DATASTAX

Cassandra & DataStax Enterprise Essentials Documentation

${ds.localized.time}

Contents

Cassandra & DataStax Enterprise Essentials

This guide shows you how to install and set up a single-node cluster on Cassandra or DataStax Enterprise for evaluation. It also contains an introduction to CQL (Cassandra Query Language), DevCenter (a visual query IDE for creating and running CQL), and basic Cassandra concepts and data modeling considerations.

This document introduces the latest versions of Cassandra, DataStax Enterprise, and OpsCenter. It shows you how to install and set up a **single-node cluster for evaluation.** Additionally, it introduces CQL (Cassandra Query Language) and DevCenter (a visual query IDE for creating and running CQL), basic Cassandra concepts and data modeling considerations.

Note: For information about setting up a production cluster, see the Cassandra 2.1 or DataStax Enterprise documentation. DataStax 4.7 uses Cassandra 2.1.

Products

- DataStax Enterprise delivers Apache Cassandra in a database platform that meets the performance and availability demands of Internet-of-Things (IoT), Web, and Mobile applications. It provides enterprises a secure, fast, always-on database that remains operationally simple when scaled in a single data center or across multiple data centers and clouds.
- Apache Cassandra is a massively scalable open source NoSQL database. Cassandra is perfect for managing large amounts of data across multiple data centers and the cloud. Cassandra delivers continuous availability, linear scalability, and operational simplicity across many servers with no single point of failure, along with a powerful data model designed for maximum flexibility and fast response times.
- DataStax OpsCenter is a visual management and monitoring solution for Apache Cassandra and DataStax Enterprise. It 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.
- DevCenter is a free visual query IDE for creating and running CQL statements against Apache Cassandra and DataStax Enterprise.
- DataStax certified drivers are modern, feature-rich and highly tunable Java client libraries for Apache Cassandra (1.2+) and DataStax Enterprise (3.1+). Use drivers in production applications to pass CQL statements from the

Querying Cassandra with the cqlsh utility

- Quickly master inserting and retrieving data from Cassandra using CQL.

Key database concepts

- A 30-second introduction to key Cassandra terminology.

Cassandra data model

- The data model distilled - a brief introduction to the basic elements of the data model.
- Getting Started with Time Series Data Modeling white paper
- Getting Started with User Profile Data Modeling white paper
- Become a Super Modeler webinar
- The Data Model is Dead, Long Live the Data Model webinar
- C* Summit 2013: The World's Next Top Data Model webinar

Technical and learning resources

- The Best Practices document provides guidelines for designing, deploying, and managing DataStax Enterprise database clusters.
- DataStax Academy provides tutorials and training on Apache Cassandra and DataStax Enterprise.
- The DataStax Developer Blog provides insights and technical articles from top experts and fellow peers on development aspects of DataStax Enterprise and its components, including Apache Cassandra, Apache Hadoop, Apache Solr, and DataStax OpsCenter.
- DataStax Support provides skilled expertise to support your big data environment.

client to a cluster and retrieve, manipulate, or remove data.	

DataStax Enterprise

A brief description of DataStax Enterprise.

DataStax Enterprise delivers Apache Cassandra in a database platform that meets the performance and availability demands of Internet-of-Things (IoT), Web, and Mobile applications. DataStax Enterprise provides enterprises with a secure, fast, always-on database that remains operationally simple when scaled in a single data center or across multiple data centers and clouds.

DataStax Enterprise provides comprehensive management of each type of workload: transactional, analytic, search, and in-memory. With DataStax Enterprise you get:

Certified Cassandra	A production-ready version of Cassandra built for enterprise deployments, which has been certified and optimized by DataStax for maximum performance and uptime.
DSE Analytics	Provides built-in analytics capabilities for running streaming, real-time, and batch analytics operations on Cassandra data. DSE delivers support for multiple analytic-engines including: • Spark - Integration with certified Apache Spark provides fast distributed analytics, including in-memory processing. • DSE Hadoop - Hadoop is integrated with DataStax Enterprise and uses built-in MapReduce, Hive, and Pig. • Bring your own Hadoop (BYOH) - Integration with external Hadoop vendors (Cloudera and HortonWorks) for merging operational information in Cassandra with historical information stored in Hadoop.
DSE Search	Delivers integrated enterprise search capabilities on Cassandra data including: • Enterprise search - Use cases supported include general web, full-text, faceted (categorization), hit prioritization and highlighting, log mining, rich document analysis, geospatial, and social media match ups. • Distributed search - Search large data stores across multiple locations for web and mobile applications. • Constant uptime and availability for search operations. • Live indexing makes new data instantly available for search operations. • Online elasticity - Additional search capacity can easily be added online. • DSE Analytics and Search integration - Analytics jobs can be performed using search queries for finer-grained queries of analytics workloads and better performance. • Easy set up using DataStax OpsCenter.
DSE Advanced Security	Provides advanced security for protecting enterprise data, including: • Kerberos network authentication protocol support. • LDAP and Active Directory support. • Encryption for both in-flight and at-rest data. • Off-server encryption key management using KMIP (Key Management Interoperability Protocol). • Data Auditing using Logback or logging to a Cassandra table.

Visual Administration and Monitoring	DataStax OpsCenter delivers advanced management and monitoring capabilities for DataStax Enterprise clusters.
DSE In-Memory	Supplies in-memory computing capabilities to Cassandra.
DSE Management Services	Automates many key maintenance and performance optimization tasks for a database cluster. Current management services include: • Performance Service - Collects and organizes performance diagnostic information into a set of data dictionary tables that can be queried with CQL (Cassandra Query Language). • Capacity Service - Collects data about a cluster's operations, including Cassandra and platform metrics. • Repair Service - Keeps data synchronized across the cluster. • Backup Service - Schedule, manage, and restore backups from all registered DataStax Enterprise clusters. • Best Practice Service - Periodically scans your clusters to detect issues affecting a cluster's health and reports any deviations with recommendations on how to fix.
DataStax Enterprise Sandbox	DataStax Sandbox is a packaged and personal virtual machine (VM) image that contains everything you need to get started with DataStax Software. DataStax Sandbox provides an easy way to get up to speed quickly on NoSQL, and learn how to use DataStax Enterprise and its visual management and query tools in the shortest time possible.

Installing DataStax Enterprise

Installing a Datastax Enterprise single-node cluster on different platforms.

Installing DataStax Enterprise using the All-In-One Installer

Installing a single-node cluster on any Linux-based platform.

About this task

Use these instructions to install a single-node cluster on any Linux-based platform. You can run the installer as root or under a user account:

- Root or sudo access allows the installer to set up support services on operating systems that support services, such as Debian-based or RHEL-based systems.
- Without root or sudo access, the installer cannot set up support services because it does not have permission to create the services files.

For installing without root or sudo access, see Using the Installer on Mac OS X or without root permissions.

