data

Download diagnostics information about the OpsCenter daemon and all the nodes in a specific cluster. Attach the diagnostic data to support tickets to facilitate resolving any issues. Downloading diagnostic data is a DSE feature.

About this task

Download a compressed tarball that contains diagnostic information about the OpsCenter daemon and all the nodes in a specific cluster. To download the tarball:

Procedure

1. Click **Help** > **Diagnostics**.
   A message appears: "Collecting cluster data; please wait, this may take a few minutes..."
2. 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. You can attach the diagnostic data to support tickets.

Configuring a cluster

Manage cassandra.yaml for an entire cluster from OpsCenter.

About this task

Manage `cassandra.yaml` for an entire cluster from OpsCenter. If a cluster exists in multiple data centers, you can configure `cassandra.yaml` for a single datacenter or for all nodes in a cluster.

Note: When configuring a cluster using a datacenter-aware snitch (EC2, EC2MR, GPFS, PFS, GCE), the topology of nodes cannot be configured through OpsCenter. Forthcoming versions

of OpsCenter will provide more robust support for additional snitches. All snitches available in Cassandra are described in the Cassandra documentation.

Follow the steps below to configure `cassandra.yaml` for all nodes in a cluster.

Procedure

1. Click **Nodes** in the left navigation pane.
2. Click **Cluster Actions** > **Configure**.
   The Edit Cluster Config - All Nodes dialog appears.
3. Edit the values for any of the options. Click the scroll bars to access all of the available options.

   For option descriptions, see the documentation for the Cassandra version that the cluster or node is running, such as `cassandra.yaml`.
4. When you are done, click **Save Configuration**.
   You are prompted to proceed with a rolling restart of the cluster.

Adding a node to a cluster

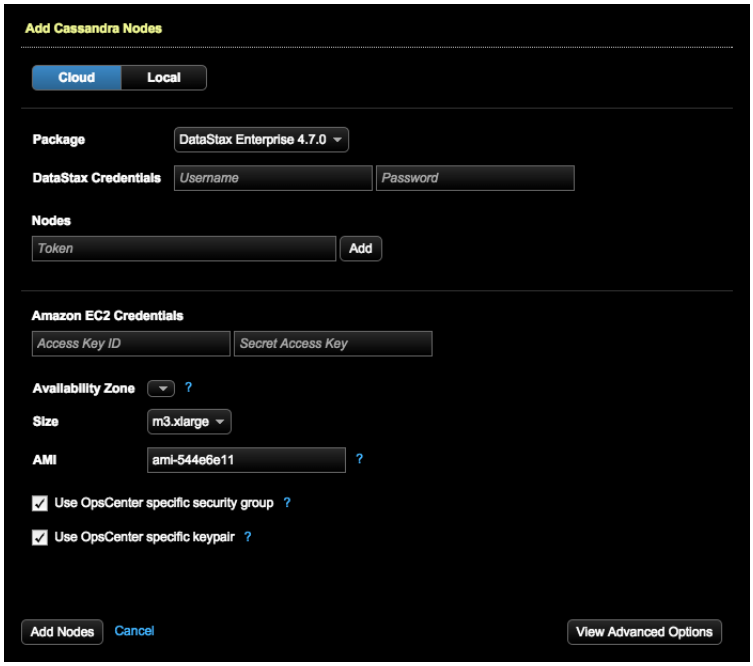
Add Cassandra, Solr, Spark, or Hadoop nodes to a cluster locally or in the cloud.

Follow these steps to add Cassandra, Solr, Spark, or Hadoop nodes to a cluster locally or in the cloud. After adding a node to a cluster, rebalancing the cluster is recommended.

Note: When adding a node to a cluster, only the following snitches are supported: SimpleSnitch, DseSimpleSnitch, and RackInferringSnitch. Forthcoming versions of OpsCenter will provide more robust support for additional snitches. All snitches available in Cassandra are described in the Cassandra documentation.

Procedure

1. Click ***cluster name*** > **Nodes** in the left navigation pane.
2. Click **Cluster Actions** > **Add Node** > **Add *node type***.
   The Add *Type* Nodes dialog appears.



3. Click **Cloud** or **Local**.

The **Cloud** and **Local** buttons appear only if OpsCenter is running in the cloud.

4. Enter the following:

Option	Value
Package	The version of DSE to install on the node.
DataStax credentials	The username and password you received when registering to Download DSE.
Nodes	The hostname or IP address, token, and software to install on the node (from Cassandra, Solr, Spark, and Hadoop). You can add more than one node by clicking Add.
Node credentials (sudo) (Local only)	The username and password to authenticate on the host. (Optional) The private SSH key to use for authentication.
Amazon EC2 Credentials (Cloud only)	The <*access-key-id*> and <*secret-access-key*> to use to authenticate on AWS EC2.
Availability Zone (Cloud only)	Which availability zone to use. The menu populates after entering your EC2 credentials.
Size (Cloud only)	Which size image to use.
AMI (Cloud only)	Which image to use.
Use OpsCenter specific security group (Cloud only)	Determines whether OpsCenter creates its own specific security group (default), or allows you to select one that is available using your EC2 credentials.
Use OpsCenter specific keypair (Cloud only)	Determines whether OpsCenter creates its own specific keypair (default), or allows you to select one that is available using your EC2 credentials.
View Advanced Options	Access `cassandra.yaml` for configuring a node.

5. Click **Add Nodes**.

Removing a cluster

Removes the cluster from OpsCenter from the OpsCenter UI. Deleting a cluster from OpsCenter does not delete the cluster itself.

Procedure

1. Click **Settings** in the upper right and then **Cluster Connections**
2. Click **Delete Cluster**.

   When you delete a cluster, any EC2 nodes are not deleted.
3. Confirm that you want to remove the cluster from the OpsCenter UI by clicking **Delete**.

Rebalancing a cluster

Cluster rebalancing ensures that each node in a Cassandra cluster manages an equal amount of data. Rebalancing a cluster is a DataStax Enterprise feature only.

About this task

Cluster rebalancing ensures that each node in a Cassandra cluster manages an equal amount of data. Rebalancing a cluster is an enterprise-only feature. 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.

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.

Procedure

1. Click **Cluster Actions** > **Rebalance Cluster**.
   OpsCenter checks if the token ranges are evenly distributed across the nodes in the cluster.
2. If the cluster is already balanced, a message indicates rebalancing is not necessary at this time. If the cluster requires rebalancing, 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.
3. If you cancel a rebalance operation before all nodes are moved, click **Rebalance Cluster** again to resume.

Restarting a cluster

Restart an entire cluster. Each node in the cluster restarts in a sequential rolling fashion after a sleep time elapses.

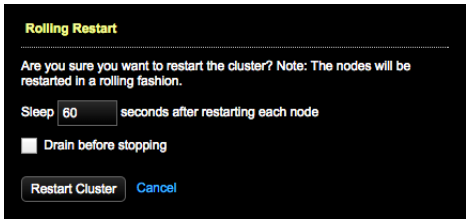
About this task

Restart an entire cluster. 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.

Procedure

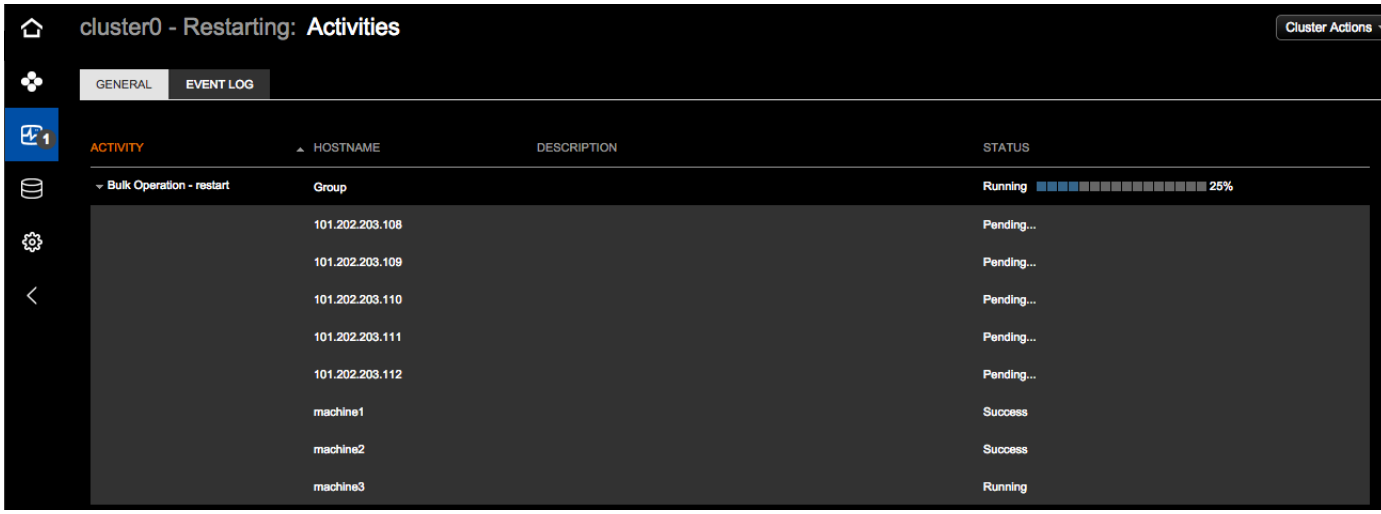
1. Click **Restart** from the **Cluster Actions** menu.

   The Rolling Restart dialog appears.



2. Set the amount of time to wait after restarting each node.
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 restarted successfully message indicates when the restart cluster operation has completed.



Changing the display name of a cluster

Change the display name of a cluster as it appears throughout the OpsCenter UI.

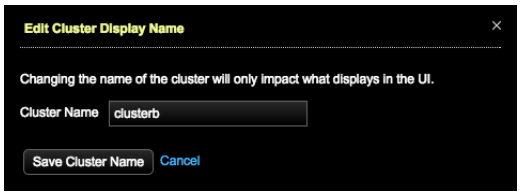
About this task

Change the display name of a cluster as it displays in the OpsCenter UI. 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 Dev, Test, and Live environments. Revert to the actual cluster name at any time.

Procedure

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.

3. Click **Edit Display Name**.
   The Edit Display Name dialog appears.
4. Enter the name you want displayed for the cluster in the **Cluster Name** box.



5. Click **Save Cluster Name**.
   The cluster display name changes throughout the OpsCenter UI. To revert to the actual cluster name, clear the Cluster Name and save again.

Performance metrics

Monitor performance metrics in the OpsCenter Dashboard. Real-time and historical performance metrics are available at different granularities: cluster-wide, per node, or per table (column family).

Monitor performance metrics in the OpsCenter Dashboard. Real-time and historical performance metrics are available at different granularities: cluster-wide, per node, or per table (column family).

Viewing performance metrics

View performance metrics in graph views from the dashboard.

Select **Dashboard** to view these types of metrics:

- Cluster Performance Metrics
- Pending Task Metrics
- Table (Column Family) Metrics

When you add a graph, you 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, and choose the default view of graphs. Click the drop down arrow next to the view name at the top of the Dashboard area. The **Make Default**, **Delete**, and **Clone** menu options are available.

Creating and editing performance graphs

Add and edit performance graphs on the Dashboard.

About this task

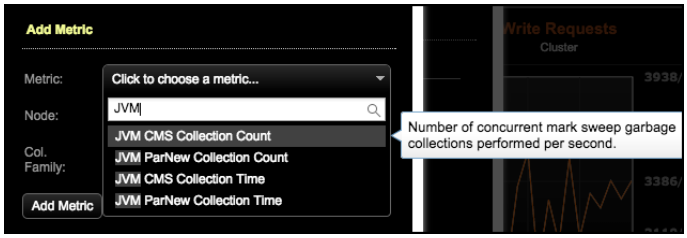
Graphs can be added containing multiple metrics provided the metrics use the same unit. For example, a graph can contain multiple metrics showing utilization as a percentage, like CPU and disk utilization. Other metrics such as write or read requests for a cluster or the operating system load for a node cannot be added to the utilization graph. Metrics can be added to a graph for a cluster or for one or more nodes.

There are also widgets that display information on Alerts, Cluster Health, and Storage Capacity.

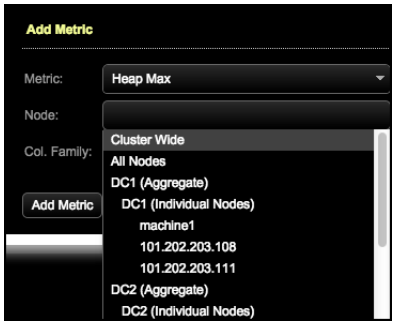
Procedure

1. Click **Dashboard** in the left pane.
2. Click **Add Graph**.
3. In the Add Metric dialog, select the metric you want to add from the **Metric** list.

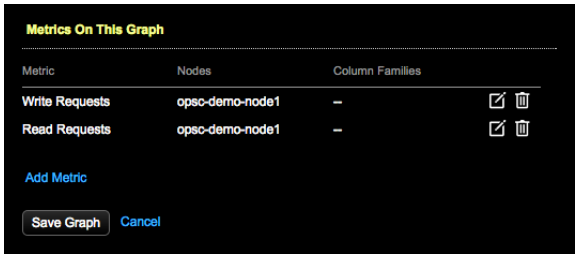
   To search for a metric, begin typing the metric you want to search for and matching metrics populate in the list. To view its description, hover over a metric.



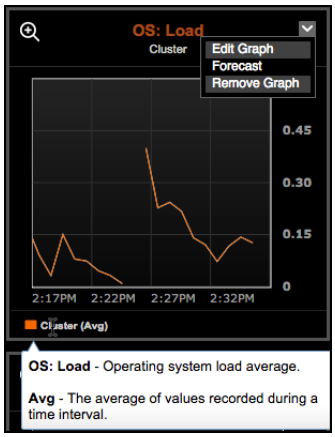
4. Select the nodes to monitor in the **Node** list. Select an individual node, multiple nodes, all nodes within a data center, all nodes or cluster wide as appropriate for the metric graph. To make multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux) to keep the list open for multiple selections.



5. Optional: To specify particular column families (tables), click **Col. Family**.
6. Click **Add Metric**.
7. To add additional metrics that are measured using the same unit, click **Add Metric** in the Metrics On This Graph dialog. To edit the details of a metric, 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**.



Hover over the metric in the graph legend to view its descriptions.

10. To enable or disable the Alerts, Cluster Health, and Storage Capacity widgets, click **Add Widget** and select the widget you want to enable or disable.

Grouping performance metrics

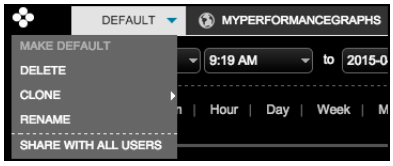
Organize groups of performance graphs with custom named presets. Clone, rename, share, or delete performance metrics views on the Dashboard.

About this task

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 for each preset view as you prefer.

Procedure

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 view, click **Clone**. For multiple clusters, select either **To This Cluster** to create a new view for the current cluster, or **To Different Cluster** to clone the view to a different cluster. In the Save Preset dialog, enter a name for the preset and click **Save**.
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.

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.

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 column families can further pinpoint the source of performance issues with an application.

Cassandra JVM memory usage

The average amount of Java heap memory (in megabytes) being used by Cassandra processes. Cassandra opens the JVM with a heap size that is half of available system memory by default, which still allows an optimal amount of memory remaining for the OS disk cache. You may need to increase the amount of heap memory if you have increased column family memtable or cache sizes and are getting out-of-memory errors. If you monitor Cassandra Java processes with an OS tool such as top, you may notice the total amount of memory in use exceeds the maximum amount specified for the Java heap. This is because Java allocates memory for other things besides the heap. It is not unusual for the total memory consumption of the JVM to exceed the maximum value of heap memory.

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

The *average* 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 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

The response time (in milliseconds) for successful read requests. The time period starts when a node receives a client read request, and ends when the node responds back to the client. Optimal or acceptable levels of read latency vary widely according to your hardware, your network, and the nature of your application read patterns. For example, the use of secondary indexes, the size of the data being requested, and the consistency level required by the client can all impact read latency. An increase in read latency can signal I/O contention. Reads can slow down when rows are fragmented across many SSTables and compaction cannot keep up with the write load.

JVM CMS Collection Count

The number of concurrent mark-sweep (CMS) garbage collections performed by the JVM per second. These are large, resource-intensive collections. Typically, the collections occur every 5 to 30 seconds.

JVM CMS Collection Time

The time spent collecting CMS garbage in milliseconds per second (ms/sec).

Note: A ms/sec unit defines the number of milliseconds for garbage collection for each second that passes. For example, the percentage of time spent on garbage collection in one millisecond (.001 sec) is 0.1%.

JVM ParNew Collection Count

The number of parallel new-generation garbage collections performed by the JVM per second. These are small and not resource intensive. Normally, these collections occur several times per second under load.

JVM ParNew Collection Time

The time spent performing ParNew garbage collections in ms/sec. The rest of the JVM is paused during ParNew garbage collection. A serious performance hit can result from spending a significant fraction of time on ParNew collections.

Data Size

The size of column family data (in gigabytes) that has been loaded/inserted into Cassandra, including any storage overhead and system metadata. DataStax recommends that data size not exceed 70 percent of total disk capacity to allow free space for maintenance operations such as compaction and repair.

Total bytes compacted

The number of SSTable data compacted in bytes per second.

Total compactions

The number of compactions (minor or major) performed per second.

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.

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.

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).

Flush Sorter Tasks Pending

The flush sorter process performs the first step in the overall process of flushing memtables to disk as SSTables.

Memtable Post Flushers Pending

Number of 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.

Write Requests Pending

Number of write requests received by the cluster and waiting to be handled.

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.

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 column families where your requirements can tolerate a certain degree of stale data, you can lower the value of the column family 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 column families.

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 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.

Manual repair tasks pending

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. Repair is a resource-intensive operation that is executed in stages: comparing data between replicas, sending changed rows to the replicas that need to be made consistent, deleting expired tombstones, and rebuilding row indexes and bloom filters. Tracking the state of this metric can help you determine the progress of a repair operation. 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.

Gossip tasks pending

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). A value greater than 0 indicates possible network problems (see network traffic for indications of network health).

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.

Internal Responses Pending

Number of pending tasks from internal tasks, such as nodes joining and leaving the cluster.

Migrations pending

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.

Miscellaneous tasks pending

The number of pending tasks from other miscellaneous operations that are not run frequently.

Request Responses Pending

Number of pending callbacks to execute after a task on a remote node completes.

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.

Column family performance metrics

Column family 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 column family metrics can provide a more granular level of detail.

Column family 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 column family 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 column family at a time. Otherwise, all column family metrics are available for specific column families as well as for all column families on a node. In addition to monitoring read latency, write latency and load on a column family, you should also monitor the hit rates on the key and row caches for column families 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 column family (or counts for all families) can help with compaction tuning.

