



DataStax 6.7 Installation Guide (Latest version)

© 2018 DataStax, Inc. All rights reserved.
DataStax, Titan, and TitanDB are registered trademark of DataStax,
Inc. and its subsidiaries in the United States and/or other countries.

Apache Cassandra, Apache, Tomcat, Lucene, Solr, Hadoop, Spark,
TinkerPop, and Cassandra are trademarks of the Apache Software Foundation
or its subsidiaries in Canada, the United States and/or other countries.

Table of Contents

Installing DSE 6.7	5
Which install method should I use?.....	5
Installing supporting software.....	6
Installing OpenJDK 8 on RHEL-based Systems.....	6
Installing OpenJDK 8 on Debian-based Systems.....	7
Installing Oracle JRE or JDK on RHEL-based Systems.....	8
Installing Oracle JRE or JDK on Debian or Ubuntu systems.....	9
Installing Python 2.7 on older RHEL installs.....	11
Installing a cluster using Lifecycle Manager 6.7.....	12
Installing from the Yum package.....	17
Installing from the Debian package.....	21
Install from the tarball on any Linux distribution.....	25
Installing on cloud providers.....	31
Installing on Docker.....	32
Uninstalling DSE.....	33
Default DSE file locations.....	34
Package installations.....	34
Tarball installations.....	39
Installing DSE 6.7 Tools	42
Installing DataStax Studio 6.7.....	42
Installing DSE Graph Loader.....	43
Installing DataStax Bulk Loader 1.2.....	44
Installing the DataStax Kafka Connector 1.0.....	44
Installing DSE OpsCenter 6.7	47
Installing from the RPM package.....	47
Installing from the Debian package.....	48
Installing from the tarball on any Linux distribution.....	50

Installing on Docker.....	52
Uninstalling OpsCenter.....	52
Installing DataStax Agents 6.7.....	53
Installing DataStax Agents automatically.....	53
Installing DataStax Agents manually.....	57
Setting permissions to run the DataStax Agent as the DSE user.....	60
Configuring JAVA_HOME.....	62

Installing DataStax Enterprise 6.7

Which install method should I use?

DataStax Enterprise 6.7 installation types

You can install DataStax Enterprise (DSE) in several ways, depending on the purpose of the installation, the type of operating system, and the available permissions. Be sure to install on a [supported platform](#).

Installation method	About
DSE OpsCenter Lifecycle Manager	<p>The Lifecycle Manager (LCM) can install DSE. It is fully integrated with DSE OpsCenter. LCM provides:</p> <ul style="list-style-type: none"> • A web-based graphical interface for installing and configuring DSE. • Performs a Services installation using RHEL or Debian packages. • Ability to configure and update all DSE settings. • Installs the necessary system requirements automatically, including Java. • Integrates with OpsCenter Best Practice Service, which detects many suboptimal configuration settings. • Automation via the LCM API. <p>Installing DSE using LCM requires Installing DSE OpsCenter (page 47), bringing your own instances on a supported platform with SSH and Python installed, and root permissions on the target nodes.</p>
Installing DataStax Enterprise 6.7 on RHEL-based systems using Yum (page 17)	Installs DSE using Yum repositories on RHEL-based systems. Requires root permissions. Typically used in production environments.
Installing DataStax Enterprise 6.7 on Debian-based systems using APT (page 21)	Installs DSE on Debian-based systems using APT. Requires root permissions. Typically used in production environments.
Binary tarball installer (page 25)	Installs DSE on any supported Linux-based platform. Does not require root permissions.
Installing and deploying DSE on cloud platforms (page 31)	Instructions for installing and deploying DSE on CenturyLink Cloud, Google Compute Engine, Microsoft Azure, and Amazon EC2.
Installing DSE 6.7 on Docker (page 32)	Instructions for using DataStax Docker images to create DSE server, DSE OpsCenter, and DataStax Studio containers in non-production environments.