Before you begin

- Oracle Java 7 or OpenJDK 7 **must be** installed. To install, see Installing Oracle JRE on RHEL-based Systems, Installing Oracle JRE on Debian or Ubuntu Systems, or OpenJDK.
- Root or sudo access is recommended.

Procedure

1. Download the installer for your computer from the DataStax download page.

- Linux 64 - `DataStaxEnterprise-4.7.x-linux-x64-installer.run`
- Mac OS X (Non-production installations only.) See the non-root installation instructions.

2. From the directory where you downloaded the install file, change the permission to executable:

```
$ chmod +x DataStaxEnterprise-4.7.x-linux-x64-installer.run
```

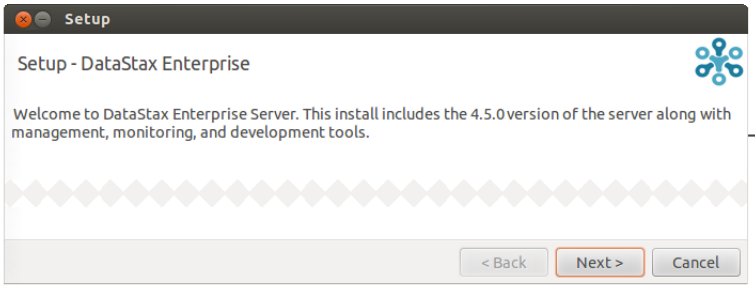
3. To view the installer help:

```
$ ./DataStaxEnterprise-4.7.x-linux-x64-installer.run --help
```

4. Start the installation:

```
$ sudo ./DataStaxEnterprise-4.7.x-linux-x64-installer.run              ##
 Install in GUI mode.
$ sudo ./DataStaxEnterprise-4.7.x-linux-x64-installer.run --mode text  ##
 Install in Text mode.
```

The installer launches.



5. Follow the instructions in the setup wizard using the following table for guidance.

Note: The setup wizard also provides information about install options.

Screen - Panel	Recommendations and additional information
Setup	Welcome page.
License Agreement	DataStax Enterprise End User License Agreement
Install Options	
Server Installation Directory	Use the default location. If you use the No Services option, you can change the location of the `dse` directory. If you install as a service, DataStax Enterprise can only be installed in the `/usr/share/dse` directory.
Install Type	Use **Simple Install**. Installs DataStax Enterprise with the default path names and options.

Screen - Panel	Recommendations and additional information
Update System	Updates some system packages and dependencies. Does not upgrade or install major components such as Oracle Java. Set to **Yes** when run as root user, otherwise set to **No**.
Default Interface	Select **127.0.0.1**. Network interface for the DataStax Enterprise server.
Service Setup	**No Services** - installs the DataStax Enterprise server as a stand-alone process. **Services Only** - installs the DataStax Enterprise server as a service running in the background. **Services and Utilities** (Linux only) - installs the DataStax Enterprise server as a service running in the background and Cassandra utilities, such as cqlsh, sstableloader, sstablescrub, and sstableupgrade to the system path.
Start Services After Install	Select **No**. Generally you select **No** when additional configuration is needed after installation. For this introduction, you'll start the services manually (step 6).
Node Setup	
Node Type	Select **Cassandra Node**. The following types of nodes are available: • Cassandra node Transactional and Bring your own Hadoop (BYOH) nodes. • Search node DSE search (Solr) nodes. • Analytics node Spark Only and Spark + Integrated Hadoop (DSE Hadoop) nodes.
Ring Name	Use the default name or enter a name of your choice.
Seeds	Select **127.0.0.1**. Cassandra nodes use the seed node list for finding each other and learning the topology of the ring.
Setup	
DataStax Agent	Select **OpsCenter Address: 127.0.0.1**. The agent provides an interface between DataStax OpsCenter and DataStax Enterprise.
System Configuration	Configuration overview and warnings about potential issues.
Ready to Install	The install wizard installs the software.
Setup finish	Post-installation tasks. To see the Pre-flight check results, select View Configuration Recommendations And Warnings

6. Start the services:

```
$ sudo service dse start ## Starts the DataStax Enterprise server
$ sudo service datastax-agent start ## Starts the DataStax Agent
```

7. Verify that DataStax Enterprise is running:

```
$ nodetool status
```

```
paul@ubuntu:~/dse$ bin/nodetool status
Note: Ownership information does not include topology; for complete information, specify a keyspace
Datacenter: Cassandra
=====================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address    Load       Tokens  Owns   Host ID                               Rack
UN  127.0.0.1  59.46 KB   256     100.0%  beb02eb3-4291-4559-bea4-fbcdc31cfa8e  rack1
```

8. Install OpsCenter using the instructions for your platform.

What to do next

- Key concepts.
- The data model distilled.
- Run the DataStax Enterprise demos.
- Learn basic Cassandra Query Language (CQL) commands using the cqlsh utility.
- Take the DevCenter tutorial.
- Uninstall DataStax Enterprise.

Installing DataStax Enterprise on Linux without root permissions or Mac OS X

Installing a single-node cluster without root permissions on any Linux platform or Mac OS X.

About this task

Installing a single-node cluster without root permissions on any Linux platform or Mac OS X. You can run the installer as root or under a user account:

- Root or sudo access allows the installer to set up support services on operating systems that support services, such as Debian-based or RHEL-based systems.
- Without root or sudo access, the installer cannot set up support services because it does not have permission to create the services files.

For root installations, see Using the All-In-One Installer.

Before you begin

Oracle Java 7 **must be** installed. To install, see Installing Oracle JRE on RHEL-based Systems or Installing Oracle JRE on Debian or Ubuntu Systems.

Procedure

1. Download the installer for your computer from the DataStax download page.
   - Linux 64 - `DataStaxEnterprise-4.7.x-linux-x64-installer.run`
   - Mac OS X - `DataStaxEnterprise-4.7.x.dmg` (Non-production installations only.)
2. From the directory where you downloaded the install file:
   - **Linux:**
     1. From the directory where you downloaded the install file, change the permission to executable:

```
$ chmod +x DataStaxEnterprise-4.7.x-linux-x64-installer.run ## Changes
 permission to executable
```

2. To view the installer help:

```
$ ./DataStaxEnterprise-4.7.x-linux-x64-installer.run --help
```

3. Start the installation:

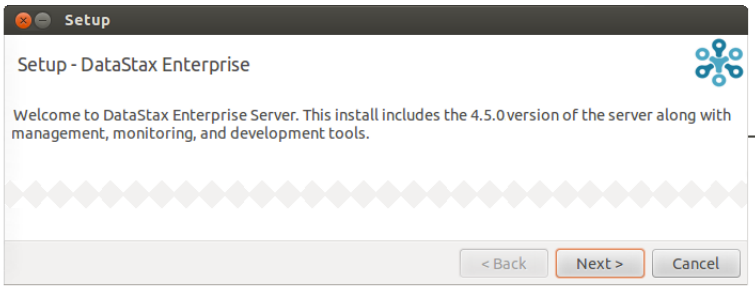
```
$ ./DataStaxEnterprise-4.7.x-linux-x64-installer.run              ##
 Install in GUI mode.
$ ./DataStaxEnterprise-4.7.x-linux-x64-installer.run --mode text  ##
 Install in Text mode.
```