OpsCenter has been optimized to efficiently handle thousands of column families. If a column family experiences a dramatic dip in performance, check the Pending Tasks metrics for a backup in queued operations.

Column Family metrics are prefaced with CF.

CF: Local Writes

The write load on a column family measured in requests per second. This metric includes all writes to a given column family, including write requests forwarded from other nodes. This metric can be useful for tracking usage patterns of an application.

CF: Local Write Latency

The response time in milliseconds for successful write requests on a column family. The time period starts when nodes receive a write request, and ends when nodes respond. Optimal or acceptable levels of write latency vary widely according to your hardware, your network, and the nature of your write load. For example, the performance for a write load consisting largely of granular data at low consistency levels would be evaluated differently from a load of large strings written at high consistency levels.

CF: Write Latency (Stacked)

The min, median, max, 90th, and 99th percentile of the response times to write data to a table's memtable and append to the commitlog. The elapsed time from when the replica receives the request from a coordinator and returns a response.

CF: Local Reads

The read load on a column family measured in requests per second. This metric includes all reads to a given column family, including read requests forwarded from other nodes. This metric can be useful for tracking usage patterns of your application.

CF: Local Read Latency

The response time in milliseconds for successful reads on a column family. The time period starts when a node receives a read request, and ends when the node responds. Optimal or acceptable levels of read latency vary widely according to your hardware, your network, and the nature of your application read patterns. For example, the use of secondary indexes, the size of the data being requested, and the consistency level required by the client can all impact read latency. An increase in read latency can signal I/O contention. Reads can slow down when rows are fragmented across many SSTables and compaction cannot keep up with the write load.

Read Latency (Stacked)

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.

CF: Live Disk Used

Disk space used by live SSTables. There might be obsolete SSTables not included.

CF: Total Disk Used

Disk space used by a table by SSTables, including obsolete ones waiting to be garbage collected.

CF: SSTable Count

The current number of SSTables for a column family. When column family memtables are persisted to disk as SSTables, this metric increases to the configured maximum before the compaction cycle is repeated. Using this metric together with SSTable size, you can monitor the current state of compaction for a given column family. Viewing these patterns can be helpful if you are considering reconfiguring compaction settings to mitigate I/O contention.

CF: SSTables per Read (Stacked)

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

CF: Pending Reads/Writes

The number of pending reads and writes on a column family. 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.

CF: Bloom Filter Space Used

The size of the bloom filter files on disk. This grows based on the number of rows in a column family and is tunable through the per-CF attribute, `bloom_filter_fp_chance`; increasing the value of this attribute shrinks the bloom filters at the expense of a higher number of false positives. Cassandra reads the bloom filter files and stores them on the heap, so large bloom filters can be expensive in terms of memory consumption.

Note: Bloom filters are used to avoid going to disk to try to read rows that don't actually exist.

CF: Bloom Filter False Positives

The number of false positives, which occur when the bloom filter said the row existed, but it actually did not exist in absolute numbers.

CF: Bloom Filter False Positive Ratio

Percentage of bloom filter lookups that resulted in a false positive.

Note: The Bloom Filter False Positive Ratio should normally be at or below .01. A higher reading indicates that the bloom filter is likely too small.

CF: Bloom Filter Off Heap

Total off heap memory used by bloom filters from all live SSTables in a table.

CF: Index Summary Off Heap

Total off heap memory used by the index summary of all live SSTables in a table.

CF: Compression Metadata Off Heap

Total off heap memory used by the compression metadata of all live SSTables in a table.

CF: Memtable Off Heap

Off heap memory used by a table's current memtable.

CF: Total Memtable Size

An estimate of the space used in memory (including JVM overhead) for all memtables. This includes ones that are currently being flushed and related secondary indexes.

CF: Key Cache Requests

The total number of read requests on the row key cache.

CF: Key Cache Hits

The number of read requests that resulted in the requested row key being found in the key cache.

CF: Key Cache Hit Rate

The percentage of cache requests that resulted in a cache hit that indicates the effectiveness of the key cache for a given column family. The key cache is used to find the exact location of a row on disk. If a row is not in the key cache, a read operation will populate the key cache after accessing the row on disk so subsequent reads of the row can benefit. Each hit on a key cache can save one disk seek per SSTable. If the hits line tracks close to the requests line, the column family is benefiting from caching. If the hits fall far below the request rate, this suggests that you could take actions to improve the performance benefit provided by the key cache, such as adjusting the number of keys cached.

CF: Row Cache Requests

The total number of read requests on the row cache. This metric is only meaningful for column families with row caching configured (row caching is not enabled by default).

CF: Row Cache Hits

The number of read requests that resulted in the read being satisfied from the row cache. This metric is only meaningful for column families with row caching configured (row caching is not enabled by default).

CF: Row Cache Hit Rate

The percentage of cache requests that resulted in a cache hit that indicates the effectiveness of the row cache for a given column family. This metric is only meaningful for column families with row caching configured (row caching is not enabled by default). The graph tracks the number of read requests in relationship to the number of row cache hits. If the hits line tracks close to the requests line, the column family is benefiting from caching. If the hits fall far below the request rate, this suggests that you could take actions to improve the performance benefit provided by the row cache, such as adjusting the number of rows cached or modifying your data model to isolate high-demand rows.

CF: SSTable Size

The current size of the SSTables for a column family. It is expected that SSTable size will grow over time with your write load, as compaction processes continue doubling the size of SSTables. Using this metric together with SSTable count, you can monitor the current state of compaction for a given column family.

Viewing these patterns can be helpful if you are considering reconfiguring compaction settings to mitigate I/O contention

Search performance metrics

Metrics for monitoring DSE Search performance.

Metrics for monitoring DSE Search 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.

Operating system performance metrics

Tracking operating system metrics on Cassandra nodes to watch for disk I/O, network, memory and CPU utilization trends helps identify and troubleshoot hardware-related performance problems.

As with any database system, Cassandra performance greatly depends on underlying systems on which it is running. Monitoring Cassandra nodes for increasing disk and CPU utilization can help identify and remedy issues before performance degrades to unacceptable levels. The graphs in OpsCenter provide a quick way to view variations in OS metrics at a glance, and drill-down for specific data points. Especially in systems with heavy write loads, monitoring disk space is also important because it allows for advanced expansion planning while there is still adequate capacity to handle expansion and rebalancing operations.

System metrics are prefaced with OS.

OS: Memory

Shows memory usage metrics in megabytes.

- Linux - Shows how much total system memory is currently used, cached, buffered or free.
- Windows - Shows the available physical memory, the cached operating system code, and the allocated pool-paged-resident and pool-nonpaged memory.
- Mac OS X - Shows free and used system memory.

OS: CPU

Shows average percentages for CPU utilization metrics, which is the percentage of time the CPU was idle subtracted from 100 percent. CPU metrics can be useful for determining the origin of CPU performance reduction.

- Linux- Shows how much time the CPU devotes to system and user processes, to tasks stolen by virtual operating systems, to waiting for I/O to complete, and to processing nice tasks. High percentages of nice might indicate that other processes are crowding out Cassandra processes, while high percentages of iowait might indicate I/O contention. On fully virtualized environments like Amazon EC2, a Cassandra cluster under load might show high steal values while other virtual processors use the available system resources.
- Windows and Mac OS X - Shows how much time the CPU spends on user processes and system processes.

OS: Load

The amount of work that a computer system performs. An idle computer has a load number of 0 and each process using or waiting for CPU time increments the load number by 1. Any value above one indicates that the machine was temporarily overloaded and some processes were required to wait. Shows minimum, average, and maximum OS load expressed as an integer.

OS: Disk usage (GB)

Tracks growth or reduction in the amount of available disk space used. If this metric indicates a growth trend leading to high or total disk space usage, consider strategies to relieve it, such as adding capacity to the cluster. DataStax recommends leaving 30-50% free disk space for optimal repair and compaction operations.

OS: Disk Usage (percentage)

The percentage of disk space that is being used by Cassandra at a given time. When Cassandra is reading and writing heavily from disk, or building SSTables as the final product of compaction processes, disk usage values may be temporarily higher than expected.

OS: Disk Throughput

The average disk throughput for read and write operations, measured in megabytes per second. Exceptionally high disk throughput values may indicate I/O contention. This is typically caused by numerous compaction processes competing with read operations. Reducing the frequency of memtable flushing can relieve I/O contention.

OS: Disk Rates

- Linux and Windows - Averaged disk speed for read and write operations.
- Mac OS X - Not supported.

OS: Disk Latency

- Linux and Windows - Measures the average time consumed by disk seeks in milliseconds. Disk latency is among the higher-level metrics that may be useful to monitor on an ongoing basis by keeping this graph posted on your OpsCenter performance console. Consistently high disk latency may be a signal to investigate causes, such as I/O contention from compactions or read/write loads that call for expanded capacity.
- Mac OS X - Not supported.

OS: Disk Request Size

- Linux and Windows - The average size in sectors of requests issued to the disk.
- Mac OS X - Not supported.

OS: Disk Queue Size

- Linux and Windows - The average number of requests queued due to disk latency issues.
- Mac OS X - Not supported.

OS: Disk Utilization

- Linux and Windows - The percentage of CPU time consumed by disk I/O.
- Mac OS X - Not supported.

Alert metrics

From the Alerts area of OpsCenter, you can configure alert thresholds for a number of Cassandra cluster-wide, column family, and operating system metrics. This proactive monitoring feature is available for DataStax Enterprise clusters.

From the Alerts area of OpsCenter, you can configure alert thresholds for a number of Cassandra cluster-wide, column family, and operating system metrics. This proactive monitoring feature is available for DataStax Enterprise clusters.

Commonly watched alert metrics

Metric	Definition
Node down	When a node is not responding to requests, it is marked as down.
Write requests	The number of write requests per second. Monitoring the number of writes over a given time period can give you and idea of system write workload and usage patterns.
Write request latency	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.
Read requests	The number of read requests per second. Monitoring the number of reads over a given time period can give you and idea of system read workload and usage patterns.
Read request latency	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.
CPU usage	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.
Load	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.

Advanced Cassandra alert metrics

Metric	Definition
Heap max	The maximum amount of shared memory allocated to the JVM heap for Cassandra processes.
Heap used	The amount of shared memory in use by the JVM heap for Cassandra processes.
JVM CMS collection count	The number of concurrent mark-sweep (CMS) garbage collections performed by the JVM per second.
JVM ParNew collection count	The number of parallel new-generation garbage collections performed by the JVM per second.
JVM CMS collection time	The time spent collecting CMS garbage in milliseconds per second (ms/sec).

Metric	Definition
JVM ParNew collection time	The time spent performing ParNew garbage collections in ms/sec.
Data size	The size of column family data (in gigabytes) that has been loaded/inserted into Cassandra, including any storage overhead and system metadata.
Compactions pending	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.
Total bytes compacted	The number of SSTable data compacted in bytes per second.
Total compactions	The number of compactions (minor or major) performed per second.
Flush sorter tasks pending	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).
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).
Gossip tasks pending	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).
Hinted hand-off pending	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.
Internal response pending	The number of pending tasks from various internal tasks such as nodes joining and leaving the cluster.
Manual repair tasks pending	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.
Memtable postflushers pending	The memtable post flush process performs the final step in the overall process of flushing memtables to disk as SSTables. The

Metric	Definition
	optimal number of pending flushes is 0 (or at most a very small number).
Migrations pending	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.
Miscellaneous tasks pending	The number of pending tasks from other miscellaneous operations that are not ran frequently.
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).
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.
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).
Request response pending	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.
Streams pending	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.
Write requests pending	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).

Advanced column family alert metrics

Metric	Definition
Local writes	The write load on a column family measured in operations per second. This metric includes all writes to a given column family, including write requests forwarded from other nodes.
Local write latency	The response time in milliseconds for successful write operations on a column family. The time period starts when nodes receive a write request, and ends when nodes respond.

Metric	Definition
Local reads	The read load on a column family measured in operations per second. This metric includes all reads to a given column family, including read requests forwarded from other nodes.
Local read latency	The response time in microseconds for successful read operations on a column family. The time period starts when a node receives a read request, and ends when the node responds.
Column family key cache hits	The number of read requests that resulted in the requested row key being found in the key cache.
Column family key cache requests	The total number of read requests on the row key cache.
Column family key cache hit rate	The key cache hit rate indicates the effectiveness of the key cache for a given column family by giving the percentage of cache requests that resulted in a cache hit.
Column family row cache hits	The number of read requests that resulted in the read being satisfied from the row cache.
Column family row cache requests	The total number of read requests on the row cache.
Column family row cache hit rate	The key cache hit rate indicates the effectiveness of the row cache for a given column family by giving the percentage of cache requests that resulted in a cache hit.
Column family bloom filter space used	The size of the bloom filter files on disk.
Column family bloom filter false positives	The number of false positives, which occur when the bloom filter said the row existed, but it actually did not exist in absolute numbers.
Column family bloom filter false positive ratio	The fraction of all bloom filter checks resulting in a false positive.
Live disk used	The current size of live SSTables for a column family. It is expected that SSTable size will grow over time with your write load, as compaction processes continue doubling the size of SSTables. Using this metric together with SSTable count, you can monitor the current state of compaction for a given column family.
Total disk used	The current size of the data directories for the column family including space not reclaimed by obsolete objects.
SSTable count	The current number of SSTables for a column family. When column family memtables are persisted to disk as SSTables, this metric increases to the configured maximum before the compaction cycle is repeated. Using this metric together with live disk used, you can monitor the current state of compaction for a given column family.
Pending reads and writes	The number of pending reads and writes on a column family. Pending operations are an indication that Cassandra is not keeping up with the workload. A value of zero indicates healthy throughput.

Advanced system alert metrics

Configure advanced system metrics for memory, CPU, and disk metrics on Linux, Windows, or Mac OS X.

Configure advanced system metrics for memory, CPU, and disk metrics on Linux, Windows, or Mac OS X. As with any database system, Cassandra performance greatly depends on underlying systems on which it is running. Before configuring advanced system metric alerts, you should first have an understanding of the baseline performance of your hardware and the averages of these system metrics when the system is handling a typical workload.

Linux memory metrics

Metric	Definition
Memory free	System memory that is not being used.
Memory used	System memory used by application processes.
Memory buffered	System memory used for caching file system metadata and tracking in-flight pages.
Memory shared	System memory that is accessible to CPUs.
Memory cached	System memory used by the OS disk cache.

Linux CPU metrics

Metric	Definition
Idle	Percentage of time the CPU is idle.
Iowait	Percentage of time the CPU is idle and there is a pending disk I/O request.
Nice	Percentage of time spent processing prioritized tasks. Niced tasks are also counted in system and user time.
Steal	Percentage of time a virtual CPU waits for a real CPU while the hypervisor services another virtual processor.
System	Percentage of time allocated to system processes.
User	Percentage of time allocated to user processes.

Linux Disk metrics

Metric	Definition
Disk usage	Percentage of disk space Cassandra uses at a given time.
Free disk space	Available disk space in GB.
Used disk space	Used disk space in GB.
Disk read throughput	Average disk throughput for read operations in megabytes per second. Exceptionally high disk throughput values may indicate I/O contention.
Disk write throughput	Average disk throughput for write operations in megabytes per second.
Disk read rate	Averaged disk speed for read operations.
Disk write rate	Averaged disk speed for write operations.

Metric	Definition
Disk latency	Average time consumed by disk seeks in milliseconds.
Disk request size	Average size in sectors of requests issued to the disk.
Disk queue size	Average number of requests queued due to disk latency.
Disk utilization	Percentage of CPU time consumed by disk I/O.

Windows memory metrics

Metric	Definition
Available memory	Physical memory that is not being used.
Pool nonpaged	Physical memory that stores the kernel and other system data structures.
Pool paged resident	Physical memory allocated to unused objects that can be written to disk to free memory for reuse.
System cache resident	Physical pages of operating system code in the file system cache.

Windows CPU metrics

Metric	Definition
Idle	Percentage of time the CPU is idle.
Privileged	Percentage of time the CPU spends executing kernel commands.
User	Percentage of time allocated to user processes.

Windows Disk metrics

Metric	Definition
Disk usage	Percentage of disk space Cassandra uses at a given time.
Free disk space	Available disk space in GB.
Used disk space	Used disk space in GB.
Disk read throughput	Average disk throughput for read operations in megabytes per second. Exceptionally high disk throughput values may indicate I/O contention.
Disk write throughput	Average disk throughput for write operations in megabytes per second.
Disk read rate	Averaged disk speed for read operations.
Disk write rate	Averaged disk speed for write operations.
Disk latency	Average time consumed by disk seeks in milliseconds.

Metric	Definition
Disk request size	Average size in sectors of requests issued to the disk.
Disk queue size	Average number of requests queued due to disk latency.
Disk utilization	Percentage of CPU time consumed by disk I/O.

Mac OS X memory metrics

Metric	Definition
Free memory	System memory that is not being used.
Used memory	System memory that is being used by application processes.

Mac OS X CPU metrics

Metric	Definition
Idle	Percentage of time the CPU is idle.
System	Percentage of time allocated to system processes.
User	Percentage of time allocated to user processes.

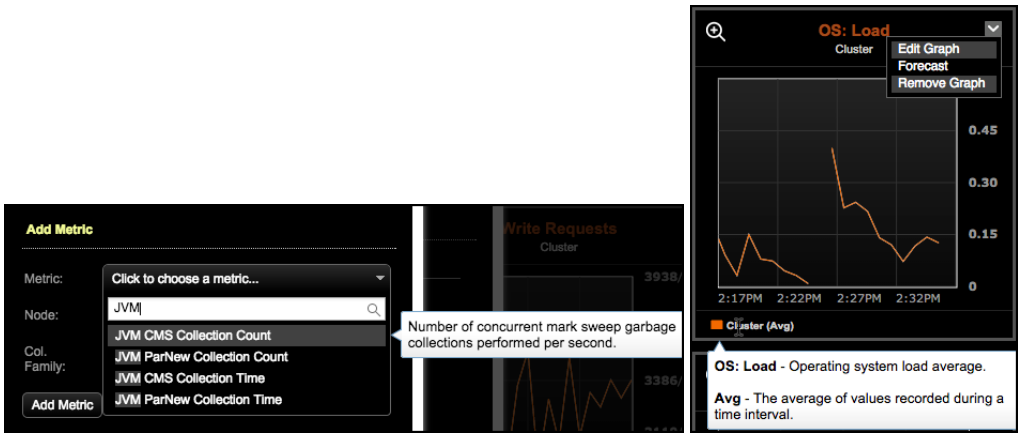
Mac OS X disk metrics

Metric	Definition
Disk usage	Percentage of disk space Cassandra uses at a given time.
Free space	Available disk space in GB.
Used disk space	Used disk space in GB.
Disk throughput	Average disk throughput for read/write operations in megabytes per second. Exceptionally high disk throughput values may indicate I/O contention.

OpsCenter Metrics Tooltips Reference

Comprehensive reference of performance metrics available in OpsCenter.