Caution: If you have installed hot fixes, be sure to manually remove the hot fix JAR files before upgrading DataStax Enterprise.

Installing DataStax Enterprise drivers

For version compatibility, see the [DataStax drivers](#) page.

Table 1: Development status of DataStax drivers

Actively developed drivers	Maintenance mode drivers ¹
C/C++ driver	PHP driver
C# driver C# DSE Graph Extension	Ruby driver
Java driver (DSE Graph Extension included)	
Node.js driver Node.js DSE Graph Extension	
Python driver Python Graph Extension	

Installing supporting software on DataStax Enterprise 6.7

Installing OpenJDK 8 on RHEL-based Systems

Configure your operating system to use the [OpenJDK 8](#) (1.8.0_151 minimum).

Tip: Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8. This change is due to the end of public updates for Oracle JRE/JDK 8.

Java 9 is not supported.

In a terminal:

1. Install the OpenJDK 8:

```
$ su -c "yum install java-1.8.0-openjdk"
```

2. If you have more than one Java version installed on your system use the following command to switch versions:

```
$ sudo alternatives --config java
```

1. **Supported by DataStax, but only critical bug fixes will be included in new versions.**

3. Make sure your system is using the correct JDK:

```
$ java -version
```

```
openjdk version "1.8.0_171"  
OpenJDK Runtime Environment (build 1.8.0_171-8u171-  
b11-0ubuntu0.16.04.1-b11)  
OpenJDK 64-Bit Server VM (build 25.171-b11, mixed mode)
```

Installing OpenJDK 8 on Debian-based Systems

Configure your operating system to use the [OpenJDK 8](#) (1.8.0_151 minimum).

Tip: Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8. This change is due to the end of public updates for Oracle JRE/JDK 8.

Java 9 is not supported.

In a terminal:

1. Download and update the repositories:

```
$ sudo apt-get update
```

2. Install OpenJDK 8:

```
$ sudo apt-get install openjdk-8-jdk
```

3. If you have more than one Java version installed on your system use the following command to switch versions:

```
$ sudo update-alternatives --config java
```

4. Make sure your system is using the correct JDK:

```
$ java -version
```

```
openjdk version "1.8.0_171"  
OpenJDK Runtime Environment (build 1.8.0_171-8u171-  
b11-0ubuntu0.16.04.1-b11)
```

```
OpenJDK 64-Bit Server VM (build 25.171-b11, mixed mode)
```

Installing Oracle JRE or JDK on RHEL-based Systems

Attention: Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8. This change is due to the end of public updates for Oracle JRE/JDK 8.

Configure your system to use the latest version of [Oracle Java SE 8 JRE or JDK](#). The minimum supported version is 1.8u151. Java 9 and later are not supported.

Note: JDK provides more classes and tools for support and troubleshooting operations.

1. Check if Java is installed:

```
$ java -version
```

If Oracle Java, the results should look like:

```
java version "1.8.0_181"  
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

2. To install Oracle Java, go to [Oracle Java SE Downloads](#), accept the license agreement, and download the installer for your distribution.

Note: If installing the Oracle Java in a cloud environment, accept the license agreement, download the installer to your local client, and then use `scp` (secure copy) to transfer the file to your cloud machines.

3. From the directory where you downloaded the package, run the install:

```
$ sudo rpm -ivh jdk-8uversion-linux-x64.rpm
```

The RPM installs the JDK into the `/usr/java/` directory.

4. Set your system to use the Oracle JDK:

```
$ sudo alternatives --install /usr/bin/java java /usr/java/  
jdk1.8.0_version/bin/java 200000
```

5. Use the `alternatives` command to switch to the Oracle JDK.