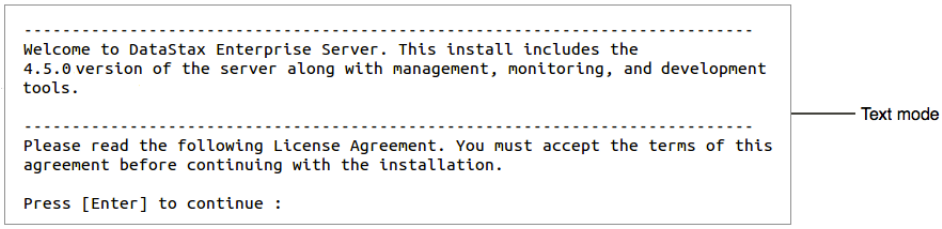
- **Mac OS X:**

1. Double-click the `DataStaxEnterprise-4.7.x.dmg` file.
2. In the installer window, double-click the installer icon.

   Depending on your permissions, you may need to use **Control-click**.

The installer launches.





3. Follow the instructions in the setup wizard using the following table for guidance.

Screen - Panel	Recommendations and additional information
Setup	Welcome page.
License Agreement	DataStax Enterprise End User License Agreement
Install Options	
Server Installation Directory	By default, DataStax Enterprise is installed in your home directory. If you use the No Services option, you can change the location of the `dse` directory. If you install as a service, DataStax Enterprise can only be installed in the `/usr/share/dse` directory.
Install Type	Use **Simple Install**. Installs DataStax Enterprise with the default path names and options.

Screen - Panel	Recommendations and additional information
Update System	Updates some system packages and dependencies. Does not upgrade or install major components such as Oracle Java. Set to **No** unless you have root or sudo access.
Default Interface	Select **127.0.0.1**. Network interface for the DataStax Enterprise server.
Service Setup	Select **No Services** - installs the DataStax Enterprise server as a stand-alone process.
Start Services After Install	Select **No**. A non-root install doesn't use services.
Node Setup	
Node Type	Select **Cassandra Node**. The following types of nodes are available: • Cassandra node Transactional and Bring your own Hadoop (BYOH) nodes. • Search node DSE search (Solr) nodes. • Analytics node Spark Only and Spark + Integrated Hadoop (DSE Hadoop) nodes.
Ring Name	Use the default name or enter a name of your choice.
Seeds	Select **127.0.0.1**. Cassandra nodes use the seed node list for finding each other and learning the topology of the ring.
Setup	
DataStax Agent	Select **OpsCenter Address: 127.0.0.1**. The agent provides an interface between DataStax OpsCenter and DataStax Enterprise.
System Configuration	Configuration overview and warnings about potential issues.
Ready to Install	The install wizard installs the software.
Setup finish	Post-installation tasks. To see the Pre-flight check results, select View Configuration Recommendations And Warnings

4. Go to the install directory:

```
$ cd /home/user/dse
```

5. From the install directory, start DataStax Enterprise and the DataStax Agent:

```
$ bin/dse cassandra ## Starts DataStax enterprise as a transactional
 (Cassandra) node
```

```
$ ./datastax-agent/bin/datastax-agent ## An interface between OpsCenter and
 DataStax Enterprise
```

6. Verify that DataStax Enterprise is running:

```
$ bin/nodetool status
```

```
paul@ubuntu:~/dse$ bin/nodetool status
Note: Ownership information does not include topology; for complete information, specify a keyspace
Datacenter: Cassandra
=====================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address    Load       Tokens  Owns   Host ID                               Rack
UN  127.0.0.1  59.46 KB   256     100.0%  beb02eb3-4291-4559-bea4-fbcdc31cfa8e  rack1
```

7. Install OpsCenter using the instructions for your platform.

What to do next

- Key concepts.
- The data model distilled.
- Run the DataStax Enterprise demos.
- Learn basic Cassandra Query Language (CQL) commands using the cqlsh utility.
- Take the DevCenter tutorial.
- Uninstall DataStax Enterprise.

Other install methods

Install using yum, APT, or a binary tarball.

Install using yum, APT, or a binary tarball.

The following install methods are described in the DataStax Enterprise documentation:

- Installing DataStax Enterprise using Yum repositories
- Installing DataStax Enterprise using APT repositories
- Installing DataStax Enterprise using the binary tarball

Uninstalling DataStax Enterprise

Remove DataStax Enterprise using the DataStax Installer.

The DataStax Installer provides the ability to uninstall parts or all of DataStax Enterprise. For more information, see Uninstalling DataStax Enterprise.

DataStax Demos

Listing of the available demos.

- Portfolio Manager demo using Spark or Portfolio Manager demo using DSE Hadoop - An application showing a sample mixed workload on a DataStax Enterprise cluster. The use case is a financial application where users can actively create and manage a portfolio of stocks.
- Solr Wikipedia demo - This demo application uses Wikipedia as an example of Solr capabilities.
- Weather Sensor demo - This demo allows you to run analytical queries with Hadoop and Spark against dynamically generated data for a number of weather sensors in different cities. It also includes a web interface for creating custom queries against the data.
- Hive example: Use a CQL composite partition key - This example first creates a CQL table, and then creates an external table in Hive that maps to the CQL table.

- Hive example: Work with an unsupported data type - DataStax Enterprise provides a user defined function (UDF) for converting Hive binary data into string representations of CQL types. In this example, you create a custom external table in Hive and map these types to binary.
- Pig demo - Uses sample data to illustrate using Pig in DataStax Enterprise. The examples in the topic show how to create a Pig relation and perform a simple MapReduce job.
- Mahout demo - Using Mahout algorithms, this demo classifies historical time series data into categories based on whether the data has exhibited relatively stable behavior over a period of time.
- Sqoop demo - Shows an example of migrating the data from a MySQL table to text files in CFS. This demo requires Oracle JDK 1.6.x; the JRE alone will not work.

Moving data to DataStaxEnterprise from other databases

DataStax Enterprise migration solutions.

DataStax offers several solutions for migrating from other databases. For more information, see Moving data to or from DataStax Enterprise.

Cassandra

A brief description of Cassandra, how it works, and links to technical and learning resources.

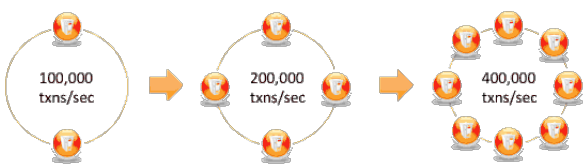
This topic contains basic information about Cassandra and how it works, plus links to the 10 Minute Cassandra Walkthrough and resources for understanding Cassandra's data model.

What is Apache Cassandra?