Metrics are available to add to any graph. View descriptions of any metric by hovering over a metric in the Add Metric dialog, or by hovering over a graph legend.



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

Write Requests

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

The *average* 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.

Write Latency (Stacked)

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.

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

The *average* 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.

Read Latency (Stacked)

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.

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.

Data Size

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

Total Bytes Compacted

Number of bytes compacted per second.

Total Compactions

Number of compaction tasks completed per second.

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.

Flushes Pending

Number of memtables queued for the flush process. A flush sorts and writes the memtables to disk, which could block writes.

Gossip Tasks Pending

Number of gossip messages and acknowledgments queued and waiting to be sent or received.

Hinted Handoff Pending

Number of hints in the queue waiting to be delivered after a failed node comes up.

Internal Responses Pending

Number of pending tasks from internal tasks, such as nodes joining and leaving the cluster.

Manual Repair Tasks Pending

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

Memtable Post Flushers Pending

Number of 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.

Migrations Pending

Number of pending tasks from system methods that modified the schema.

Misc. Tasks Pending

Number of pending tasks from infrequently run operations, such as taking a snapshot or processing the notification of a completed replication.

Read Requests Pending

Number of pending read requests. Read requests read data off of disk and deserialize cached data.

Read Repair Tasks Pending

Number of read repair operations in the queue waiting to run.

Repl. on Write Tasks Pending

Number of pending counter increment tasks that will read then write on the replicas after a coordinator's local write. Depending on consistency level used on writes, tasks may back up outside of the normal write path.

Request Responses Pending

Number of pending callbacks to execute after a task on a remote node completes.

Write Requests Pending

Number of write requests received by the cluster and waiting to be handled.

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.

Dropped Tasks

Aggregate of different messages that might be thrown away.

Dropped Counter Mutations

Mutation was seen after the timeout (write_request_timeout_in_ms) so was thrown away. This client might have timed out before it met the required consistency level, but might have succeeded as well. Hinted handoffs and read repairs should resolve inconsistencies but a repair can ensure it.

Dropped Mutations

Mutation was seen after the timeout (write_request_timeout_in_ms) so was thrown away. This client might have timed out before it met the required consistency level, but might have succeeded as well. Hinted handoffs and read repairs should resolve inconsistencies but a repair can ensure it.

Dropped Reads

A local read request was received after the timeout (read_request_timeout_in_ms) so it was thrown away because it would have already either been completed and sent to client or sent back as a timeout error.

Dropped Ranged Slice Reads

A local ranged read request was received after the timeout (range_request_timeout_in_ms) so it was thrown away because it would have already either been completed and sent to client or sent back as a timeout error.

Dropped 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.

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.

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.

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.

CF: Local Writes

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

CF: Local Write Latency

Average response time to write data to a table's memtable and append to the commitlog. The elapsed time from when the replica receives the request from a coordinator and returns a response.

CF: Write Latency (Stacked)

The min, median, max, 90th, and 99th percentile of the response times to write data to a table's memtable and append to the commitlog. The elapsed time from when the replica receives the request from a coordinator and returns a response.

CF: Local Reads

Local read requests per second. Local reads retrieve data from a table's memtable and any necessary SSTables on disk.

CF: Local Read Latency

Average 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.

CF: Read Latency (Stacked)

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.

CF: Live Disk Used

Disk space used by live SSTables. There might be obsolete SSTables not included.

CF: Total Disk Used

Disk space used by a table by SSTables, including obsolete ones waiting to be garbage collected.

CF: SSTable Count

Total number of SSTables for a table.

CF: SSTables per Read (Stacked)

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

CF: Pending Reads/Writes

Estimate of the number of mutation threads blocked on a memtable flush or truncate.

CF: Bloom Filter Space Used

The total size of all the SSTables' bloom filters for this table.

CF: Bloom Filter False Positives

Number of bloom filter false positives per second.

CF: Bloom Filter False Positive Ratio

Percentage of bloom filter lookups that resulted in a false positive.

CF: Bloom Filter Off Heap

Total off heap memory used by bloom filters from all live SSTables in a table.

CF: Index Summary Off Heap

Total off heap memory used by the index summary of all live SSTables in a table.

CF: Compression Metadata Off Heap

Total off heap memory used by the compression metadata of all live SSTables in a table.

CF: Memtable Off Heap

Off heap memory used by a table's current memtable.

CF: Total Memtable Size

An estimate of the space used in memory (including JVM overhead) for all memtables. This includes ones that are currently being flushed and related secondary indexes.

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.

In-Memory Percent Used

The percentage of memory allocated for in-memory tables currently in use.

OS: Memory (stacked)

Stacked graph of used, cached, and free memory.

OS: Memory (stacked)

Stacked graph of used and free memory for OSX.

OS: Memory (stacked)

Stacked graph of committed, cached, paged, non-paged, and free memory for Windows.

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 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 for OSX.

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

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

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

OS: CPU Idle
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.

DSE Management Services

DataStax Enterprise (DSE) comes bundled with enterprise management services that you can configure and run using OpsCenter.

Backup Service

Describes the Backup Service, which allows you to automatically or manually backup and restore data in your clusters.

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.

Overview of the Backup Service

The Backup Service allows you to backup and restore your cluster data.

Using OpsCenter, you can schedule and manage backups, and restore from those backups, across all registered DSE clusters. The Backup Service:

- Can perform all functions using the REST API or visually through the OpsCenter UI
- Delivers smart backups that always ensure full data protection, including backups of commit logs
- Can backup data to a local server or Amazon S3
- Compresses backup files to save storage
- Allows for the specification of 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 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.

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 the location.

Note: OpsCenter Data Backups does 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.

The backup files are stored in S3 in the following hierarchy:

```
mybucket/
  snapshots/
    node-id1/
      sstables/
        MyKeyspace-MyTable-ic-5-Data.db
        ...
        MyKeyspace-MyTable-ic-5-TOC.txt
        MyKeyspace-MyTable-ic-6-Data.db
        ...
      1234-ABCD-2014-10-01-01-00/
        backup.json
        MyKeyspace/schema.json
      1234-ABCD-2014-09-30-01-00/
        backup.json
        MyKeyspace/schema.json
    node-id2/
      sstables/
        MyKeyspace-MyTable-ic-1-Data.db
        ...
        MyKeyspace-MyTable-ic-2-Data.db
        ...
      1234-ABCD-2014-10-01-01-00/
        backup.json
        MyKeyspace/schema.json
      1234-ABCD-2014-09-30-01-00/
        backup.json
        MyKeyspace/schema.json
 commitlogs/
   node1/
     1435432324_Commitlog-3-1432320421.log
     1435433232_Commitlog-3-1432320422.log
     ...
```

The `backup.json` file contains metadata about which of the backed up SSTables are included in that backup.

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 will handle 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 this 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 will scan the snapshot data for outdated files that do not belong to other snapshots, and remove them.

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 separately from snapshot backups.

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`

- Windows installations: `Program Files (x86)\DataStax Community\opscenter\conf\clusters\`*`cluster_name.conf`*

Why use the Backup Service?

The Backup Service provides backup and recovery for an enterprise's data even though DataStax Enterprise has built-in replication.

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 Cassandra's powerful and flexible replication capabilities, proper backup and restore procedures are still very important to implement for production clusters.

While Cassandra's replication does provide for copies of data to exist in multiple locations, data centers, 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 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.

We strongly recommended that organizations that use 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.

Backing up a cluster

OpsCenter provides a way to schedule and run backup operations on a cluster.

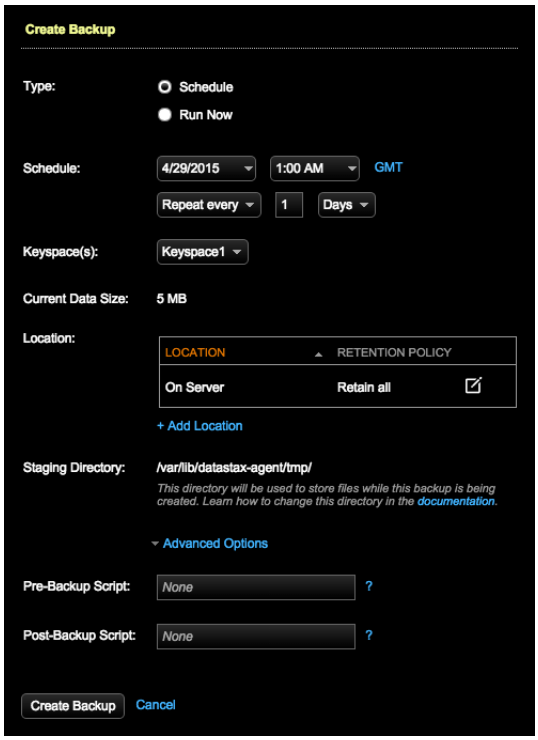
OpsCenter allows organizations to run one-time backup jobs as well as schedule backup jobs to run at a later date and on a recurring basis. Commitlog backups allow you to restore the data to a particular date and time.

Creating a recurring backup

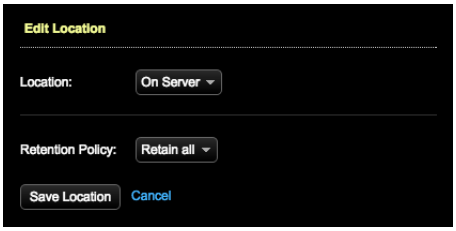
OpsCenter can create a scheduled backup to run periodic backup operations.

Procedure

1. Click the name of the cluster you want to manage from the left navigation pane and click **Services**.
2. Click **View Details** for the Backup Service.
3. Click **Create Backup**.
   The Create Backup dialog appears.



4. Select the backup parameters:
   a) Under **Type**, click **Schedule**.
   b) **Schedule**—Select a date and 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) **Select a Keyspace to backup**—Select the keyspace that you want to back up, or select All Keyspaces.
   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`. In the **Add Location** pane, click the edit icon to edit the Location and Retention Policy.



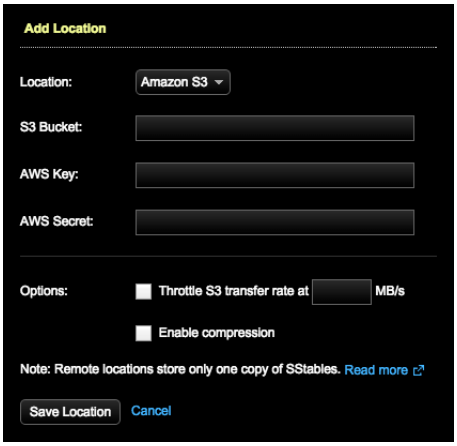
   e) Select the location type of On Server or Amazon S3 in **Location**. Indicate how long the snapshot data should be saved by selecting a **Retention Policy**. The Retention Policy can be set to Retain All, which saves the snapshot data indefinitely, or to a set period of time. After the snapshot data is older than the time set in Retention Policy, the snapshot data is deleted.

      To save the snapshot data to Amazon S3, select **Amazon S3** for the Location, set the bucket name, and enter your AWS key and secret. The bucket name must conform to the S3 guidelines.

      Note: You must use Java 7 or greater to store at an S3 location.

      (Applicable to Amazon S3 only) To avoid saturating your network, set a maximum upload rate. Select **Throttle S3 transfer rate** and set the maximum MB per second to upload the data. To compress the data, select **Enable compression**. This reduces the amount of data going through your network and reduces the disk and data usage in S3, but increases the CPU load for the server.

f) If you want the snapshot copied to an additional location in Amazon S3, click **Add Location**. The Add Location dialog appears.



g) Optional: 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 is started.

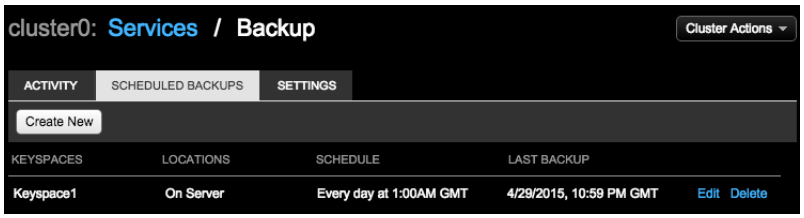
Post-Backup Script— Enter the name of a script to run after the backup is completed.

For details, see Using custom scripts before and after backups.

5. Click **Create Backup**.

Results

The scheduled backup appears in the **Scheduled Backups** tab.



Creating a one-time backup

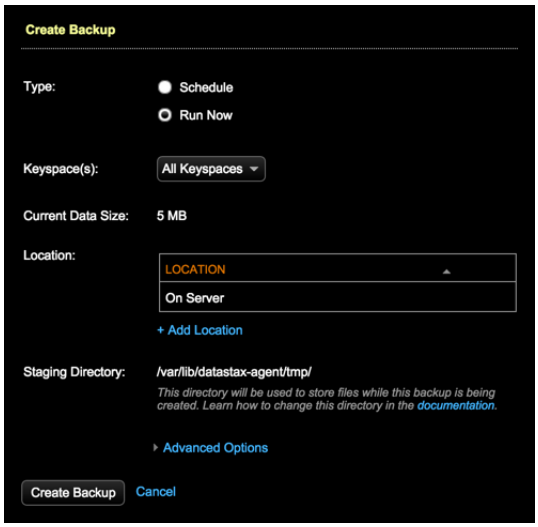
OpsCenter can create a one-time backup operation.

About this task

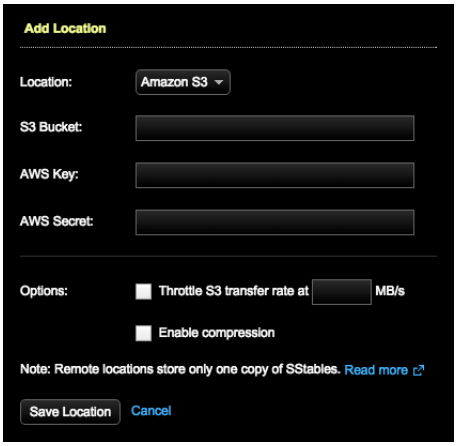
Run an ad hoc backup. A one-time backup only runs once and starts immediately after creating the job in the OpsCenter UI.

Procedure

1. Click the name of the cluster you want to manage from the left navigation pane and click **Services**.
2. Click **View Details** for the Backup Service.
3. Click **Create Backup**.
   The Create Backup dialog appears.



4. Under **Type**, click **Run Now**.
5. **Select a Keyspace to backup**—Select the keyspace that you want to back up, or select All Keyspaces.
   a) **Location**—Snapshots are saved to the node's snapshot directory for the table being saved. For example, `/var/lib/cassandra/data/OpsCenter/settings/snapshots`. If you want the snapshot copied to an additional location in Amazon S3, click **Add Location**.
   The Add Location dialog appears.



   b) **Note:** You must use Java 7 or greater to store at an S3 location.

   To save the snapshot data to Amazon S3, select **Amazon S3** for the Location, set the bucket name, and enter your AWS key and secret. The bucket name must conform to the S3 guidelines.
   c) To avoid saturating your network, set a maximum upload rate. Select **Throttle S3 transfer rate** and set the maximum MB per second to upload the data. To compress the data, select **Enable compression**. This reduces the amount of data going through your network and reduces the disk and data usage in S3, but increases the CPU load for the server.
   d) Optional: 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 is started.

   Post-Backup Script— Enter the name of a script to run after the backup is completed.

   For details, see Using custom scripts before and after backups.
6. Click **Create Backup**.

Results

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.

Enabling commitlog backups

Commitlog backups allow point-in-time restores.

Before you begin

- You must use Java 7 when backing up commitlogs.

About this task

Commitlog backups allow you to restore your data to a particular point in time.

Commitlog backups are only available for the following versions of DataStax Enterprise:

- For DataStax Enterprise 3.2: 3.2.8 or greater
- For DataStax Enterprise 4.0: 4.0.5 or greater
- For DataStax Enterprise 4.5: 4.5.3 or greater
- 4.6 or greater

Note:

Enabling commitlog archiving modifies Cassandra's `commitlog_archiving.properties` configuration file.

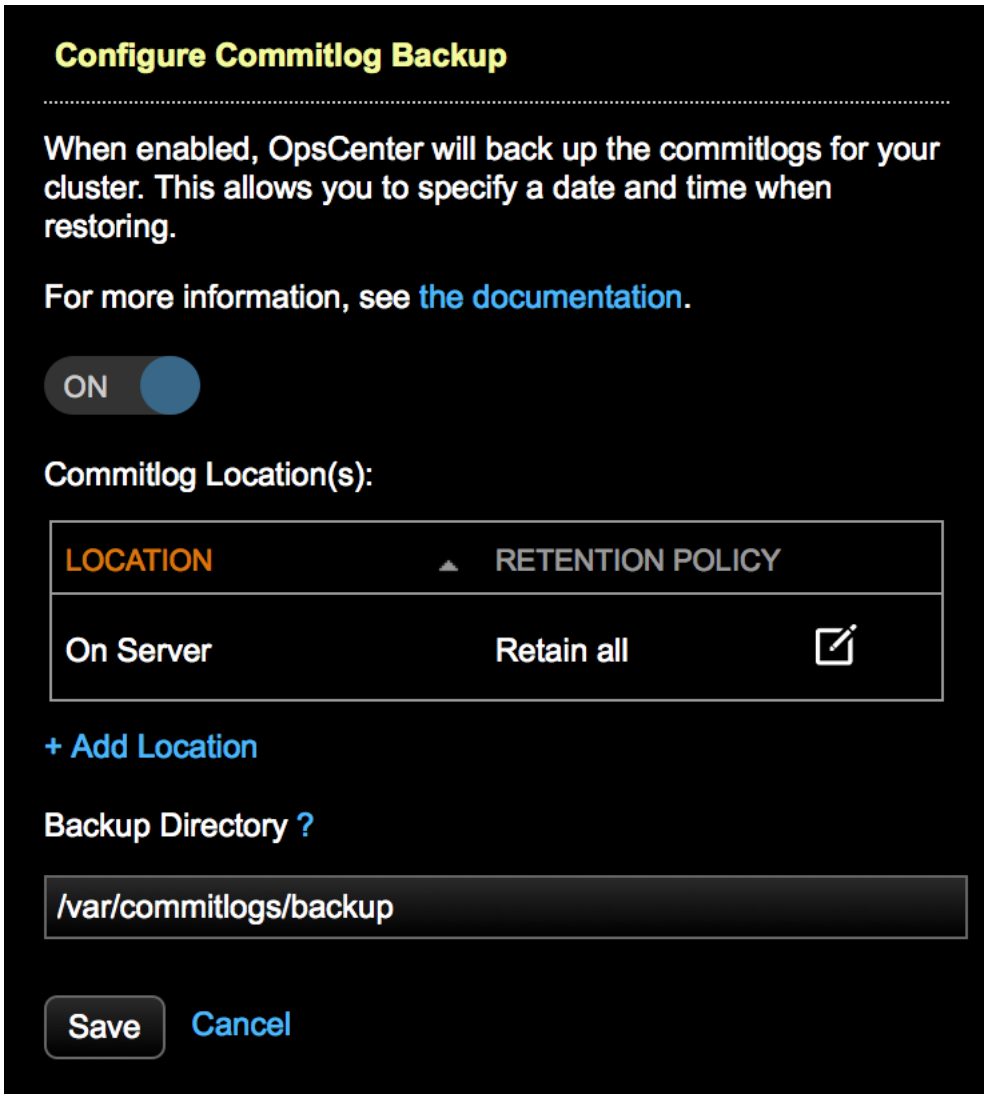
OpsCenter automatically enables commitlog archiving on the new nodes when adding new nodes to a cluster using OpsCenter if commitlog archiving is enabled on the cluster. If you manually add nodes to a cluster and commitlog archiving is enabled, you must manually copy `commitlog_archiving.properties` to the new nodes prior to starting DataStax Enterprise.