```
$ sudo alternatives --config jav
```


Note: If you have trouble, you may need to set JAVA_HOME and PATH in your profile, such as `.bash_profile`.

The following examples assume that the JDK is in `/usr/java` and which `java` shows `/usr/bin/java`:

- Shell or bash:

```
$ export JAVA_HOME=/usr/java/latest &&
export PATH=$JAVA_HOME/bin:$PATH
```

- C shell (csh):

```
$ setenv JAVA_HOME "/usr/java/latest" &&
setenv PATH $JAVA_HOME/bin:$PATH
```

6. Make sure your system is using the correct JRE or JDK:

```
$ java -version
```

```
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

Installing Oracle JRE or JDK on Debian or Ubuntu systems

Attention: Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8. This change is due to the end of public updates for Oracle JRE/JDK 8.

Configure your system to use the latest version of [Oracle Java SE 8 JRE or JDK](#). The minimum supported version is 1.8u151. Java 9 and later are not supported.

Note: JDK provides more classes and tools for support and troubleshooting operations.

The Oracle Java Platform, Standard Edition (JRE or JDK) has been removed from the official software repositories of Ubuntu and only provides a binary (`.bin`) version. You can get the JRE or JDK from the [Java SE Downloads](#).

1. Check if Java is installed:

```
$ java -version
```

If Oracle Java, the results should look like:

```
java version "1.8.0_181"
```

```
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

2. To install Oracle Java, go to [Oracle Java SE Downloads](#), accept the license agreement, and download the installer for your distribution.

Note: If installing the Oracle Java in a cloud environment, accept the license agreement, download the installer to your local client, and then use `scp` (secure copy) to transfer the file to your cloud machines.

3. Make a directory for the JDK:

```
$ sudo mkdir -p /usr/lib/jvm
```

4. Unpack the tarball and install the JRE or JDK. For example

```
$ sudo tar zxvf jdk-8u65-linux-x64.tar.gz -C /usr/lib/jvm
```

The JDK files are installed into a directory called `/usr/lib/jvm/jdk-8u_version`.

5. Tell the system that there's a new Java version available:

```
$ sudo update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/jdk1.8.0_version/bin/java" 1
```

If updating from a previous version that was removed manually, you may need to execute the above command twice, because you'll get an error message the first time.

6. Set the new JDK as the default using the following command:

```
$ sudo update-alternatives --config java
```

7. Make sure your system is using the correct JRE or JDK:

```
$ java -version
```

```
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

Installing Python 2.7 on older RHEL-based package installations

How to install Python 2.7 on older distributions such as CentOS 6.5.

Note: DataStax Enterprise does not support all older RHEL-based platforms. See [Supported platforms](#).

1. Verify your Python version:

```
$ python -V
```

2. If Python is not version 2.7.x, install it from the operating system software collection:

- For RHEL [Software Collections](#):

```
$ sudo yum update ## optional
```

```
$ sudo yum install scl-utils
```

```
$ sudo yum-config-manager --enable rhel-server-rhscl-6-rpms
```

```
$ sudo yum install python27
```

```
$ sudo scl enable python27 bash
```

```
$ export PYTHONPATH="/usr/lib/python2.7/site-packages/" :
$PYTHONPATH
```

- For CentOS [Software Collections \(SCL \) Repository](#):

```
$ sudo yum update ## optional
```

```
$ sudo yum install scl-utils
```

```
$ sudo yum install centos-release-scl-rh
```

```
$ sudo yum install python27
```

```
$ sudo scl enable python27 bash
```

```
$ export PYTHONPATH="/usr/lib/python2.7/site-packages/" :
$PYTHONPATH
```

3. Verify the update:

```
$ python -V
```

```
Python 2.7.8
```

4. After logging out or restarting, you must enable python 2.7 and set the export command:

```
$ sudo scl enable python27 bash &&  
export PYTHONPATH="/usr/lib/python2.7/site-packages/" :$PYTHONPATH
```

Enabling Python 2.7 in `.bash_profile` or `.bashrc` causes the machine to hang because CentOS 6 relies on Python 2.6 for Yum.