Apache Cassandra™ is a massively scalable open source NoSQL database. Cassandra is perfect for managing large amounts of structured, semi-structured, and unstructured data across multiple data centers and the cloud. Cassandra delivers continuous availability, linear scalability, and operational simplicity across many commodity servers with no single point of failure, along with a powerful dynamic data model designed for maximum flexibility and fast response times.

How does Cassandra work?

Cassandra's built-for-scale architecture means that it is capable of handling petabytes of information and thousands of concurrent users/operations per second.

Cassandra is a partitioned row store database	Cassandra's architecture allows any authorized user to connect to any node in any data center and access data using the CQL language. For ease of use, CQL uses a similar syntax to SQL. The most basic way to interact with Cassandra is using the CQL shell, cqlsh. Using cqlsh, you can create keyspaces and tables, insert and query tables, plus much more. If you prefer a graphical tool, you can use DataStax DevCenter. For production, DataStax supplies a number drivers so that CQL statements can be passed from client to cluster and back. Other administrative tasks can be accomplished using OpsCenter.
Automatic data distribution	Cassandra provides automatic data distribution across all nodes that participate in a *ring* or database cluster. There is nothing programmatic that a developer or administrator needs to do or code to distribute data across a cluster because data is transparently partitioned across all nodes in a cluster.
Built-in and customizable replication	Cassandra also provides built-in and customizable replication, which stores redundant copies of data across nodes that participate in a Cassandra ring. This means that if any node in a cluster goes down, one or more copies of that node's data is available on other machines in the cluster. Replication can be configured to work across one data center, many data centers, and multiple cloud availability zones.
Cassandra supplies linear scalability	Cassandra supplies linear scalability, meaning that capacity may be easily added simply by adding new nodes online. For example, if 2 nodes can handle 100,000 transactions per second, 4 nodes will support 200,000 transactions/sec and 8 nodes will tackle 400,000 transactions/sec:
	

10 Minute Cassandra Walkthrough

Planet Cassandra provides a 10 Minute Cassandra Walkthrough where you can download a Cassandra virtual machine (VMware or VirtualBox). Plus you can take short video courses for developers and administrators that demonstrate various Cassandra's features.

Cassandra data model

The following resources are helpful for understanding data modeling:

- The data model distilled - a brief introduction to the basic elements of the data model.
- Getting Started with Time Series Data Modeling white paper
- Getting Started with User Profile Data Modeling white paper
- Become a Super Modeler webinar
- The Data Model is Dead, Long Live the Data Model webinar
- C* Summit 2013: The World's Next Top Data Model webinar

Installing Cassandra

Installing Cassandra on different platforms.

Installing DataStax Community on RHEL, CentOS, or Oracle Linux

Install using a yum repository.

About this task

If you have trouble installing, see the full installation documentation. For information about setting up a production cluster, see the Cassandra documentation.

Before you begin

- Oracle Java 7 **must be** installed. To install, see Installing Oracle JRE on RHEL-based Systems.
- Root or sudo access.
- Python 2.6+ (needed if installing OpsCenter).
- 256MB of memory (only for testing light workloads). If using a virtual machine, be sure to use the recommended memory allocation or more for your operating system.

Procedure

In a terminal window:

1. Add a DataStax Community repository file called `/etc/yum.repos.d/datastax.repo`.

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

2. Install the packages:

   ```
   $ sudo yum install dsc21
   ```

3. Start DataStax Community (as a single-node cluster):

   ```
   $ sudo service cassandra start
   ```

   On some Linux distributions, you many need to use:

```
$ sudo /etc/init.d/cassandra start
```

4. Verify that DataStax Community is running:

```
$ nodetool status
```

```
paul@ubuntu:~/cassandra-1.2.0$ bin/nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address          Load       Tokens  Owns   Host ID                               Rack
UN  10.194.171.160   53.98 KB   256     100%   a9fa31c7-f3c0-44d1-b8e7-a2628867840c  rack1
```

What to do next

- Install and set up OpsCenter (Optional).
- Key concepts.
- The data model distilled.
- Learn basic Cassandra Query Language (CQL) commands using the cqlsh utility.
- Take the DevCenter tutorial.
- Set up a single or multiple data center cluster.

Installing DataStax Community on Debian or Ubuntu

Install using an APT repository.

About this task

If you have trouble installing, see the full installation documentation. For information about setting up a production cluster, see the Cassandra documentation.

Before you begin

- Oracle Java 7 **must be** installed. To install, see Installing Oracle JRE on Debian or Ubuntu Systems.
- Root or sudo access.
- Python 2.6+ (needed if installing OpsCenter).
- 256MB of memory (only for testing light workloads). If using a virtual machine, be sure to use the recommended memory allocation or more for your operating system.

Procedure

In a terminal window:

1. Add the DataStax Community repository to the `/etc/apt/sources.list.d/cassandra.sources.list`

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

```
$ sudo apt-get update
$ sudo apt-get install dsc21
```

This installs the DataStax Community distribution of Cassandra. The Debian packages start the Cassandra service automatically.

4. Verify that DataStax Community is running:

```
$ nodetool status
```

```
paul@ubuntu:~/cassandra-1.2.0$ bin/nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address          Load       Tokens  Owns   Host ID                               Rack
UN  10.194.171.160   53.98 KB   256     100%   a9fa31c7-f3c0-44d1-b8e7-a2628867840c  rack1
```

What to do next

- Install and set up OpsCenter (Optional)
- Set up a single or multiple data center cluster.
- Key concepts.
- Learn basic Cassandra Query Language (CQL) commands using the cqlsh utility.
- Take the DevCenter tutorial.
- The data model distilled.

Installing DataStax Community on any Linux platform or Mac OS X from the tarball

Use this method to install a single-node cluster on Mac OS X and platforms without package support, or if you do not have or want a root installation.

About this task

If you have trouble installing, see the full installation documentation. For information about setting up a production cluster, see the Cassandra documentation.

Before you begin

- Oracle Java 7 **must be** installed. To install, see Installing Oracle JRE on RHEL-based Systems or Installing Oracle JRE on Debian or Ubuntu Systems.
- Python 2.6+ (needed if installing OpsCenter).
- 256MB of memory (only for testing light workloads). If using a virtual machine, be sure to use the recommended memory allocation or more for your operating system.

Procedure

In a terminal window:

1. Download and untar the DataStax Community tarball:

   ```
   $ curl -L http://downloads.datastax.com/community/dsc.tar.gz | tar xz
   ```

   You can also download from Planet Cassandra.
2. Go to the install directory:

   ```
   $ cd dsc-cassandra-2.1.x
   ```
3. For instructions about installing without root permissions, click here.
4. Start DataStax Community from the install directory:

   ```
   $ sudo bin/cassandra
   ```
5. Verify that DataStax Community is running. From the install directory:

   ```
   $ bin/nodetool status
   ```

```
paul@ubuntu:~/cassandra-1.2.0$ bin/nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address          Load       Tokens  Owns   Host ID                               Rack
UN  10.194.171.160   53.98 KB   256     100%   a9fa31c7-f3c0-44d1-b8e7-a2628867840c  rack1
```

What to do next

- Install and set up OpsCenter (Optional)
- Key concepts.
- The data model distilled.
- Learn basic Cassandra Query Language (CQL) commands using the cqlsh utility.
- Take the DevCenter tutorial.
- Set up a single or multiple data center cluster.