Procedure

1. Click the name of the cluster you want to manage from the left pane.
2. Click **Services**.
3. Click **View Details** for the Backup Service.
4. Click the **Settings** tab.
5. Under **Commitlog Backup** click **Configure**.
6. Set the slider to On.
7. Enter the path where you want to store the commitlogs on each node under **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. Commitlog files record every mutation of the data in a cluster, and can grow quite large over time. The disk performance of this location is also extremely important, as the disk write speed will likely be a bottleneck for write-heavy use cases, and the read performance a bottleneck for restore operations.



8. If you want to backup the commitlogs to a cloud storage provider like Amazon S3, click **Add Location**.
   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**. This will reduce the amount of data going through your network, but will increase 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**.
9. Click **Save**.
10. OpsCenter will perform a rolling restart of your cluster after enabling commitlog backups. Set the sleep time between restarting each node, and optionally select **Drain before stopping** if you want to perform a `nodetool drain` operation, and click **Restart Cluster**.

Using custom scripts before and after backups

Scripts can be run before or after backup operations for further customization.

You can configure custom scripts that will run before or after a backup.

Scheduled backups can be configured to run custom scripts before and after the backup is performed. These custom scripts need to be located in `/usr/share/datastax-agent/bin/backup-scripts` for package installations, or in *`install location`*`/bin/backup-scripts` for tarball installations. This directory also contains example scripts. The scripts need to be executable, and are 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, it 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 will not have any arguments passed to them.

Configuring the Backup Service to upload very large files to Amazon S3

Change the settings of the agent to allow it to upload very large files to Amazon S3.

The default settings for the DataStax agent prevent it from uploading SSTables to S3 that are over a certain size. 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. This limitation can be increased by modifying some properties on the agent 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"
```

In order 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 we can upload. Increasing the chunk size also requires using more memory on the agent, so the agent heap size also needs to be increased.

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 and allow for the larger SSTable sizes.

Restoring a cluster

Restore the data in a cluster from the stored backups.

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.

When restoring from backups stored on Amazon S3, OpsCenter will choose 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 original node. If the cluster topology is unchanged from when the backup was taken, OpsCenter will tell 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 will try to match the SSTables to the node that originally stored the SSTable, and will distribute the remaining SSTables to the remaining nodes to distribute the load evenly.

If you are doing a point-in-time restore, your cluster topology must not have changed since the backup. Attempting to perform a point-in-time restore on a cluster whose topology has changed will result in a failure. We recommend that you perform a snapshot backup before any topology changes. You can then restore the cluster based on that backup.

Restoring from a backup

OpsCenter allows restoring data from a previously completed backup operation.

About this task

You can restore from any local or Amazon S3 backups that have been run by OpsCenter, but not from snapshots run from `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.

Note: Automatic schema recreation is not currently supported in Cassandra 2.1+ or DataStax Enterprise 4.7+ when using User Defined Types (UDTs). When restoring a table using UDTs, please ensure the table exists before starting the restore operation.

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.

Before you begin

To restore an encrypted backup, the agent must be granted password-less `sudo` access on the DSE 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.

Procedure

1. Click the name of the cluster you want to manage from the left pane.
2. Click **Services**.
3. Click **View Details** for the Backup Service.
4. Click **Restore Backup**.
5. Find the backup you wish 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, you can restore from a particular point-in-time by selecting the **Point In Time** tab. For more details, see Performing a point-in-time restore.
   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, 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.

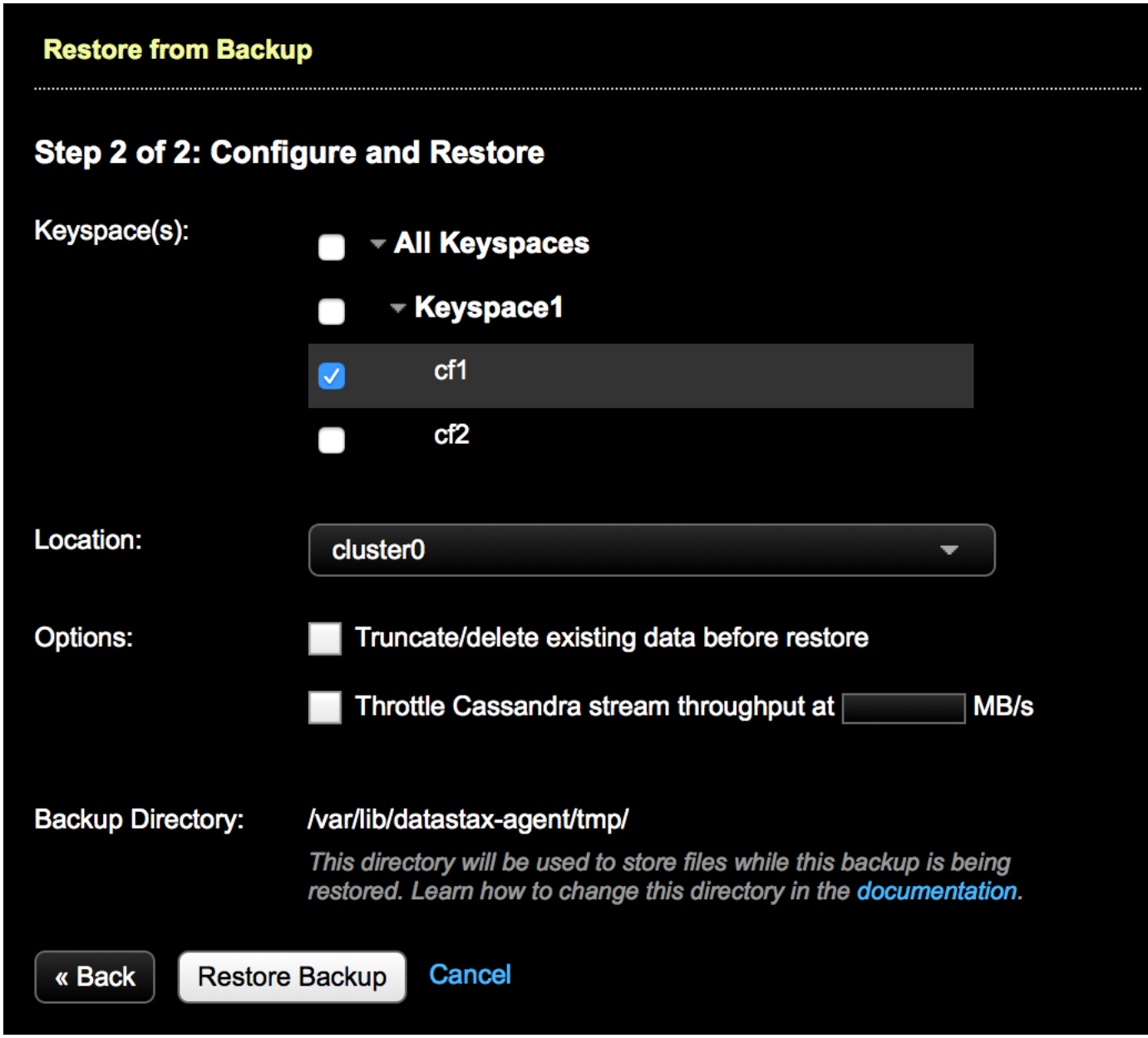
   To select only specific tables, expand the keyspace name and select the tables.
7. Under **Location**, select the target cluster for the restored data.

   - The Location list is only available when both clusters are managed by the same instance of OpsCenter.
   - Restoring to a different cluster is only supported when the backup resides in S3; restoring from a local backup is not currently supported.
   - 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 will completely remove 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.

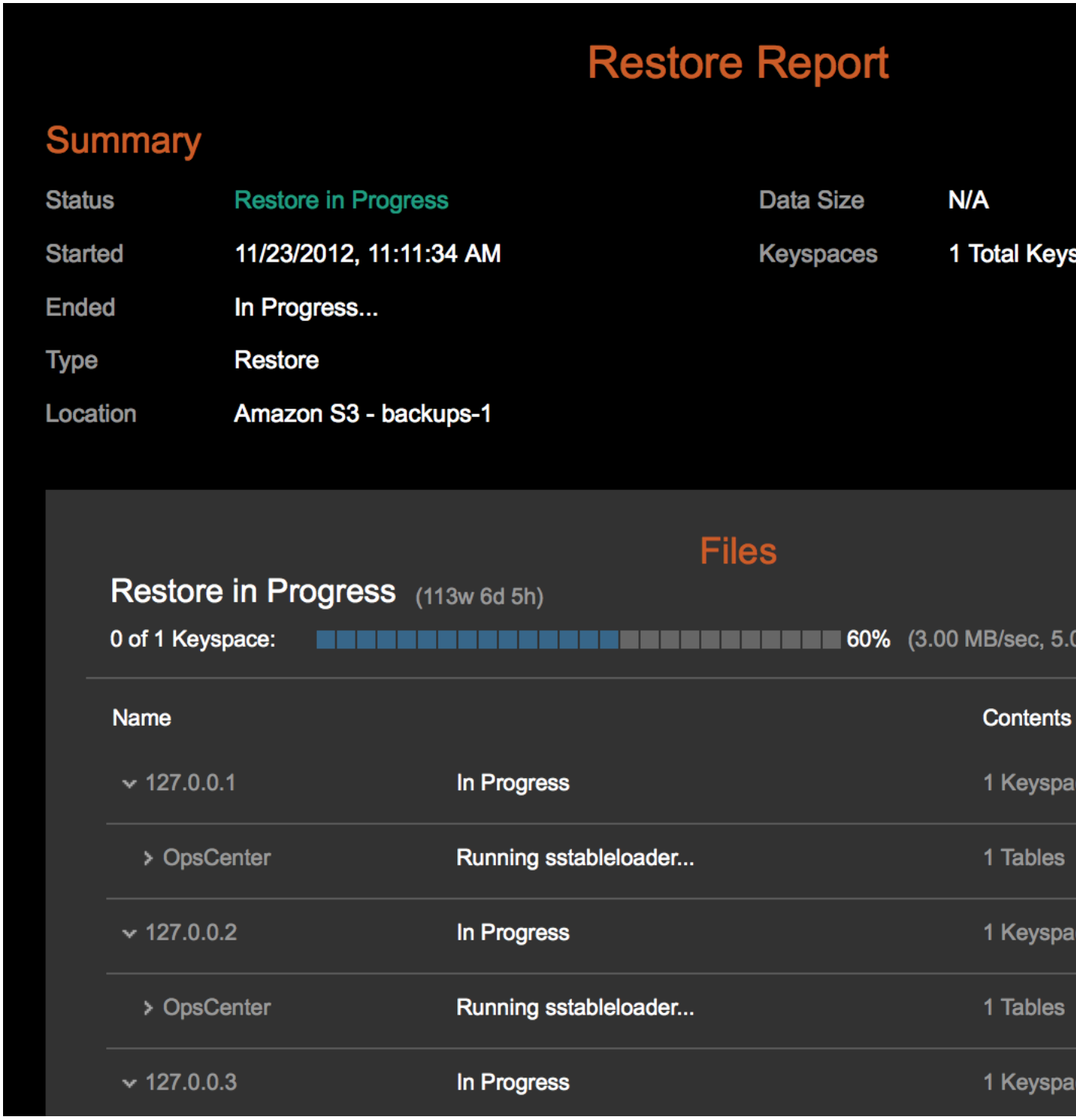


10. Click **Restore Backup**.
11. Click **Start Restore**.

Results

After the restore starts, a dialog displays detailed information about the status of the restore. This dialog can be closed at any time without affecting the restore process, and can be reopened by clicking on the **In Progress** restore in the **Activity** section in the OpsCenter UI.

Note: If you are restoring (essentially cloning) from an S3 backup, and you close the Restore Report dialog, you must reopen the status report from the destination cluster.



Performing a point-in-time restore

A point-in-time restore uses commitlog archives to restore data from a particular date and time using OpsCenter.

About this task

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:

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/dse/dse-env.sh`	X	X		
`install_location``/dse/dse-env.sh`			X	X

Before you begin

- For point-in-time restores to work, you must have enabled commitlog backups and performed at least one snapshot backup before the time to which you are restoring.
- **Known limitations**:
  - Point-in-time restore cannot restore commitlogs for keyspaces or tables that would have to be recreated in Cassandra 2.1+ and DSE 4.7+.
  - Point-in-time restore fails if any tables were recreated during the time period of the actual point-in-time restore.

Procedure

1. Click the name of the cluster you want to manage from the left pane.
2. Click **Services**.
3. Click **View Details** for the Backup Service.
4. Click **Restore Backup**.
5. Click the **Point In Time** tab.
6. Set the date and time to which you want to restore your data.
7. Select the location of the commitlogs under **Commitlogs Location** and the location of the snapshot under **Backup Location**, then click **Next**.
8. 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.

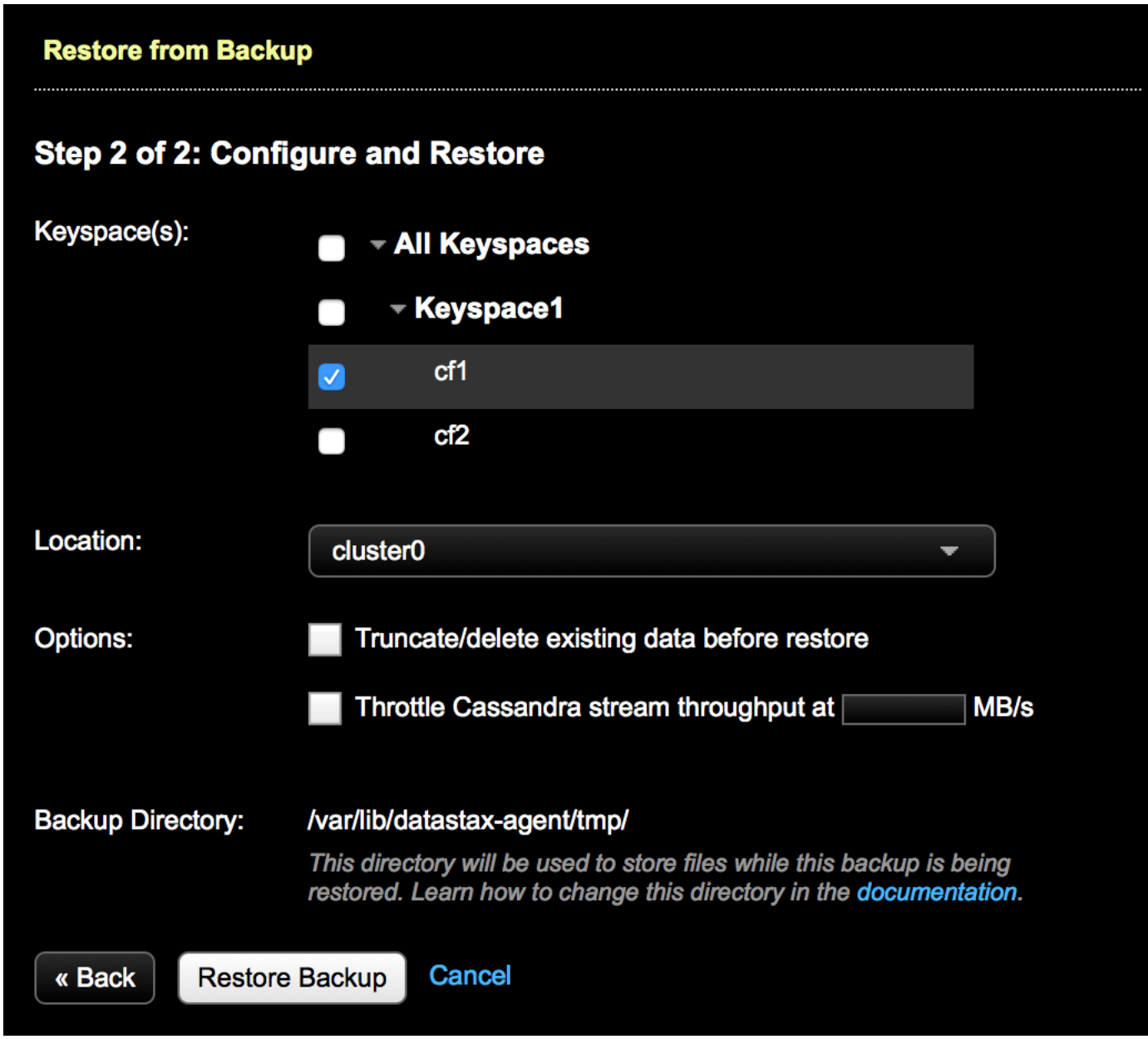
   To select only specific tables, expand the keyspace name and select the tables.
9. Under **Location**, select the target cluster for the restored data.
   - The Location list is only available when both clusters are managed by the same instance of OpsCenter.
   - Restoring to a different cluster is only supported when the backup resides in S3; restoring from a local backup is not currently supported.

- 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.

10. To remove the existing keyspace data before the data is restored, select **Truncate/delete existing data before restore**. This will completely remove any updated data in the cluster for the keyspaces you are restoring.
11. 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.



12. Click **Restore Backup**.
13. Click **Start Restore**.

Results

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:

```
export JVM_OPTS="$JVM_OPTS -Dcassandra.replayList=Keyspace1.Standard1"
```

Configuring restore to continue after a download failure

By default, if a file fails to download from S3 during a restore, the restore operation fails. Override the default configuration to allow the restore operation to continue after a file download fails.

About this task

By default, if a file fails to download from S3 during a restore, the restore operation fails. You can override the default configuration and allow the restore to continue after a file download fails.

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`

Procedure

1. Open the agent configuration file address.yaml and set the `restore_on_transfer_failure` option to `true`:
   `restore_on_transfer_failure: true`
2. Restart OpsCenter.

Cloning a cluster

Cluster data can be copied from one cluster to another using OpsCenter.

About this task

Cloning a cluster using the Backup Service is a simple way to copy the data from one cluster to another cluster while keeping the clusters separate. Unlike adding another data center to the cluster, where updates to the data will propagate across the data centers, a cloned cluster is a completely different cluster that is isolated from the original cluster. This is useful for development, where you need to update applications using real data but do not want to touch the production cluster. You can also use cloning for testing or performance tuning.

Before you begin

You must have backed up your cluster to an Amazon S3 location in order to clone the data.

Procedure

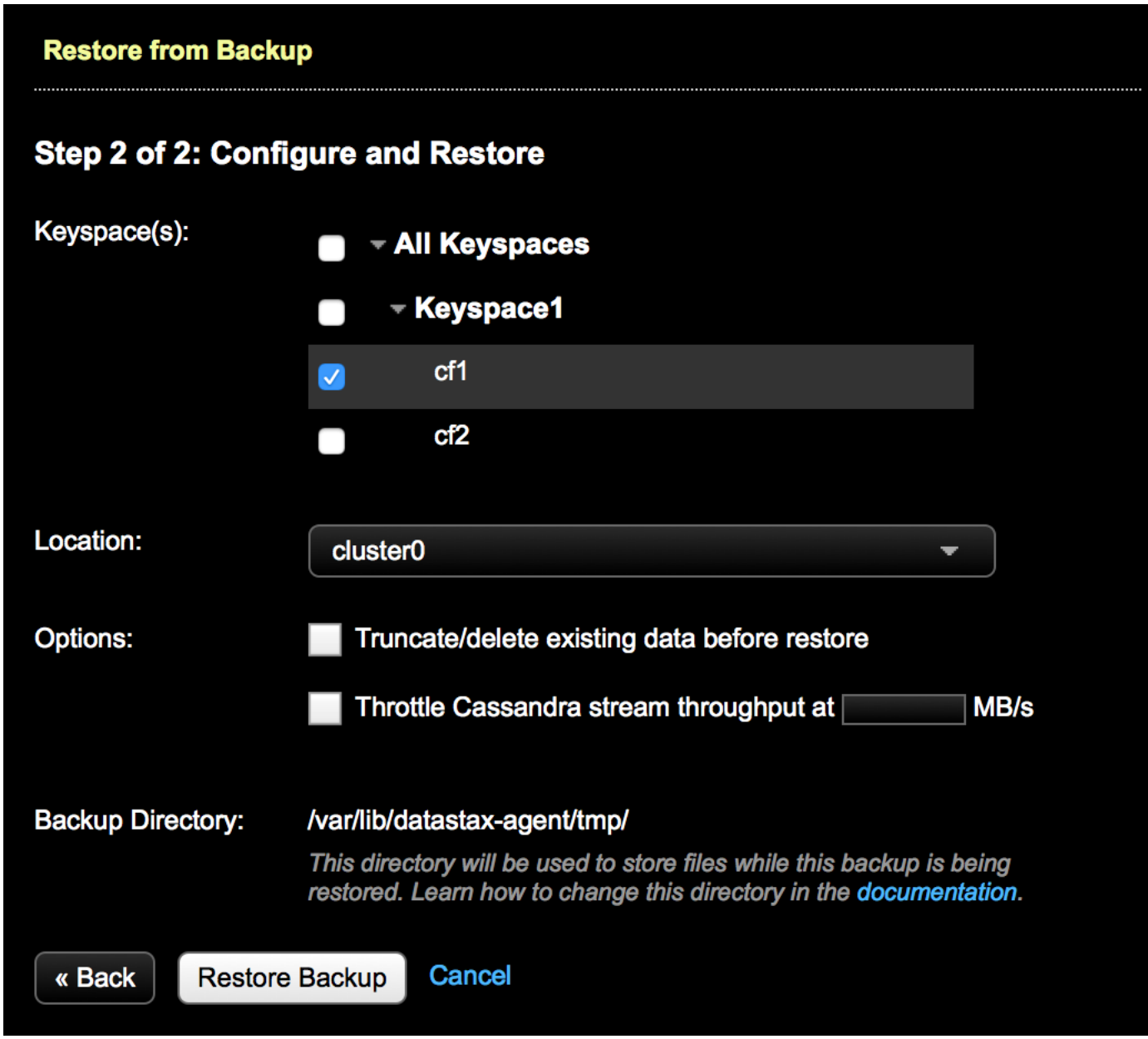
1. Click the name of the cluster you want to manage from the left pane.
2. Click **Services**.
3. Click **View Details** for the Backup Service.

4. Click **Restore Backup**.
5. Select the backup that contains the data you want to clone and click **Next**.

   If the OpsCenter instance does not manage the cluster you want to clone, you can clone it if the cluster data was backed up to an S3 location from another OpsCenter instance. Then select the **Other Location** tab in **Restore from Backup** to enter the S3 information to retrieve the backed up data.

   a) Click **Other Location**.
   b) Enter the S3 bucket name under **S3 Bucket**.
   c) Enter your AWS credentials under **AWS Key** and **AWS Secret**.
   d) Click **Next**.
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.
7. Under **Location** select the target cluster for the clone operation.
8. To remove the existing keyspace data before the data is restored, select **Truncate/delete existing data before restore**. This will completely remove 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. Click **Restore Backup**.
11. Click **Start Restore**.

Results

The details and progress of the restore operation will be displayed in a 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 destination cluster's **Backup Activity** section.

Configuring the Backup Service

Some Backup Service options can be configured in the OpsCenter UI.

Procedure

1. Click the name of the cluster you want to manage from the left navigation pane and click **Services**.
2. Click **View Details** for the Backup Service.

3. Click the **Settings** tab.
4. To enable commitlog backups, see the instructions on enabling commitlog backups.
5. To set a disk space threshold below which backup operations will not start, click **Configure** under **Disk Space Threshold**.
   a) Check the box and set the percentage of free space that must exist in order for a backup operation to start, then click **Save**.
6. Click **Configure** under **Encryption Key Storage** to set enable or disable whether OpsCenter will store the encryption keys for each node along with the SSTables. This is enabled by default, and highly recommended.

   If Encryption Key Storage is enabled and your cluster has encrypted keyspaces, the encryption key for each node will be stored in the backup location along with the data. If you disable this option you must ensure that each node's encryption key is available before attempting to restore encrypted tables.

   a) Set the slider to **On** to enable, or **Off** to disable, storing encryption keys alongside the backup data, then click **Save**.

Viewing backup and restore history

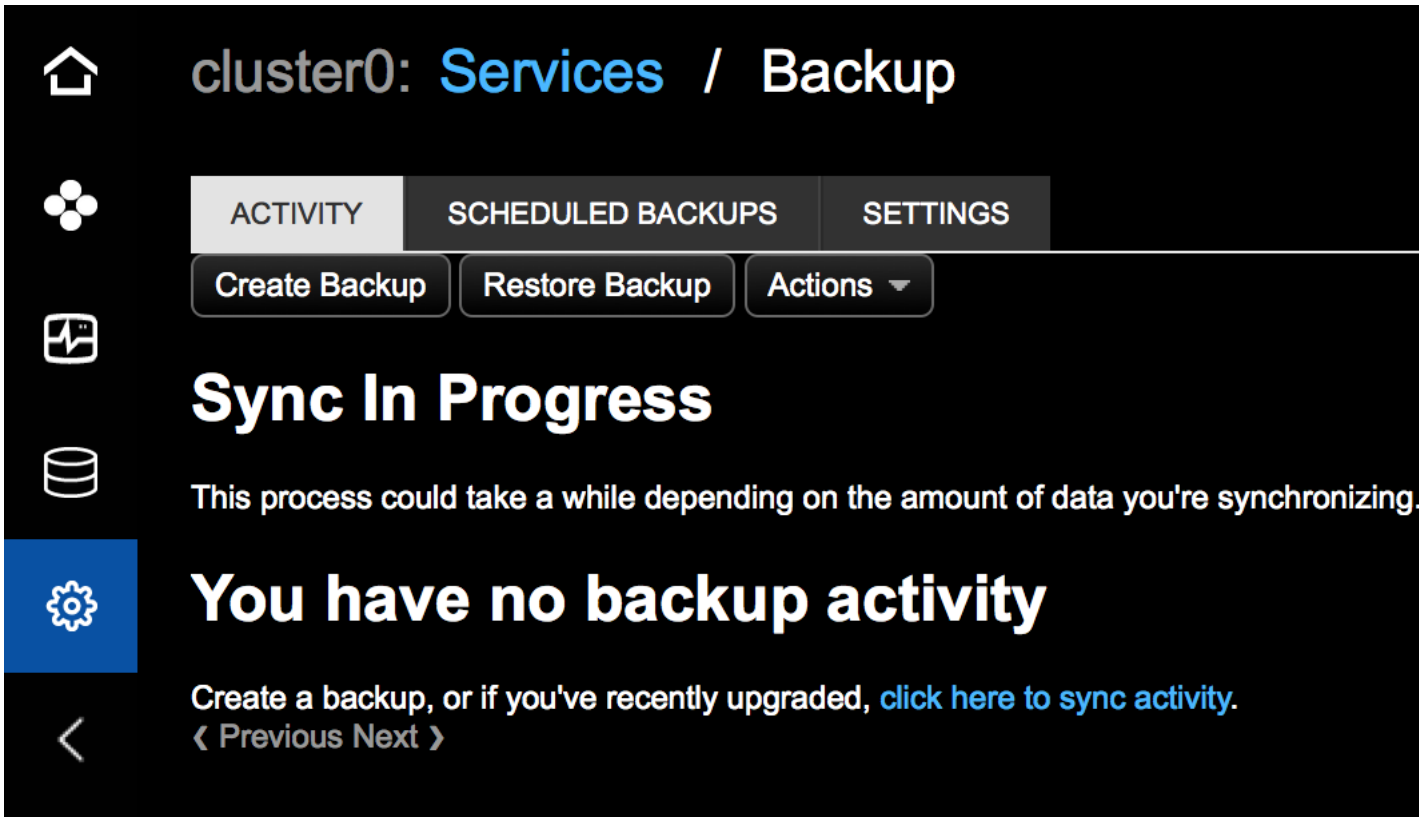
View in-process and completed backup or restore operations in OpsCenter, and synchronize the OpsCenter UI with the history log.

About this task

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 **Activities** tab.

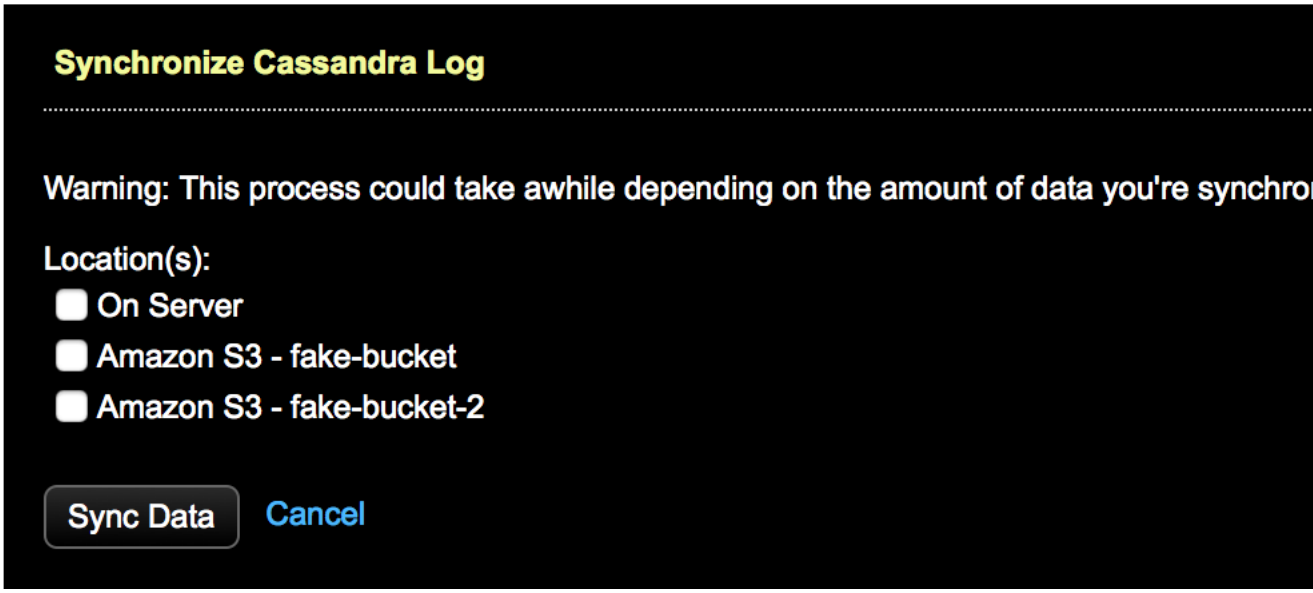
The details of all completed backup and restore operations are stored in the OpsCenter keyspace in Cassandra in the `backup_reports` table. The data is stored whether or not the operation was successful.

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.



Procedure

1. Click the name of the cluster you want to manage from the left pane.
2. Click **Services**.
3. Click **View Details** for the Backup Service.
4. Click **Activities** then **Synchronize Data** to manually synchronize the `backup_reports` table.
5. Select the locations whose history you want to synchronize and click **Sync Data**.



Troubleshooting Backup Service errors

Common solutions to errors encountered when using the Backup Service.

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.

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 to read and modify the files owned by the DataStax Enterprise user.

Error starting restore for a table using UDT

(Applicable to Cassandra version 2.1+ and DSE version 4.7+ only) Automatically recreating a schema is not currently supported when using User Defined Types (UDTs). When restoring a table using UDTs, please ensure the table exists before starting the restore operation. Future versions of OpsCenter will support automatic schema creation for UDTs.

Repair Service

The Repair Service is configured to run continuously and perform repair operations across a DSE cluster in a minimally impactful way.

The Repair Service is configured to run continuously and perform repair operations across a DSE 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: DSE 3.0 or greater is required.

How the Repair Service works

The Repair Service runs continuously as a background process. The Repair Service incrementally and cyclically repairs a DSE 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 works by repairing small chunks of a cluster in the background. The service takes a single parameter, time_to_completion, which is the maximum amount of time it takes to repair the entire cluster once. Typically, you set this to a 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.

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 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.

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.

Known limitations

If a cluster is datacenter aware and has keyspaces using `SimpleStrategy`, the repair service will fail to start. Follow the prompts to change the keyspaces to NetworkToplogyStrategy.

Changes in cluster topology

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. This check occurs every five minutes.

Topology changes:

- Nodes moving
- Nodes joining a cluster
- Nodes leaving a cluster
- Nodes being decommissioned

Changes in schemas

- Keyspaces added while the repair service is running are repaired when the next subrange repair is started.
- Column families (tables) added to existing keyspaces are repaired immediately during the current cycle of the Repair Service.
- Keyspaces or column families (tables) can be removed while the Repair Service is running without causing any issues.

Incremental repairs

Overview of incremental repairs, including considerations and limitations when running and configuring incremental repairs within OpsCenter.

For version 5.1.2, 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 Repairs Service currently repairs an entire datacenter at a time. Future versions of OpsCenter will remedy this current limitation in version 5.1.2.

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. For more information on migrating to incremental repairs, see the Cassandra documentation. Read more about efficient incremental repairs in the DataStax Developer blog post.

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

Whether incremental repairs should do subrange repair or full repair of a node's entire range.

[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

Start and stop the Repair Service.

Starting the Repair Service

Configure the days for the repair cycle to complete and start the Repair Service for a cluster.

About this task

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.

Before you begin

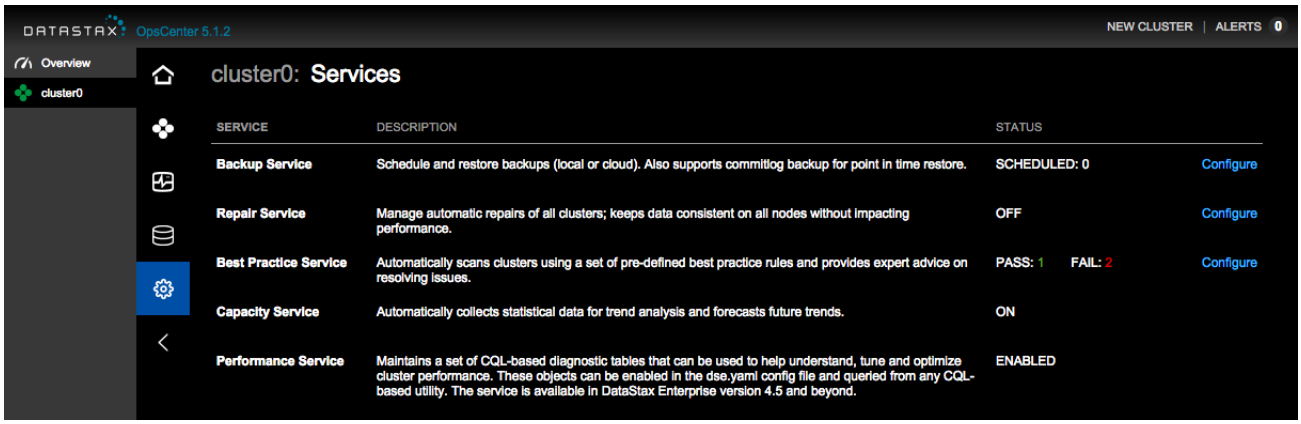
In a multi-datacenter environment, the NetworkTopologyStrategy is required as the replication strategy for all keyspaces to run the Repair Service. The Repair Service will not run with SimpleStrategy. Prompts guide you to change the strategy from SimpleStrategy to NetworkTopologyStrategy when necessary.

About this task

To configure and start the Repair Service:

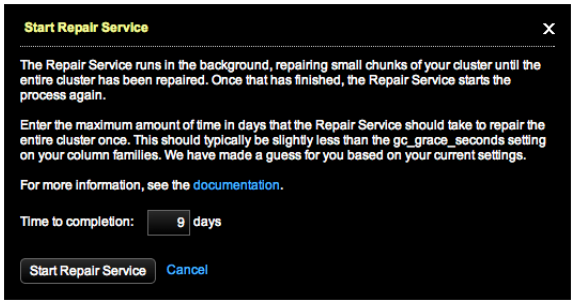
Procedure

1. In the left navigation pane, click **Services**.
2. Click **Configure** for the Repair Service.



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.



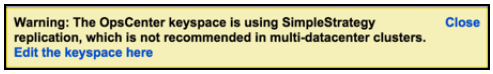
4. Click **Start Repair Service**.
   The Repair Service starts and updates its status after one minute and every minute thereafter.

Changing the OpsCenter keyspace

Running the Repair Service in a multi-datacenter environment requires using the NetworkTopologyStrategy.

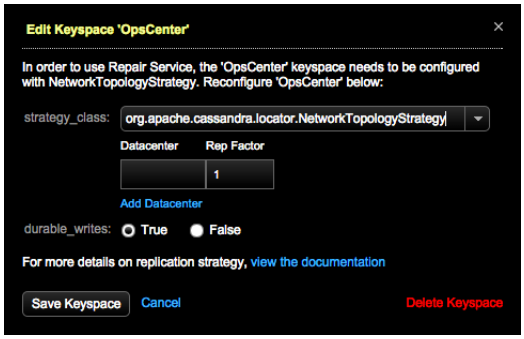
About this task

Running the Repair Service in a multi-datacenter environment requires using the NetworkTopologyStrategy. The OpsCenter keyspace defaults to SimpleStrategy. The Repair Service will not run with SimpleStrategy replication. Prompts guide to change the keyspace to a compatible replication strategy.

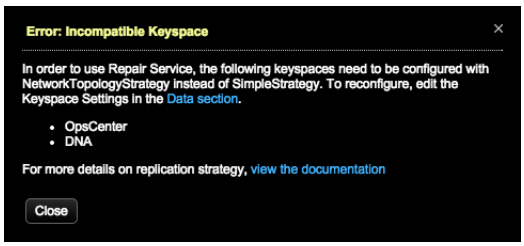


Procedure

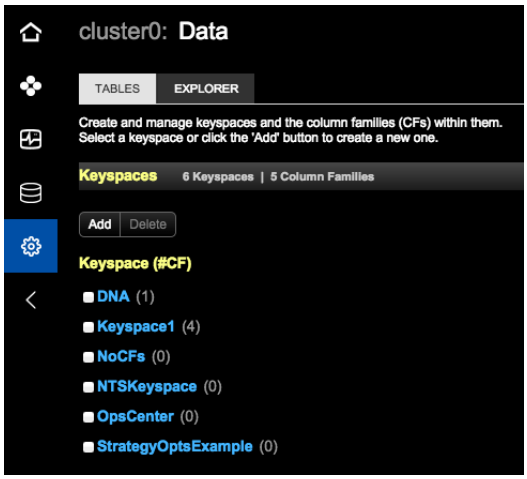
1. If the OpsCenter keyspace warning is visible when you first launch OpsCenter, expand the warning and click the link to edit the keyspace.
   The Edit Keyspace *keyspace_name* dialog for the keyspace appears. The dialog also appears if you attempt to start the Repair Service despite the warning.



2. Change the strategy from SimpleStrategy to NetworkTopologyStrategy.
3. Enter the datacenters and replication factor for each.
4. Click **Save Keyspace**.
5. If there are multiple incompatible keyspaces, the Error: Incompatible Keyspace dialog appears after clicking Configure for the Repair Service.



6. Click the **Data section** link to jump directly to the Data section to edit the keyspaces.



7. Repeat the above steps as necessary until all keyspace warnings disappear.

Stopping the Repair Service

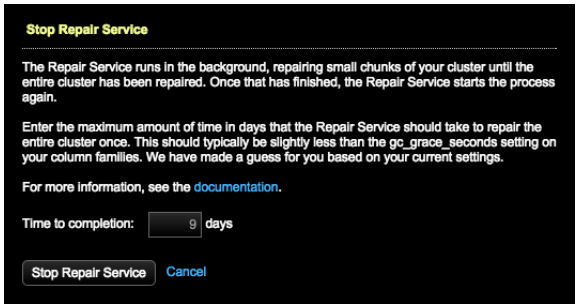
Stopping the Repair Service stops the service after any repairs in progress complete the currently running repair cycle.

About this task

Stopping the Repair Service stops the service after any repairs in progress complete the currently running repair cycle.

Procedure

1. In the left navigation pane, click **Services**.
2. Click **Configure** for the Repair Service.



3. Click **Stop Repair Service**.

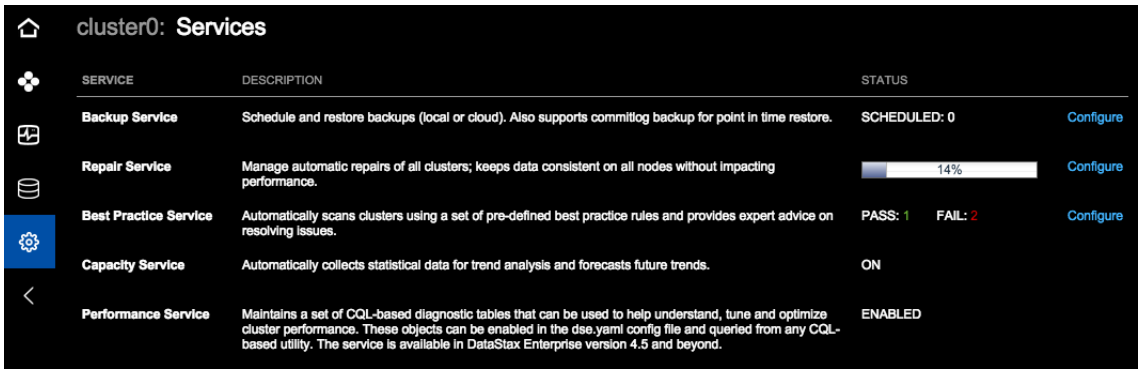
   Any repairs that were running when the repair service was stopped continue running until completion. Repairs in progress are not cancelled.

Checking repair progress

View progress of the current Repair Service cycle in the Services section of the OpsCenter UI.

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.



Logging

All Repair Service activity is logged to by default to:

Package install location

`/var/log/opscenter/repair_service/<cluster_name>.log`

Tarball install location

`<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

Reference of available configuration options for the Repair Service. Set the configuration options in either opscenterd.conf or cluster_name.conf. The settings in cluster_name.conf override settings in opscenterd.conf.

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] log_directory

The location in which to store repair service logs. The default location is `/var/log/opscenter/repair_service/` for package installations and `install_location/log/repair_service` for tarball installations.

[repair_service] log_length

Logs will rotate after the specified number of bytes. Defaults to 10485760 (10MB).

[repair_service] max_rotate

The maximum number of logs to retain. The default is 10.

[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. The default value is 0.

[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] repair_estimation_factor

Estimated reduced efficiency due to other issues like 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

Whether incremental repairs should do subrange repair or full repair of a node's entire range.

[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.

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

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
- Windows installations: Program Files (x86)\DataStax Community\opscenter\conf\clusters*cluster_name.conf*

Troubleshooting Repair Service errors

Errors encountered when running the Repair Service. Adjust repair service configuration options to resolve the errors.

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.