Installing without root permissions

Installing Cassandra when you don't have or want to use sudo or root permissions.

About this task

Before performing this steps, you must have completed steps 1 and 2 in Any Linux system or Mac OS X.

Procedure

1. In the install directory, create the data and log directories:

   ```
   $ mkdir cassandra-data; cd cassandra-data
   $ mkdir data saved_caches commitlog
   ```

2. Edit the `cassandra.yaml` file:

   a) `cd` *`path_to_install`*`/conf/`

   b) Edit these settings:

   ```
   initial_token: 0
   data_file_directories: - path_to_install/cassandra-data/data
   commitlog_directory: path_to_install/cassandra-data/commitlog
   saved_caches_directory: path_to_install/cassandra-data/saved_caches
   ```

3. In the `conf` directory the `log4j-server.properties` file to include the settings:

   ```
   log4j.appender.R.File= path_to_install/cassandra-data/system.log
   ```

4. Start DataStax Community from the install directory:

   ```
   $ bin/cassandra
   ```

5. Verify that DataStax Community is running. From the install directory:

   ```
   $ bin/nodetool status
   ```

```
paul@ubuntu:~/cassandra-1.2.0$ bin/nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address          Load       Tokens  Owns   Host ID                               Rack
UN  10.194.171.160   53.98 KB   256     100%   a9fa31c7-f3c0-44d1-b8e7-a2628867840c  rack1
```

What to do next

- Install and set up OpsCenter (Optional)
- Key concepts.

- The data model distilled.
- Learn basic Cassandra Query Language (CQL) commands using the cqlsh utility.
- Take the DevCenter tutorial.
- Set up a single or multiple data center cluster.

Installing DataStax Community on Windows

Install Cassandra and OpsCenter web application on 32- or 64-bit Windows 7 or Windows Server 2008.

Before you begin

- 32-bit systems require the Visual C++ 2008 runtime (Microsoft Visual C++ 2008 Redistributable Package (x86)).
- Chrome or Firefox (for OpsCenter)
- OpenSSL: 0.9.8. (SSL is disabled by default; see Enabling SSL – Windows.)

Note: Windows 8 has not been tested.

Procedure

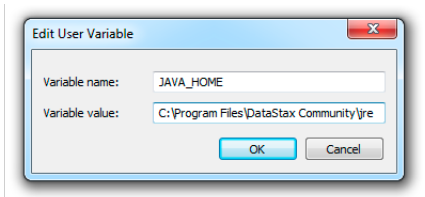
1. Download the Windows installer for your platform from Planet Cassandra.

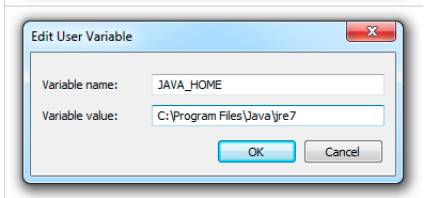
   Choose either DataStax Community 2.1.x. or 2.0.x.
2. Follow the DataStax Community setup wizard to install. During installation, accept the options to automatically start the services. This option starts the DataStax server, OpsCenter, and OpsCenter agent automatically when the installation completes (and whenever the computer reboots).
3. Accept the option to launch immediately.

OpsCenter is displayed.

4. If you have trouble:

   - Check that the *JAVA_HOME* environmental variable points to either of the following (Cassandra requires Java 7):





   - If nodetool does not start:

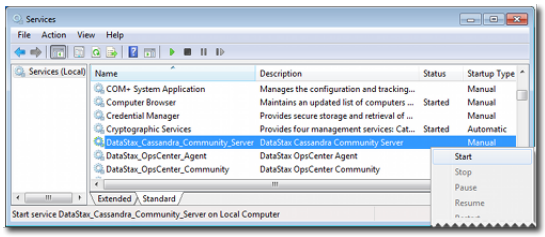
     ```
     'nodetool' is not recognized as an internal or external command,
     ```

```
operable program or batch file
```

Run nodetool commands from the installation directory:

```
$ cd \Program Files\DataStax Community\apache-cassandra\bin
$ nodetool status
```

- OpsCenter does not start up:
  1. Check that the Cassandra and OpsCenter services are running. If not, start them in the Control Panel.



  2. Check that the Microsoft Visual C++ 2008 Redistributable Package is installed. If not, download and install the package from Microsoft.
- The OpsCenter Agent service takes a long time to stop, be patient. Manually stopping the OpsCenter Agent service takes time.

5. To uninstall DataStax Community, use either:
   - The **Uninstall DataStax Community Edition** on the **Start menu**.
   - The Windows **Uninstall a program** feature on the **Control Panel**.

What to do next

- Key concepts.
- The data model distilled.
- Learn basic Cassandra Query Language (CQL) commands using the cqlsh utility.
- Take the DevCenter tutorial.
- Set up a single or multiple data center cluster.

Moving data to Cassandra from other databases

Cassandra migration solutions.

Cassandra offers several solutions for migrating from other databases. For more information, see Moving data to or from Cassandra.

OpsCenter

A brief description about OpsCenter.

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.

Installing OpsCenter

Instructions on installing OpsCenter to manage Cassandra and DataStax Enterprise clusters.

Installing OpsCenter with the standalone installer

Installs the 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 during Installation of DataStax Enterprise or Cassandra.

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.
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.version number-linux-64-installer.run
        ```
     2. Start the installation:

        ```
        ./DataStaxOpsCenter-5.1.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/
```

Installing OpsCenter on RHEL-based systems

Installing OpsCenter web application on a single node cluster using yum.

About this task

Attention: If you have already installed OpsCenter using the RHEL install of DataStax Enterprise, see Setting up OpsCenter.

Before you begin

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

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
```

4. Start OpsCenter:

```
$ sudo service opscenterd start
```

5. Connect to OpsCenter.

Installing OpsCenter on Debian or Ubuntu

Installing OpsCenter web application on a single node cluster using APT.

About this task

Attention: If you have already installed OpsCenter using the Debian install of DataStax Enterprise, see Setting up OpsCenter.

Before you begin

- APT Package Manager is installed
- Python 2.6+

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:

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

4. Start OpsCenter:

   ```
   $ sudo service opscenterd start
   ```

5. Connect to OpsCenter.

Installing OpsCenter on any Linux platform or Mac OS X

Installing OpsCenter web application on a single-node cluster.

About this task

Follow the instructions below when you have installed DataStax Community, **not** DataStax Enterprise; use these instructions instead.

Note: Although you can install and run OpsCenter from Mac OS X, provisioning a node is not supported. You can install agents on other Linux platforms. See the OpsCenter User Guide.