To resolve errors, try adjusting the configuration options in the [repair_service] section. 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_error_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:

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
/etc/opscenter/opscenterd.conf	X	X		

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
install_location/conf/opscenterd.conf			X	X

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
- Windows installations: Program Files (x86)\DataStax Community\opscenter\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.

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 metrics, network metrics), at both the node and column-family level (where applicable). These metrics are stored in Cassandra on the cluster being managed by default. That 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 column family, as well as aggregates across the entire cluster.

Forecasting

Using the Forecast feature, view a predicted trend for any metric, based on historical data that has been collected.

Using forecasting

Use forecasting to predict trends in graphs based on past performance.

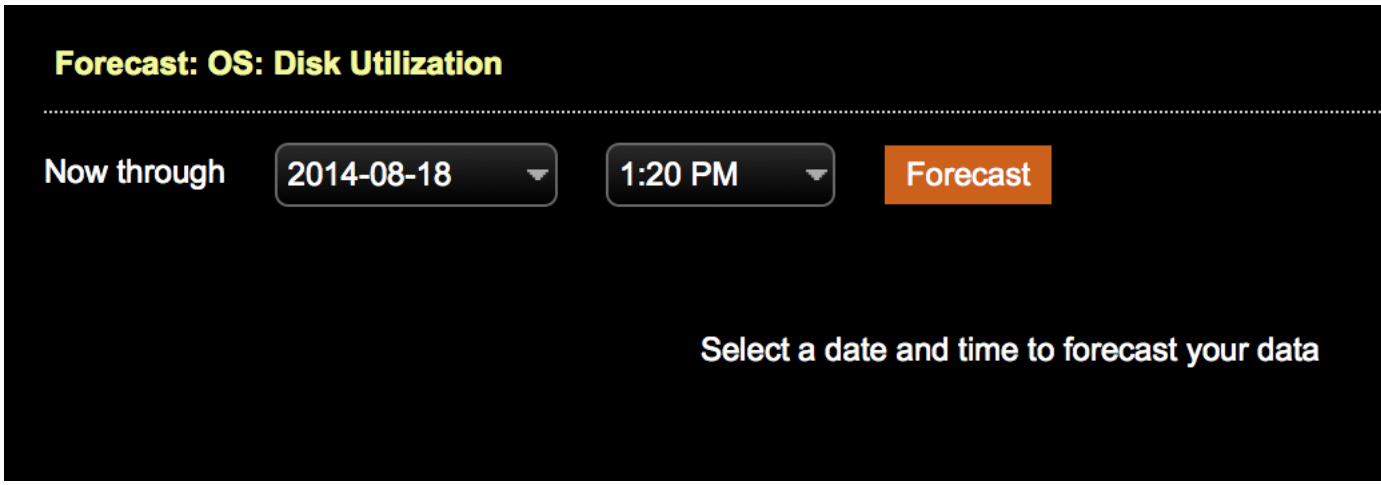
About this task

Use forecasting to predict trends in graphs based on past performance.

Procedure

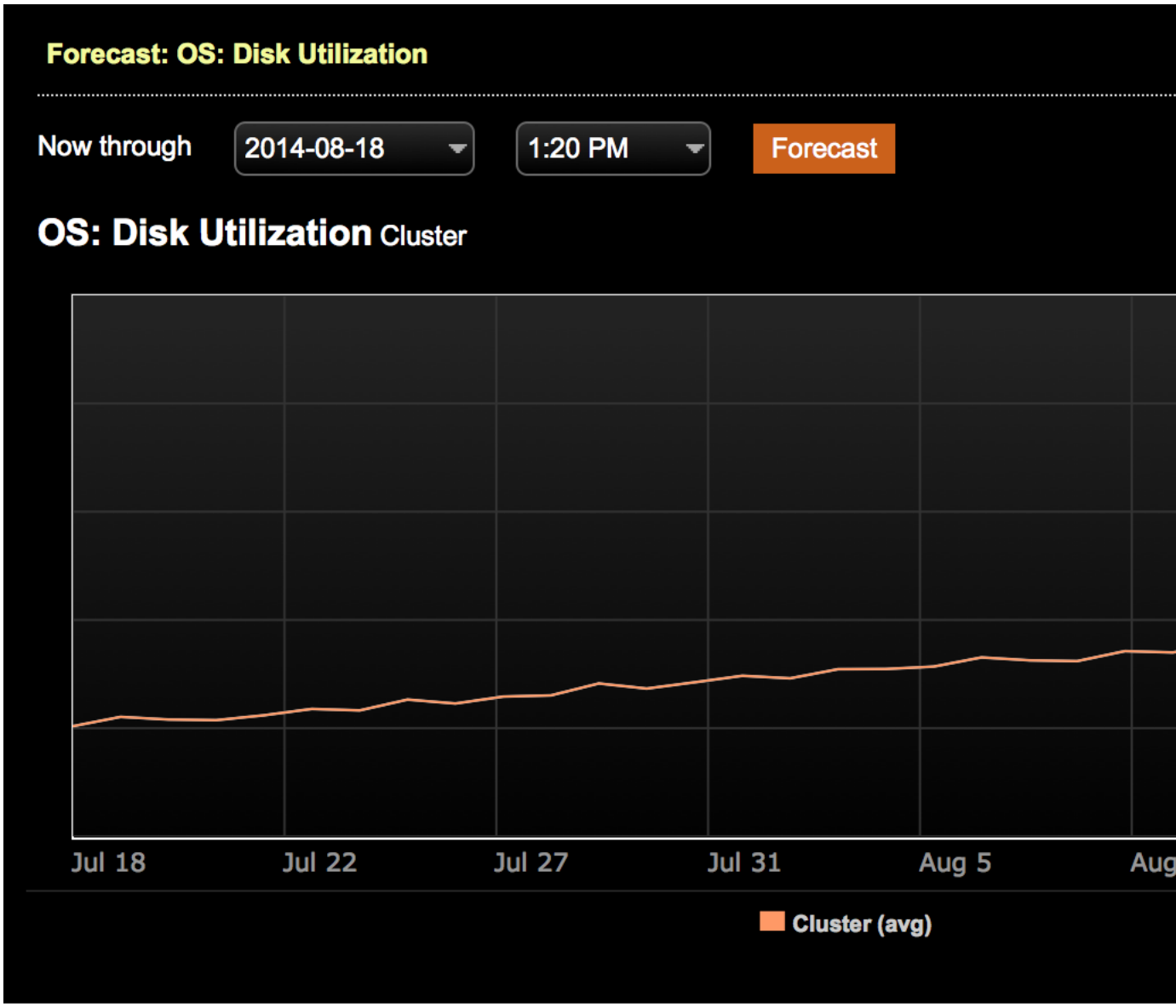
1. In the **Dashboard** section of the OpsCenter web interface, locate the graph you'd like to forecast and click the drop-down on the upper right and then click **Forecast**.
2. In the **Forecast** dialog, enter the date and time you'd like to use for the end of the trend prediction.

   The end date and time must be a minimum of two days into the future and a maximum of one year.



3. Click the **Forecast** button.

   A predicted trend using polynomial curve fitting against historical data displays.



The forecast above shows that our node will be at 60% disk usage in a month, so we should probably start thinking about adding capacity now.

Advanced configuration of forecasting

Configure advanced forecasting options in opscenterd.conf.

The following is currently configurable by adding a `[forecasting]` section to the opscenterd.conf file.

`polyfit_degree = 2`

Which degree polynomial equation to use in calculating the forecasting. (The default is three.)

`range_multiplier = 3`

In order to generate a meaningful prediction, historical data is analyzed for a period larger than the range being forecasted. (The default is three times larger than the range being forecasted.) For example, to forecast one month into the future, three months of data is analyzed.

`required_data_percentage = 0.5`

The percentage of the historical data required to be available. (The default is 50%.)

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

Best Practice service

The Best Practice service allows you to schedule pre-defined best practice rules that check various properties of clusters and environments and report back which nodes pass or fail the rules.

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 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. 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.

About this task

Best Practice service rules can be enabled, scheduled, and disabled from OpsCenter.

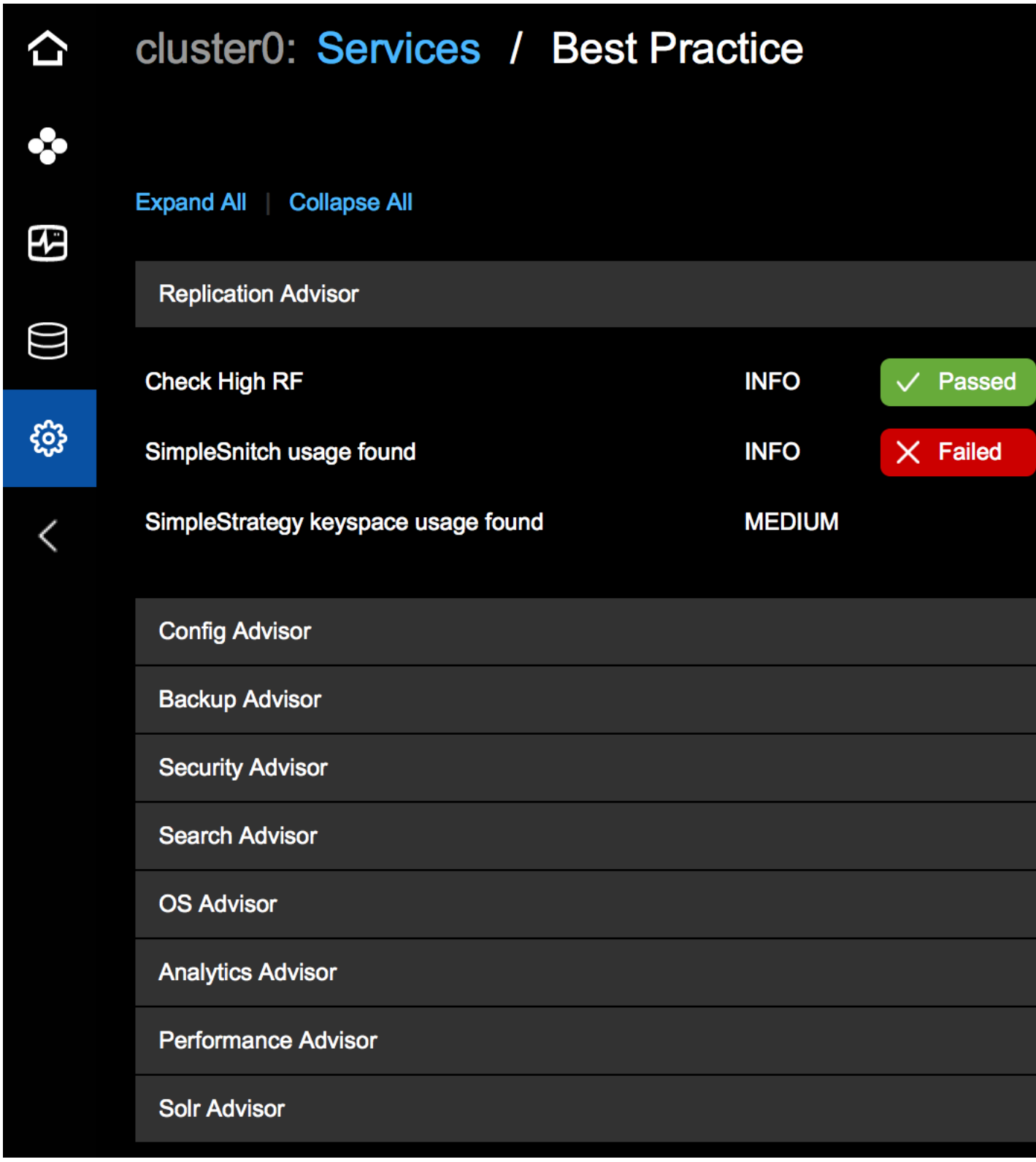
Procedure

1. In the OpsCenter UI, hover over the cluster name on the left and click the **Services** icon.
2. Click **Configure** to the right of 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) Choose a date and time to start the rule, and indicate how often the rule should run.
   c) Click **Configure Rule**.
5. To disable a rule, click **Turn Rule Off**.
6. To change when the schedule for when a rule is run:
   a) Click Configure to the right of the rule.
   b) Modify the date and time to start the rule and how often it should run.
   c) Click **Configure Rule**.

Monitoring the results of Best Practice service scans

The results of a scan by the Best Practice service displays as a number of nodes that have passed or failed a rule.

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 number of rules that have passed and failed displays on the top right. Clicking the Pass or Fail number filters the rules to only display the rules 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.

Data modeling

Create and manage keyspaces and the column families within keyspaces.

Keyspaces

Add and manage keyspaces.

Click **Data** in the left pane to list the keyspaces in the cluster that you are monitoring.

When you create a keyspace, name the keyspace, choose a replica placement strategy, indicate the total number of replicas you want and how those replicas are divided across your data centers (if you have a multiple data center cluster).

Creating a keyspace

Follow these steps to add a keyspace to a cluster. Optionally, you can also create a column family for the keyspace.

About this task

To add a keyspace to a cluster:

Procedure

1. Click **Data** in the left navigation pane, then **Add** under Keyspaces.
2. Give the keyspace a name. Keyspace names should not contain spaces or special characters or exceed the filename size limit of your operating system (for example, 255 bytes on most Linux file systems). Keyspace names are case sensitive.
3. Set the replica placement strategy. The replica placement strategy (along with the cluster-configured snitch) determines how replicas are placed on nodes throughout the cluster. Select one of the following built-in replica placement strategies:
   a) **SimpleStrategy**: (Default) Single datacenter, rack-unaware replica placement.
   b) **NetworkTopologyStrategy**: (Recommended) Single or multiple datacenters, rack-aware replica placement.
   c) **OldNetworkTopologyStrategy**: (Deprecated) Two datacenters only, rack-aware replica placement.
4. Enter how many total copies that you want of your keyspace data in the **replication factor** field. The `NetworkTopologyStrategy` requires configuring replicas per datacenter. The datacenter name should match the datacenter name used by the cluster-configured snitch. Make sure to name the datacenter(s) correctly according to the snitch configuration.
5. If you do not want to start defining column families within the new keyspace right away, uncheck the **I would like to create a Column Family** check box.
6. Click **Save Keyspace**.

Managing a keyspace

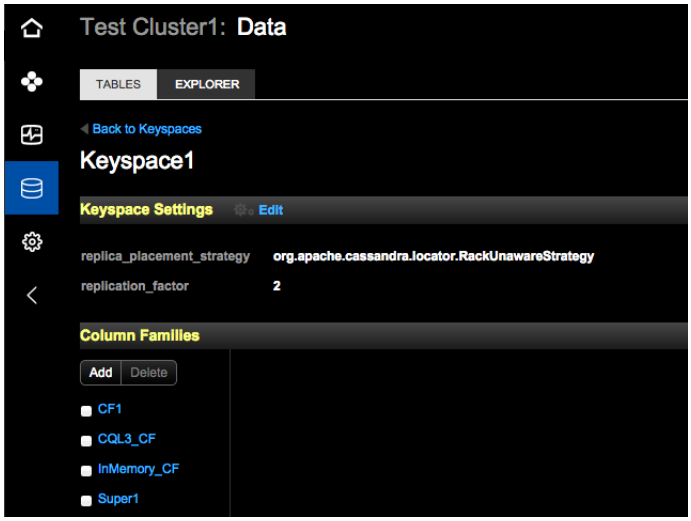
Edit keyspace settings or delete a keyspace.

About this task

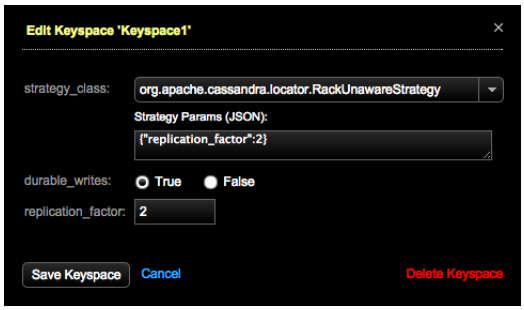
To manage a keyspace:

Procedure

1. Click **Data** in the left navigation pane.
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.



4. To delete a keyspace, click **Delete Keyspace**.
5. To view column family properties, select a column family under Column Families. For more information, see Managing a column family.

Managing column families

When you create a column family in Cassandra using an application, the CLI, or CQL 2 or earlier, the column family appears in OpsCenter.

When you create a column family in Cassandra using an application, the CLI, or CQL 2 or earlier, the column family appears in OpsCenter. You can also create one type of column family: the dynamic column family. Dynamic column families are those that do not specify column names or values when the column family is created. An application typically supplies this metadata. CQL 3, the default query language in Cassandra, does not support dynamic column families. Earlier versions of CQL and the CLI support dynamic column families.

In-memory tables are indicated next to the column family name in the details section.

This version of OpsCenter does not support defining static column families (per-column metadata), row key data types, or schema information for super column sub-columns described in Cassandra 1.0, or earlier.

Creating a dynamic column family

About this task

To create a new dynamic column family:

Procedure

1. Click **Data** in the left pane.
2. From the list of keyspaces, select the keyspace to contain the column family.
3. Under Column Families, click **Add**.
4. Give the column family a name.

   Column family names should not contain spaces or special characters and cannot exceed the filename size limit of your operating system (255 bytes on most Linux file systems).

   By default, column families are created with standard columns (**column_type: Standard**). If you want a column family containing super columns, choose **column_type: Super**.
5. Use **comparator_type** to set the default data type for column names (or super column names).

   Setting the default data type also sets the column sort order for the column family. For example, choosing **LongType** sorts the columns within a row in numerical order. The sort order cannot be changed after a column family is created, so choose wisely.
6. Use **default_validation_class** to set the default data type for column values (or super column sub-column values). Always set this for dynamic column families.
7. Click **Save Column Family**.

Managing a column family

Add or delete a column family in a keyspace, view metrics for a column family, or truncate (delete) data from a column family without deleting the column family (table).

About this task

To manage a column family:

Procedure

1. Click **Data** in the left navigation pane.
2. From the list of keyspaces, select a keyspace.

   The #CFs columns shows how many column families each keyspace contains.
3. From the list of the column families, select a column family.

   Click one of the following buttons:

   - **Add**: See Creating a dynamic column family.
   - **Delete**: Completely removes the column family from the keyspace. You can select more than one column family in a keyspace to delete.
   - **View Metrics**: Presents metrics for a column family. In the Metric Options dialog, select a column family (CF) 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 column family to measure.
   - **Truncate**: Deletes all data from the column family but does not delete the column family itself.

     Note: Removal of the data is irreversible.
4. When you select a column family, a list of manageable attributes appears: Properties (fully editable), Metadata (Add or Delete), and Secondary Indexes (Add or Delete).

Browsing data

The Data Explorer feature in OpsCenter is in the process of being deprecated in favor of DataStax DevCenter, a visual CQL tool.

Note: The Data Explorer feature in OpsCenter is in the process of being deprecated in favor of DataStax DevCenter, a visual CQL tool. You can find more information about DevCenter and a link to download at http://www.datastax.com/what-we-offer/products-services/devcenter.

Browsing a Cassandra database

The Data Explorer feature in OpsCenter is in the process of being deprecated in favor of DataStax DevCenter, a visual CQL tool.

About this task

Note: The Data Explorer feature in OpsCenter is in the process of being deprecated in favor of DataStax DevCenter, a visual CQL tool. You can find more information about DevCenter and a link to download at http://www.datastax.com/what-we-offer/products-services/devcenter.

To browse a Cassandra database:

Procedure

1. Click **Data** in the left hand pane, then click the **Explorer** tab.
   A drop-down list of keyspaces in the cluster appears. By default, the list includes the OpsCenter keyspace, which contains column families of data in the cluster.
2. Select one of the keyspaces. For example, click the OpsCenter keyspace.
   The list of column families appears.
3. Click a column family. For example, click `events_timeline` and expand the OpsCenter console window so you can see more values.
   A row keys, columns, and data values of the `events_timeline` column family appear in tabular format. You may notice that some data values are hex-encoded, not human-readable values, and other values are perfectly readable. Internally, Cassandra stores row key names, column names and column values as hex byte arrays. If possible, OpsCenter converts data to text you can read.
4. Click the heading row of the table to see the sort control (the arrow on the right). Toggle the sort control to sort the rows in ascending or descending order.
5. To browse through the data, use the scroll bar or click Next to page through the data.

Examining row data

The Data Explorer feature in OpsCenter is in the process of being deprecated in favor of DataStax DevCenter, a visual CQL tool.

About this task

Note: The Data Explorer feature in OpsCenter is in the process of being deprecated in favor of DataStax DevCenter, a visual CQL tool. You can find more information about DevCenter and a link to download at http://www.datastax.com/what-we-offer/products-services/devcenter.

To zoom in on a particular row:

Procedure

1. Assuming the `events_timeline` column family is still selected from browsing a database, type the key for a row in the **Key** text entry box.

   For example, type the row key `timeline_start`. If you set up a secondary index on the column family, the Key menu lists key choices.
2. Click the search icon.
   The columns and values of the row appear.

Removing all traces of Cassandra or DSE packages

After a failed attempt to provision a node or a cluster, you must remove all traces of Cassandra or DSE packages from the affected nodes.

About this task

After a failed attempt to provision a node or a cluster, you must remove all traces of Cassandra or DSE packages from the affected nodes. If a provisioning task failed, use the following script to remove all packages from the node.

Procedure

1. Log into each node.
2. Run the following BASH script on each node using `sudo`.

```
#!/bin/bash

# Stop services
/etc/init.d/cassandra stop
/etc/init.d/dse stop
/etc/init.d/datastax-agent stop

# Remove packages
PACKAGES=(dsc dsc1.1 dsc12 cassandra apache-cassandra1 dsc-demos \
dse-libspark \
dse dse-libhadoop-native dse-libhadoop dse-libcassandra dse-hive dse-
libhive dse-pig \
dse-libpig dse-demos dse-libsqoop dse-libtomcat dse-liblog4j dse-libsolr
 dse-libmahout dse-full)
DEB_PACKAGES=(python-cql python-thrift-basic)
RPM_PACKAGES=(python26-cql python26-thrift)
if [ `which dpkg` ]; then
PLIST=(${PACKAGES[@]} ${DEB_PACKAGES[@]})
dpkg -P ${PLIST[*]}
rm -rf /etc/apt/sources.list.d/datastax.list
else
PLIST=(${PACKAGES[@]} ${RPM_PACKAGES[@]})
yum -y remove ${PLIST[*]}
rm -rf  /etc/yum.repos.d/datastax.repo
fi

# Cleanup log and configuration files
rm -rf /var/lib/cassandra/* /var/log/{cassandra,hadoop,hive,pig}/* /etc/
{cassandra,dse}/* \
/usr/share/{dse,dse-demos} /etc/default/{dse,cassandra}
```

- copy, paste, and run the script from the command line
- save the script to a local file and copy to each node to run from the command line

Troubleshooting

This section lists some issues experienced with OpsCenter and solutions or workarounds.

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.

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

Increasing the nodelist polling period can reduce CPU usage when 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
- Windows installations: Program Files (x86)\DataStax Community\opscenter\conf\clusters*cluster_name.conf*

About this task

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+.

Procedure

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. Optional: 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. Optional: Refresh the browser.

Configuring a sleep delay for CPU issues when starting opscenterd

Configuring a delay between clusters during startup helps alleviate opscenterd CPU usage on startup, allowing OpsCenter to function properly.

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

About this task

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+.

Procedure

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 validation for self-signed certificates

Set ssl_validate to False if you use a self-signed certificate and experience difficulty connecting to a Cassandra 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`
- Windows installations: `Program Files (x86)\DataStax Community\opscenter\conf\clusters\`*`cluster_name.conf`*

About this task

Set `ssl_validate` to `False` if you use a self-signed certificate and experience difficulty connecting to a Cassandra cluster.

Procedure

1. Open cluster_name.conf for editing.

```
[cassandra]
username =
ssl_ca_certs = .cer file location
ssl_validate = False
```

2. Restart OpsCenter.

Zero nodes detected in cluster

Workaround for browser environments where the OpsCenter UI is unable to establish a persistent streaming connection to opscenterd.

About this task

Some environments may 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. Follow the workaround steps to resolve the issue:

opscenterd.conf

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

Location	Package	Installer (GUI or text mode)		Tarball
		Service	No-service	
`/etc/opscenter/opscenterd.conf`	X	X		
`install_location``/conf/opscenterd.conf`			X	X

Procedure

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

Currently supported browsers include Chrome, Firefox, and Safari. IE is not supported at this time.

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

The SSTables in this snapshot '<tag>' are not compatible

Instructions to upgrade the OpsCenter snapshot.

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 Cassandra or DSE.

How to

1. Log in to each node.
2. Run the `sstableupgrade` script for every keyspace and column family you need to restore; passing it the keyspace, column family, and OpsCenter snapshot tag received from the error message.

   How you run the script depends on how you installed Cassandra or DSE.
3. Retry the restore from OpsCenter.

OpsCenter data growing too large

For OpsCenter versions prior to 3.2.1, truncate column families to reclaim space.

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 column families in the OpsCenter keyspace to grow too large. The most common offenders typically are the `pdps` (raw data points) and `rollups60` (1m data points) column families.

If any of the column families have grown too large, you can truncate 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

Upgrade Python if you encounter any difficulty adding 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.

Error `exceptions.ImportError:libssl.so.0.9.8`

Fix for the error when OpenSSL is installed on Linux distributions.

Occurs when OpenSSL 1.0.0 is installed on RHEL 5.x, CentOS 5.x, OEL 5.5, Debian, or Ubuntu systems:

```
message.exceptions.ImportError: libssl.so.0.9.8
```

To fix, you can do either of the following:

- Install OpenSSL 0.9.8:
  1. RHEL-based systems: `sudo yum install openssl098`
  2. Debian-based systems: `sudo apt-get install libssl0.9.8`
- Install Python libssl 1.x:
  1. Remove all OpenSSL modules from the OpsCenter installation `lib` directory. You do not have to remove them globally.

     Packaged installs: `/usr/share/opscenter/content/lib`

     Tarball installs: *`install_location`*`/lib`

You can easily find these directories using:

```
find . -name OpenSSL -type d | xargs rm -rf
```

2. Install the latest pyOpenSSL module:

   Debian-based systems: `apt-get install python-openssl`

   RHEL-based systems: `yum install python-openssl`
3. Start or restart opscenterd.

Python used to run OpsCenter not built with SSL

Resolve the error by compiling Python with SSL support.

To protect your AWS credentials when launching EC2 instances, OpsCenter needs to use HTTPS. If the version of Python that is running opscenterd was not compiled with SSL support, OpsCenter will not run even if SSL has been disabled in the configuration file.

To resolve the issue, first ensure that OpenSSL 0.9.8 is installed on your system. If you compiled Python manually, it is recommended that you install Python 2.6+ through your package manager. On CentOS and RedHat Enterprise Linux, this is most easily done through EPEL packages.

If you must compile Python manually, make sure that SSL support is enabled. This blog post explains the process for Python 2.5, but the same steps should work for Python 2.6 or 2.7.

DataStax agent port setting conflict

Ensure there are no conflicts with port 7199 used by the DataStax Agent.

If you have a problem with OpsCenter, check for conflicts in port settings. 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 this port.

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, intermittent problems can be expected.

Limiting the metrics collected by OpsCenter

Limit monitored metrics to conserve disk space.

If a cluster keyspace has many column families, the number of metrics OpsCenter collects can become quite large. For information about reducing the number keyspaces or column families that are monitored, see Controlling data collection.

Java not installed or JAVA_HOME environment variable not set

Error when Java is not installed or the path is 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

Refer to the recommended settings for insufficient user resource limits errors in the Cassandra documentation.

Insufficient resource limits may result in an insufficient nofiles error:

```
2012-08-13 11:22:51-0400 [] INFO: Could not accept new connection  (EMFILE )
```

See Recommended settings under Insufficient user resource limits errors in the Cassandra documentation.

Installing EPEL on CentOS 5.x or RHEL 5.x

EPEL is a prerequisite for OpsCenter 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). EPEL contains dependent packages, such as Python 2.6+.

To install for both 32- and 64-bit systems:

```
$ sudo rpm -Uvh http://dl.fedoraproject.org/pub/epel/5/i386/epel-
release-5-4.noarch.rpm
```

Note: You do not have to install EPEL on other machines.

Problems with provisioning

Troubleshooting tips for typical provisioning issues.

General troubleshooting steps

Basic troubleshooting for resolving errors include checking firewalls, logs, and prior installs.

Ensure firewalls are properly opened between the opscenterd machine and each node.

Check the following files on any of the nodes having problems for any errors:

- `/var/log/cassandra/system.log`
- `/var/log/datastax-agent/agent.log`

Verify that Cassandra (or DSE) was not previously installed on any of the machines; if it was, all binaries, configuration files, logs, etc. must be cleared out first.

Invalid repository credentials

Error messages encountered if using invalid credentials when installing DSE.

Invalid repository credentials

Debian

When installing DSE, if you enter invalid values for DataStax Credentials, an error dialog displays text along the lines of:

```
Installation stage failed: Installation failed on node 172.16.1.2: 'apt-get
update' failed.
```

Clicking **Details** displays the entire output from both stdout and stderr. If, in this output, you see the "401 Unauthorized", it means that the credentials entered were invalid.

RHEL

When installing DSE, if you enter invalid values for DataStax Credentials, an error dialog displays text along the lines of:

```
Installation failed on node 172.16.1.2: 'yum install' failed.
```

Clicking **Details** displays the entire output from both stdout and stderr. If, in this output, you see "The requested URL returned error: 401", it means that the credentials used were invalid.

Timed out waiting for Cassandra to start

Check the Cassandra system log for timeout errors.

If you receive this error, it most likely means that the Cassandra process failed to start on one or more nodes. You should look in `/var/log/cassandra/system.log` for any errors that may need to be resolved.

The following packages are already installed

Error that indicates Cassandra or DSE is already installed on the system. Before provisioning a new cluster in OpsCenter, remove any instances of Cassandra or DSE.

If you receive an error that starts with this message, it means Cassandra (or DSE) is already installed on the system. OpsCenter provisioning requires that any instances of Cassandra (or DSE) be completely removed or purged before provisioning a new cluster.

Removing all Cassandra or DSE files after failed provisioning

Prior to retrying provisioning, remove Cassandra or DSE.

If you provision a new cluster or add a node to an existing cluster via the OpsCenter UI and an error occurs, you can retry the request, but before you retry you must remove all traces of Cassandra or DSE packages from the nodes in the cluster. You can also remove simply remove the files if you do not wish to retry the provisioning.

Agents cannot connect to opscenterd

Investigate and resolve the installed agent is unresponsive error.

If you receive an error message that includes "The installed agent doesn't seem to be responding", there is most likely a firewall issue preventing the installed agent from connecting to the opscenterd machine. You should ensure that port 61620 is open on the opscenterd machine and check the agent logs.

OpsCenter cannot create a local cluster using a public IP address

Use a private IP address if you encounter difficulty creating a local cluster with a public IP address.

Error that occurs when attempting to create a local cluster using a public IP address:

```
Error Installation Stage Failed: The installed agent doesn't seem to be
responding
```

If you encounter any difficulty creating a local cluster with a public IP address:

- Providing the EC2 public hostname instead of the IP address results in successful creation of the cluster.
- Using a private IP address when provisioning on GCE results in successful creation of the cluster.

Note: Using a private IP address is recommended. In cases where communication is possible by both public and private IP addresses, private IP addresses are usually preferable. Private IP addresses often have more direct communication paths, which results in fewer addressing problems, better performance, and lower costs in situations where network traffic might be metered.

Problems running `sstableloader`

Workarounds for sstableloader issues on older versions of Cassandra and DSE.

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:

```
$ sudo killall sstableloader
```

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 DSE 4.0.x.

Timeout connecting to Cassandra 2.0 clusters

Change settings in cassandra.yaml to fix a Cassandra connection timeout in OpsCenter.

OpsCenter will timeout when connecting to Cassandra 2.0 clusters when `rpc_server_type=hsha` in `cassandra.yaml`.

Due to a bug in Cassandra 2.0 (CASSANDRA-6373), the connection from `opscenterd` over `thrift` will hang, causing a timeout in OpsCenter. The workaround is to change the `rpc_server_type` setting to `rpc_server_type=sync` in `cassandra.yaml`.

Sophos Web Protection breaks browser access to OpsCenter on Windows

Add a firewall rule allowing access to port 8888 or disable Web Protection to fix the browser access issue.

Sophos Web Protection prevents the OpsCenter UI from loading. The JavaScript console for the web browser will show 404 errors, and the interface will not load. Adding a firewall rule in Web Protection allowing access to port 8888 or disabling Web Protection fixes this issue.

Error getting version update information

Disables the check for the latest versions of OpsCenter and DSE.

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 DSE 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 API reference

The OpsCenter API facilitates the development of websites and programs to retrieve data and perform Cassandra administrative actions. The OpsCenter API includes RESTful requests for programmatically performing the same set of operations as the OpsCenter GUI.

OpsCenter Release Notes

Information about new features and resolved issues in the following OpsCenter releases.

OpsCenter 5.1.3 Release Notes

Release notes for the OpsCenter version 5.1.3 release.

Resolved issues

- Fixed Repair Service to no longer complain that the `system_traces` keyspace is using SimpleStrategy. (OPSC-5408)
- Fixed agent startup issue in some cases for clusters using JMX authentication. (OPSC-5426)

Compatibility

To see which versions of OpsCenter are compatible with the various Cassandra and DataStax Enterprise versions, see the OpsCenter Compatibility chart.

OpsCenter 5.1.2 Release Notes

Release notes for the OpsCenter version 5.1.2 release.

Release notes for the OpsCenter version 5.1.2 release include many major improvements and resolved issues.

Support for DSE version 4.7

- Improved management and monitoring capabilities of DSE In-Memory tables. (OPSC-4685, OPSC-4688).
- Monitoring alerts for off-site key storage in DSE Encryption. (OPSC-4695, OPSC-4697)
- Ability to load the Spark Management interface directly from OpsCenter. (OPSC-4408)
- Support for the DataStax agent on Java 8.
- Properly indicate whether the Performance Service is enabled. (OPSC-3251)

Dashboard Metrics

- Inline metrics explanations appear when hovering over graph legends or metrics in the Metric menu when creating or editing graphs. (OPSC-1816)
- Metrics for read latency and write latency now allow viewing percentiles (50th, 90th, 99th) rather than averages, which presents a more accurate view of data. These percentiles are now the default in place of previous latency graphs. (OPSC-2673)
- Added SSTable read histogram metric. (OPSC-641, OPSC-3954)
- Added historic tpstats and dropped messages. (OPSC-2432)
- Added read repair metrics. (OPSC-2652)
- Ability to select multiple nodes when editing a graph. (OPSC-3958)
- Fixed an issue with permissions that would cause the UI to hang for non-admins in some cases. (OPSC-4912)
- Fixed an issue that displayed empty graphs. (OPSC-4225)
- Fixed an issue that overly restricted metric choices when editing graphs. (OPSC-4250)
- Fixed an issue with stacked graphs reporting incorrect values when displaying results from multiple nodes. (OPSC-4616)

DSE Management - Repair Service

Repair Service improvements:

- When applicable, proactively prompt users to change the replication strategy on the OpsCenter keyspace in a multi-datacenter environment. (OPSC-2231)
- Snapshot repairs have been disabled by default. (OPSC-4692)
- When nodes go down, changed the behavior to skip those nodes rather than restarting the Repair Service. (OPSC-3672)
- Before issuing a new subrange repair, check for too many running repairs. (OPSC-3700)
- Support for incremental repairs in Repair Service. (OPSC-4823)
- Improved validation and error handling/messages. (OPSC-2482, OPSC-2595, OPSC-3684)

DSE Management - Backup Service

Backup Service improvements:

- No longer require Edit Connection Settings permissions to enable Commitlog Archiving. (OPSC-4342)
- Show active restore or restores as a notification when the UI loads. (OPSC-2810)
- Ability to configure rolling restart parameters when initiating a Point-in-Time restore. (OPSC-4513)
- Improved error handling when `schema.json` fails to upload to Amazon S3. (OPSC-4486)
- Removed background polling of last commitlog archival in favor of retrieving on user request. (OPSC-4268)
- Fixed an issue with stored schemas containing redundant index definitions. (OPSC-4781)
- Fixed an issue that prevented all backups from being shown to the user. (OPSC-5068)
- Fixed an issue with status not respecting collapsed state after update. (OPSC-4378)
- Fixed an issue with some backups displaying 100% complete before actual completion. (OPSC-4382)
- Fixed a minor pagination issue on the Activity tab. (OPSC-4559)

Other miscellaneous improvements

- Support for CentOS/RHEL 7.
- Ability to configure a custom display name for clusters. (OPSC-2227)
- Ability to configure an alias for nodes in `address.yaml`. (OPSC-789)
- Node uptime since DSE was last started is now displayed in the Node Details dialog. (OPSC-2724)
- Ability to add a Spark Node to a vnode Cluster (Spark supports vnodes). (OPSC-3399)
- Allow disabling poll of ec2 api on agents. (OPSC-3489)
- Reduced noise from failed DSE/Cassandra connections in `agent.log`. (OPSC-3903)
- Obscure keystore password and truststore password in Cluster Configure dialog. (OPSC-4882)
- Improved connection time for agents in multi-datacenter environments. (OPSC-4166)
- Minor improvements to login page behavior. (OPSC-4617)

Other resolved issues

- Fixed race condition when sending config from opscenterd to the agent, which removes the need to set all properties in `address.yaml`. (OPSC-4667)
- Fixed UI not loading on newer Windows machines and other touch-enabled devices. (OPSC-5017)
- Error getting latest software versions when https is enabled. (OPSC-3923)
- Fixed the inability to delete roles. (OPSC-4799)
- Removed warnings about `system_traces` replication strategy. (OPSC-3964)
- Installing the agent through OpsCenter fails when installing or provisioning on the same machine as opscenterd. (OPSC-4563)
- Edit Cluster Settings dialog always returns `ssl_validate` as true. (OPSC-4894)
- Fixed issue with the process for managing and existing cluster timing out in some cases. (OPSC-5145)

- Fixed scrolling issues in the Configure Cluster dialog. (OPSC-4125)
- Added more logging for troubleshooting stream and status progress when showing > 100%. (OPSC-3383, OPSC-3770)
- For job tracker proxy, opscenterd uses `listen_address` instead of `rpc_address`. (OPSC-1678)
- Removed log4j appender warning on datastax-agent startup. (OPSC-1899)
- Cannot add a node to a cluster if `num_tokens` is explicitly set to 1 on existing nodes. (OPSC-2330)
- Fixed unfriendly stack traces being displayed in some UI notifications. (OPSC-2948)
- Configure submenu is visible to unprivileged users when there are multiple DCs. (OPSC-4471)
- Spelling mistake in `opscenterd.log` messages. (OPSC-4615)

Known issues

- **Backup Service:**
  - Automatic schema recreation is not currently supported during a restore when using User Defined Types (UDTs). (OPSC-4866)
  - Due to a bug in Cassandra (CASSANDRA-9129), point-In-time restore cannot restore commitlogs for keyspaces or tables that would have to be recreated in Cassandra 2.1+ and DSE 4.7+. Also, point-in-time restore fails if any tables were recreated during the time period of the actual point-in-time restore. (OPSC-5081)

Compatibility

To see which versions of OpsCenter are compatible with the various Cassandra and DataStax Enterprise versions, see the OpsCenter Compatibility chart.

OpsCenter 5.1.1 Release Notes

Release notes for the OpsCenter version 5.1.1 release.

Improvements

- Support for connecting to a separate DSE storage cluster with SSL enabled. (OPSC-1336)
- OpsCenter only stops the Repair Service based on real rather than estimated low throughput. (OPSC-2073)

Resolved issues

- Fixed error with restoring from the Backup Service in certain schemas. (OPSC-4587)
- Agent no longer throws `OutOfMemoryError` when improperly trying to connect to a cluster that has authentication enabled. (OPSC-4699)
- Workarounds to rectify prohibitively high CPU usage when opscenterd runs or starts up managing multiple clusters. (OPSC-4421)
- Fixed rolling restart errors persisting in the UI by allowing users to dismiss the error. (OPSC-4023)

Known issues

- A bug in the Edit Cluster Settings dialog always sets Validate SSL Certificates to enabled (true), which can cause the connection to Cassaandra to fail if you are using a self-signed certificate. See the workaround for Troubleshooting SSL validation for self-signed certificates. (OPSC-4894)
- Agents do not connect automatically if Cassandra is not listening on 127.0.0.1. If the `rpc_address` property in `cassandra.yaml` is configured to an IP address other than 127.0.0.1 (localhost) or 0.0.0.0, you must configure the hosts property in `address.yaml` for the agent on each node (e.g., hosts: ["55.200.55.200"]). (OPSC-4833)

Compatibility

To see which versions of OpsCenter are compatible with the various Cassandra and DataStax Enterprise versions, see the OpsCenter Compatibility chart.

OpsCenter 5.1.0 Release Notes

Release notes for the OpsCenter version 5.1.0 release.

New features

- **Backup Service** - The basic backup and restore functions of OpsCenter have been replaced by the Backup Service, which provides more enterprise-class backup and restore functionality.
- **Automatic failover** - When all monitored clusters are DSE, failover from the primary OpsCenter instance to the backup OpsCenter instance happens automatically without any manual intervention or downtime.
- **Improved agent user permissions** - For package installs, the DataStax agent now runs as the Cassandra user by default, which provides a smoother permissions experience out of the box.
- **Standalone installer** - There is now a standalone GUI installer for Mac OS X and a command line installer for Linux that makes installing and uninstalling OpsCenter fast and convenient.

Improved features

- **Email alerts** - The cluster name is now included in email alerts and you can customize the email subject.

Notable changes

- The default heap size of the DataStax agent has been increased from 40 MB to 128 MB.
- For OpsCenter to connect to your cluster, `start_native_transport` must be set to `true` on all of your nodes. (OPSC-4219)
- OpsCenter Authentication now uses sessions in place of HTTP Basic Auth.

  Note: If your application directly accesses the OpsCenter REST APIs, you need to update your code to authenticate using sessions.
- The `hosts` option in `address.yaml` now determines which nodes the agent connects to. For further information on configuration changes and migration paths, see the Upgrade Guide. (OPSC-4567)

Resolved issues

- Added support for `require_client_auth: true` in Cassandra client_encryption_options. (OPSC-3249)
- Removed support for SSL v3 due to POODLE vulnerability. (OPSC-3775)

Known issues

- **Backup Service**
  - To support uploading sstables larger than 75 GB to Amazon S3, some options will need to be tuned. See Configuring the Backup Service to upload very large files to Amazon S3. (OPSC-4479)
  - Real time progress of S3 backups and restores can be delayed in some cases.
  - Collapsing nodes on an in progress Backup or Restore dialog will automatically expand with updates (that is, collapsing does not work properly). (OPSC-4378)
  - Restoring from a failed backup is not currently possible. (OPSC-4175)
  - During a clone operation (restoring to a different cluster), there is a known UI-only issue where the restore entry in the backup activity log will only show up in the destination cluster of the clone. This issue arises when the restore progress dialog is closed. (OPSC-4304)

    - The Edit Connection Settings permission is required to enable Commit Log Archiving in conjunction with the backup service. (OPSC-4274)
- OpsCenter is unable to create a local cluster using a public IP address. Current workaround is to use a private IP address. See troubleshooting. (OPSC-3971)
- The log4j warnings and errors in the agent log when provisioning a node can be safely ignored. (OPSC-3903)
- The "Error getting version update information" message is displayed when internet access or the DataStax update URL is unavailable. You can ignore the error or disable the latest version check. (OPSC-3247)
- Agents do not connect automatically if Cassandra is not listening on 127.0.0.1. If the `rpc_address` property in `cassandra.yaml` is configured to an IP address other than 127.0.0.1 (localhost) or 0.0.0.0, you must configure the hosts property in `address.yaml` for the agent on each node (e.g., hosts: ["55.200.55.200"]). (OPSC-4833)

Compatibility

To see which versions of OpsCenter are compatible with the various Cassandra and DataStax Enterprise versions, see the OpsCenter Compatibility chart.

OpsCenter 5.0.2 Release Notes

Release notes for the OpsCenter version 5.0.2 release.

New features

- Support for DataStax Enterprise 4.6

Resolved issues

- Fixed issue with latency metrics collecting incorrect values. (OPSC-3905)
- Many more minor fixes

Compatibility

To see which versions of OpsCenter are compatible with the various Cassandra and DataStax Enterprise versions, see the OpsCenter Compatibility chart.

OpsCenter 5.0.1 Release Notes

Release notes for the OpsCenter version 5.0.1 release.

New features

- 4 new Best Practice rules:
    - Oracle JRE is being used
    - JNA is properly enabled
    - At least 2 seeds specified per data center
    - Use different interfaces for `listen_address` and `rpc_address` when possible
- Improved dashboard loading
- Added date/time controls to zoomed graphs
- Support for provisioning all recommended EC2 sizes
- Always display data center name on the Ring tab, even if there is only one data center
- Larger Alerts window for more comfortable viewing

Resolved issues

- Fixed issue with UI not recognizing any clusters (displaying the Welcome screen) when the browser does not have an internet connection (OPSC-3285)
- Fixed positioning of nodes on the Ring view in multi-data center and mixed vnode environments (OPSC-3256)
- Fixed memory leaks related to frequent and/or large Data Backup jobs (OPSC-2219)
- Fixed issues with Dashboard preset migrations when upgrading to 5.0.x (OPSC-3373)
- Fixed memory leak in Nodes section of the UI (OPSC-3345)
- Properly display `AVAILABLE` for the Performance Service when using DataStax Enterprise 4.5 or greater. (OPSC-3245)
- Properly retrieve streaming status based on the version of Cassandra in your cluster (OPSC-3335)
- Fixed issue with inability to display metrics for some Solr cores (OPSC-3220)
- Fixed restoring schemas and data with certain non-ASCII characters (OPSC-3184)
- Fixed overwriting `num_tokens` when configuring a node that doesn't have `num_tokens` set (OPSC-3003)
- Many more minor fixes

Compatibility

To see which versions of OpsCenter are compatible with the various Cassandra and DataStax Enterprise versions, see the OpsCenter Compatibility chart.

OpsCenter 5.0.0 Release Notes

Release notes for the OpsCenter version 5.0.0 release.

New features

- Support for 1000 node clusters.
- Granular role-based security.
- New Best Practice service to identify and fix potential problems with a cluster.
- Updated cluster views.
- Redesigned UI.

Resolved issues

- Ensure SSL certificate passwords are not logged on the agent (OPSC-3017)
- Fixed issue where deleting a group in the Performance section required a browser refresh (OPSC-2062)
- Minor bug fixes and improvements

Known issues

- Users with the admin role can make themselves non-admins, so be careful (OPSC-3079)
- Differences between server time and local browser time may cause the **Dashboard** to complain about the end date and time being in the future. To workaround this simply change the end date and time to reflect your computer's. (OPSC-3190)
- If a cluster is in multiple data centers and some but not all of those data centers have vnodes enabled, the data centers with vnodes disabled may display the nodes in incorrect positions. (OPSC-3256)

Compatibility

To see which versions of OpsCenter are compatible with the various Cassandra and DataStax Enterprise versions, see the OpsCenter Compatibility chart.

Tips for using DataStax documentation

Navigating the documents

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

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

Other resources

You can find more information and help at:

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

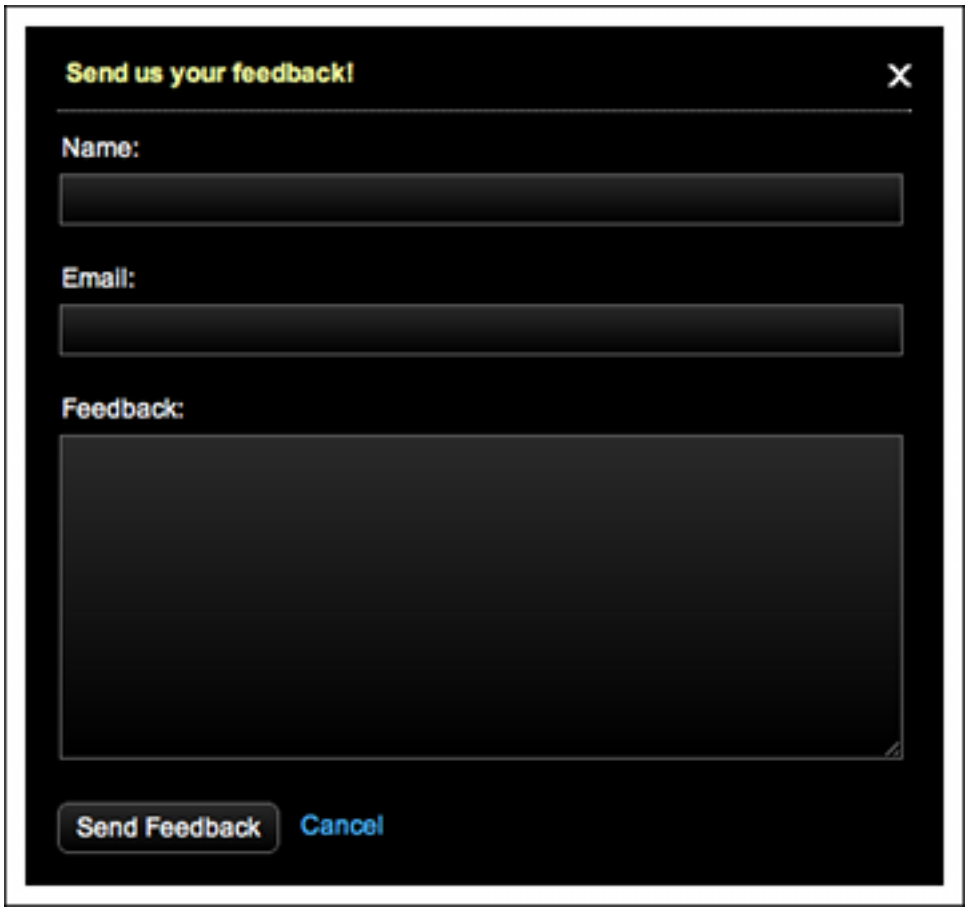
Send us feedback

About this task

Thanks for using OpsCenter. Please take a moment to let us know what you think.

Procedure

1. Click **Help**, then **Feedback** at the top of the console.
   The feedback form appears.



2. Enter your name, email address, and feedback.
3. Click **Send Feedback**.
   We appreciate your comments.