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
   ```

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

2. Go to the agent directory:

   ```
   $ cd opscenter_install_location/agent
   ```

3. Start the agent:

   ```
   $ bin/datastax-agent
   ```

   Use -f to start in the foreground.

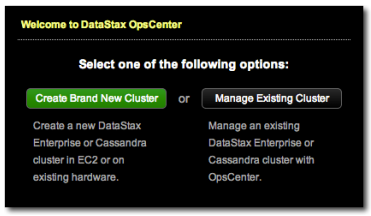
4. Go to the install directory and start OpsCenter:

   ```
   $ cd ..
   $ bin/opscenter
   ```

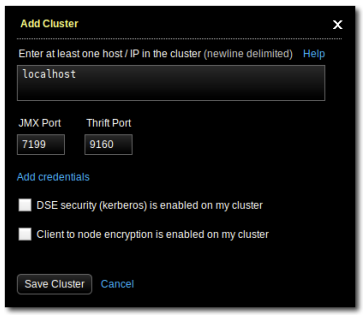
   Use -f to start in the foreground.

5. Open a browser window and go to the OpsCenter URL at `http://localhost:8888`.

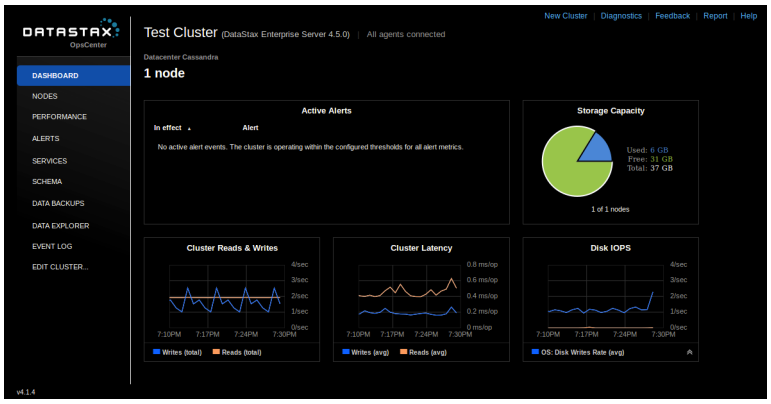
   When you start OpsCenter for the first time, you will be prompted to connect to your cluster:



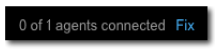
6. In **Welcome to DataStax Opscenter**, click **Manage Existing Cluster**.
7. In **Add Cluster**, enter `localhost` into the **Host/IP** field, and then click **Save Cluster**.



OpsCenter connects to the cluster.



8. If a **Fix** link appears, see Troubleshooting agent installs.



Installing DataStax Community on Windows

Install Cassandra and OpsCenter web application on 32- or 64-bit Windows 7 or Windows Server 2008.

Before you begin

- 32-bit systems require the Visual C++ 2008 runtime (Microsoft Visual C++ 2008 Redistributable Package (x86)).
- Chrome or Firefox (for OpsCenter)

- OpenSSL: 0.9.8. (SSL is disabled by default; see Enabling SSL – Windows.)

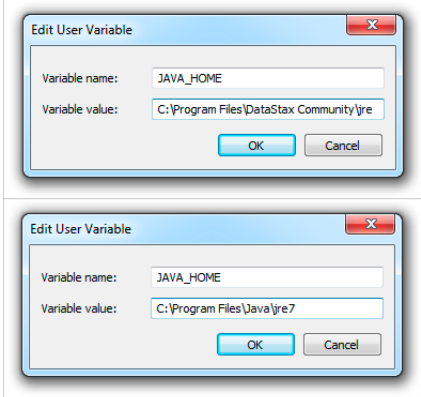
Note: Windows 8 has not been tested.

Procedure

1. Download the Windows installer for your platform from Planet Cassandra.

   Choose either DataStax Community 2.1.x. or 2.0.x.
2. Follow the DataStax Community setup wizard to install. During installation, accept the options to automatically start the services. This option starts the DataStax server, OpsCenter, and OpsCenter agent automatically when the installation completes (and whenever the computer reboots).
3. Accept the option to launch immediately.

   OpsCenter is displayed.
4. If you have trouble:
   - Check that the *JAVA_HOME* environmental variable points to either of the following (Cassandra requires Java 7):



- If nodetool does not start:

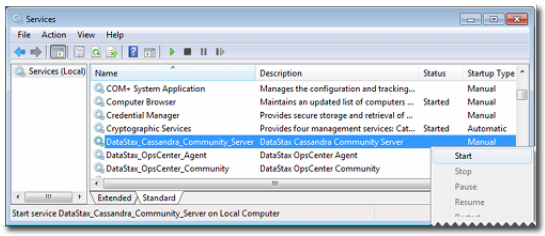
  ```
  'nodetool' is not recognized as an internal or external command,
  operable program or batch file
  ```

  Run nodetool commands from the installation directory:

  ```
  $ cd \Program Files\DataStax Community\apache-cassandra\bin
  $ nodetool status
  ```

- OpsCenter does not start up:

  1. Check that the Cassandra and OpsCenter services are running. If not, start them in the Control Panel.

  2. Check that the Microsoft Visual C++ 2008 Redistributable Package is installed. If not, download and install the package from Microsoft.

- The OpsCenter Agent service takes a long time to stop, be patient. Manually stopping the OpsCenter Agent service takes time.

5. To uninstall DataStax Community, use either:

   - The **Uninstall DataStax Community Edition** on the **Start menu**.
   - The Windows **Uninstall a program** feature on the **Control Panel**.

What to do next

- Key concepts.
- The data model distilled.
- Learn basic Cassandra Query Language (CQL) commands using the cqlsh utility.
- Take the DevCenter tutorial.
- Set up a single or multiple data center cluster.

Setting up OpsCenter

How to configure the OpsCenter web application after it has already been installed.

Before you begin

- Your cluster is properly configured and running.

- Python 2.6+

Procedure

1. Start OpsCenter:

```
$ sudo service opscenterd start
```

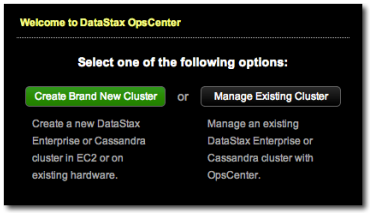
2. Connect to OpsCenter in a web browser at `http://localhost:8888`.

   Note:

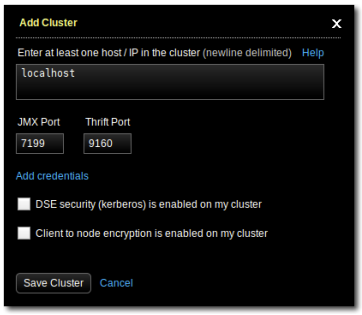
   If installing on a headless system, you must configure the [webserver] interface to use its IP address instead of localhost. It is located in the `/etc/opscenter/opscenterd.conf` file. For example:

```
[webserver]
port = 8888
interface = 10.183.170.161
```

   When you start OpsCenter for the first time, you will be prompted to connect to your cluster:



3. In **Welcome to DataStax OpsCenter**, click **Use Existing Cluster**.
4. In **Add Cluster**, enter `localhost` into the **Cluster Hosts/IPs** field, and then click **Save Cluster**.



   The Dashboard appears and Opscenter looks for the agent.

   After the agent is connected, OpsCenter is ready for use.

   Note: If OpsCenter cannot find the agent, follow these instructions.

Results

It takes a few minutes for OpsCenter to complete the agent installation. After all agents are installed, a success message is displayed.

Setting up OpsCenter after installing DSE from the tarball

Configure OpsCenter web application after it has already been installed from a DataStax Enterprise tarball.

About this task

If you did not download and untar the OpsCenter, run this command in a terminal window:

```
$ curl -L http://downloads.datastax.com/community/opscenter.tar.gz | tar xz
```

Note: Although you can install and run OpsCenter from Mac OS X, provisioning a node is not supported. You can install agents on other Linux platforms. See the OpsCenter User Guide.

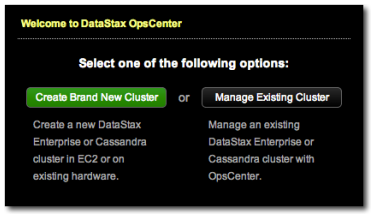
Procedure

1. Go to the OpsCenter install directory.
2. Start OpsCenter:

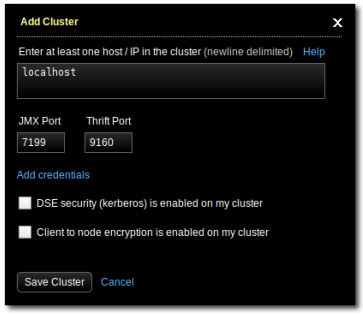
   ```
   $ bin/opscenter
   ```

   Use -f to start in the foreground.
3. Open a browser window and go to the OpsCenter URL at `http://localhost:8888`.

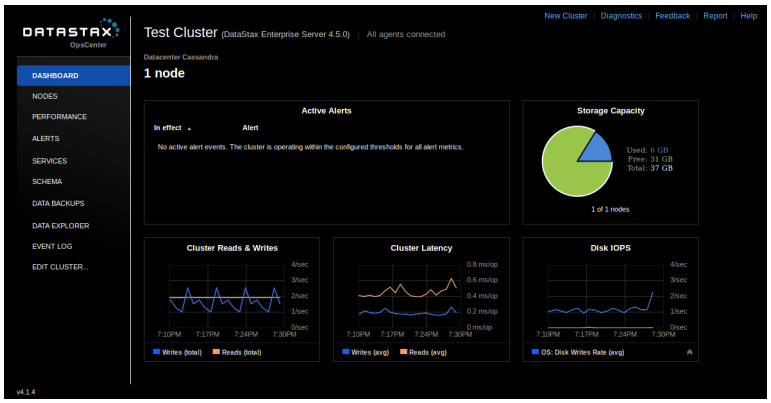
   When you start OpsCenter for the first time, you will be prompted to connect to your cluster:



4. In **Welcome to DataStax Opscenter**, click **Manage Existing Cluster**.
5. In **Add Cluster**, enter `localhost` into the **Host/IP** field, and then click **Save Cluster**.



   OpsCenter connects to the cluster.



6. If a **Fix** link appears, see Troubleshooting agent installs.

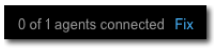
Troubleshooting agent installs

Instructions when OpsCenter cannot find the agents.

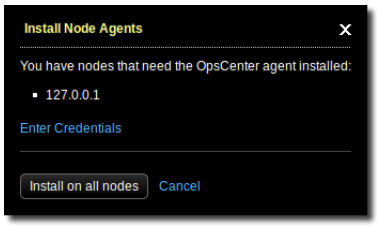
About this task

Follow these instructions **only** if OpsCenter cannot find the agents.

1. If OpsCenter cannot find the agent, a **Fix** link appears:

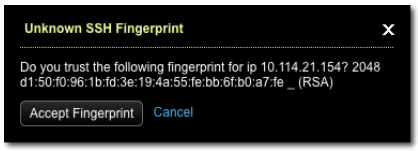


2. Click **Fix** link.
3. In **Install Node Agents**, click **Enter Credentials**.



4. In **SSH Credentials**, enter the **Username** and **Password** for the machine where you are installing the agent, and then click **Done**.
5. In the **Install Nodes Agent** dialog, click **Install on all nodes**.

   Note: If the credentials you entered are correct and OpsCenter still cannot find the agent, go to Restarting the DataStax agent and OpsCenter.
6. If prompted, click **Accept Fingerprint** to add a node to the known hosts for OpsCenter.



Restarting the DataStax agent and OpsCenter

If the DataStax agent still doesn't connect, restart the agent and then restart the OpsCenter. The procedure depends on the type of install:

Packaged installs:

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

Tarball installs:

1. Stop the DataStax agent by finding its Java process ID (PID) and killing the process using its PID number:

   ```
   $ ps -ef | grep datastax-agent
   $ kill pid
   ```
2. Start the DataStax agent:
   - If using DataStax Community, start the DataStax agent from the `agent` directory:

     ```
     $ agent_install_location/bin/datastax-agent
     ```

     Use -f to start in the foreground.
   - If using DataStax Enterprise, start the DataStax agent from the `dse-4.7.x` directory:

     ```
     $ ./datastax-agent/bin/datastax-agent
     ```

     Use -f to start in the foreground.
3. Stop the OpsCenter by finding its PID and killing the process:

   ```
   $ ps -ef | grep opscenter
   $ kill pid
   ```
4. Start the OpsCenter:

   ```
   $ opscenter_install_location/bin/opscenter
   ```
5. In your browser, reload the OpsCenter page.

   The OpsCenter should now connect to the agent.

Creating a cluster using OpsCenter

How to create a single-node cluster using the OpsCenter UI.

Before you begin

Before you can install the cluster, you must meet the following prerequisites:

- The target machine must be running a supported version of Linux.
- The target machine must have Oracle Java 7 installed (instructions for RHEL-based systems and Debian or Ubuntu systems).
- The target machine must have a user capable of using `sudo`.
- If you are installing DataStax Enterprise, you must have your DataStax login credentials.

About this task

Procedure

1. Click **New Cluster**.
2. Click **Create Brand New Cluster**. The **Create Cluster** dialog will open.
3. Fill out the form with your settings.
   a) Under **Name** enter the name for this cluster.
   b) Under **Package** select the version of DataStax Enterprise or DataStax Community you want to install.
   c) If you are installing a DataStax Enterprise cluster, enter the username and password emailed to you when you registered to download DataStax Enterprise.
   d) Under **Nodes** enter the hostname or IP address of the target machine.
   e) Under **Node Credentials (sudo)** enter the username and password of a user account capable of running `sudo` on the target machine.
4. Click **Build Cluster**. OpsCenter will install, configure, and start the cluster on the target machine.

Key concepts and data model

Basic concepts for understanding Cassandra and DataStax Enterprise and an introduction to data modeling in Cassandra.

Key concepts

The following concepts are important for understanding Cassandra:

Cluster

A group of nodes where you store your data. In this guide, you create a single-node cluster.

Replication

The process of storing copies of data on multiple nodes to ensure reliability and fault tolerance. The number of copies is set by the replication factor.

Partitioner

A partitioner distributes data evenly across the nodes in the cluster for load balancing.

Data Center

A group of related nodes configured together within a cluster for replication purposes. It is not necessarily a physical data center. In DataStax Enterprise, the term *related nodes* refers to the type of node: transactional, analytics, search. Each type of node must be contained in its own data center.

Links to related information in Cassandra

- Cassandra key features
- Cassandra Architecture

The data model distilled

Cassandra is a partitioned row store. It is an open-source, distributed-database system that is designed for storing and managing large amounts of data across commodity servers.

You design the data model

The design of the data model is based on the queries you want to perform, not on modelling entities and relationships like you do for relational databases.

Column family

In general, the term **Table** has replaced Column family. A Cassandra database consists of column families. A column family is a set of key-value pairs. Every column family has a key and consists of columns and rows. You can think of column family as a table and a key-value pair as a record in a table.

Note: In CQL 3 (the latest implementation of the Cassandra Query Language), *column families* are called *tables*. The Cassandra CLI client utility (deprecated) , API classes, and OpsCenter continue to use column family.

Table

The definition of a table depends on the version of CQL:

- In CQL 3, a table is a collection of ordered (by name) columns.
- In previous versions of CQL, a column family was synonymous, in many respects, to a table. In CQL 3 a table is sparse, which means it only includes columns for rows that have been assigned a value.

Keyspaces

The outermost grouping of data, similar to a schema in a relational database. All tables go inside a keyspace. Typically, a cluster has one keyspace per application.

Links to related information about data modeling

- Cassandra key features in the Cassandra documentation
- Cassandra Architecture in the Cassandra documentation
- Getting Started with Time Series Data Modeling white paper
- Getting Started with User Profile Data Modeling white paper
- Become a Super Modeler webinar
- The Data Model is Dead, Long Live the Data Model webinar
- C* Summit 2013: The World's Next Top Data Model webinar

Querying Cassandra

Using CQL or DevCenter to inserting and retrieving data.

Using DevCenter

A short tutorial on using DataStax DevCenter to create and maintain Cassandra databases.

About this task

This tutorial introduces DevCenter. Using sample queries files, it shows you how to use CQL to extract data from a Cassandra datatbase.

Before you begin

- Download and install DataStax DevCenter.
- A Cassandra or DataStax Enterprise cluster installed and running

About this task

This tutorial uses a simple CQL schema included with DevCenter and provides some data.

Procedure

1. Select **Open CQL Script** in the CQL Scripts pane to open the sample VideoDb schema CQL file.

   devcenter-home-directory/examples/videodb-schema.cql
2. Select the **Execute CQL Script** button.

   A new node, videodb, appears in the **Schema Navigator** pane.
3. Open the sample VideoDB data file from the **CQL Scripts** pane to insert some data in the videodb schema.

   devcenter-home-directory/examples/videodb-inserts.cql
4. Execute the CQL statements.
5. Open the sample VideoDB queries file from the **CQL Scripts** pane.

   The file contains some sample queries to execute against the videodb schema. Copying and pasting them into a blank CQL script and then executing them illustrates how to extract data from a Cassandra datatbase.

   devcenter-home-directory/examples/videodb-sample-queries.cql
6. Open the VideoDB user-defined type (UDT) and tuples file from the **CQL Scripts** pane.

   This file demonstrates usage of UDTs and tuples which are features that require Cassandra 2.1 or higher.

   devcenter-home-directory/examples/videodb-udts-tuples.cql

Using cqlsh

Quickly master inserting and retrieving data from Cassandra 2.1 using the cqlsh utility.

Attention: The information presented here applies only to Cassandra 2.x not to Cassandra 1.2.

About this task

You can run Cassandra Query Language (CQL) using the cqlsh utility to:

1. Create a keyspace, which is akin to the namespace of an SQL database.
2. Use the keyspace to create a table, which is similar to an SQL table.
3. Insert data into the table.
4. Use queries to sort, retrieve, alter, automatically expire, and drop the data.

Procedure

From a terminal:

1. Assuming Cassandra is running, start cqlsh on Windows or Linux from the installation directory. In a shell on Mac OS X, for example:

   ```
   $ ./bin/cqlsh
   ```
2. At the cqlsh prompt, use the DESCRIBE cqlsh command to see the keyspaces that already exist in Cassandra:

   ```
   DESCRIBE keyspaces;
   ```

   The output is a list of system keyspaces containing tables of details about database objects and cluster configuration:

   ```
   system system_auth system_traces
   ```
3. Create a keyspace.

   ```
   CREATE KEYSPACE mykeyspace WITH REPLICATION = { 'class' : 'SimpleStrategy',
    'replication_factor' : 1 };
   ```
4. Use the keyspace, just as you would use an SQL database.

   ```
   USE mykeyspace;
   ```
5. Create a simple table with three columns for the ids, first names, and last names of users.

   ```
   CREATE TABLE users (
     user_id int PRIMARY KEY,
     fname text,
     lname text
   );
   ```
6. Check that your table and keyspace has been created.

   ```
   DESCRIBE TABLES;
   ```

   The output is the list of tables, in the case just one, in the keyspace you're using:

   ```
   users
   ```
7. Insert the ids, first name, and last name of a few users into the table.

   ```
   INSERT INTO users (user_id,  fname, lname)
     VALUES (1745, 'john', 'smith');
   INSERT INTO users (user_id,  fname, lname)
     VALUES (1744, 'john', 'doe');
   INSERT INTO users (user_id,  fname, lname)
     VALUES (1746, 'john', 'smith');
   ```
8. Retrieve all the data from the users table.

   ```
   SELECT * FROM users;
   ```

   The output lists the data in the order Cassandra stores it.

   ```
    user_id | fname | lname
   ---------+-------+-------
       1745 |  john | smith
   ```

```
    1744 |  john |   doe
    1746 |  john | smith
```

9. Retrieve data about users whose last name is smith by first creating an index, and then querying the table.

   ```
   CREATE INDEX ON users (lname);
   ```

   Note: You need the index because your WHERE clause will use a column that isn't the primary key.

   ```
   SELECT * FROM users WHERE lname = 'smith';
   ```

   The output lists the data:

   ```
    user_id | fname | lname
   ---------+-------+-------
       1745 |  john | smith
       1746 |  john | smith
   ```

10. Drop the users table.

    ```
    DROP TABLE users;
    ```

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