Installing a DataStax Enterprise 6.7 cluster using Lifecycle Manager and DSE OpsCenter 6.7

Lifecycle Manager (LCM) in OpsCenter allows you to easily provision, configure, and install DataStax Enterprise (DSE) clusters.

The workflow when using LCM is to first install OpsCenter on a dedicated server, then use LCM to configure and install the nodes in the DSE clusters.

Prerequisites:

These steps assume that:

- All target machines are running a [supported platform](#).
- You have authentication credentials (either username and password, or an SSH key) for each machine on which DSE will be installed.
- The machines on which DSE will be installed are network accessible by the machine on which OpsCenter will be installed.
- You have access to either the public DSE repository or to an internal mirror set up by your organization.

The topology of your clusters is important. Before you begin, you should know how many clusters, datacenters, and nodes in each datacenter you need. Decide on how you will arrange your workloads within the clusters. Will you use a [single workload per datacenter](#), or a [mixed workload cluster](#)?

The instructions show how to install a cluster consisting of a datacenter with 3 nodes dedicated to transactional workloads with DSE Graph enabled. Internode encryption, client encryption, and both authentication and authorization are enabled on all nodes.

The examples show how to install a cluster consisting of 2 datacenters, each with 3 nodes. One datacenter, `dc1`, will be dedicated to transactional workloads. The other datacenter, `dc2`, will be dedicated to Analytics workloads. Both datacenters will use DSE Graph. Internode and

client encryption will be enabled on all nodes in the cluster, and internal authentication and authorization are enabled.

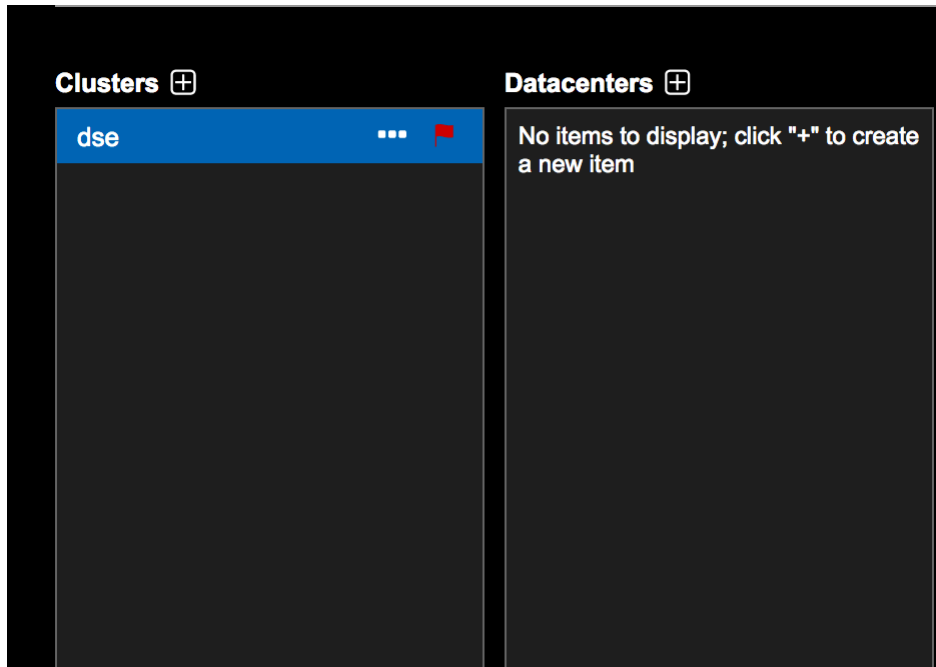
1. [Install and start OpsCenter \(page 47\)](#) on one of the target machines using the installation method of your choice.
2. In a web browser, go to `http://OpsCenter host:8888`.
3. After OpsCenter loads, make sure **Create a new cluster** is selected and click **Get Started**. Lifecycle Manager will load in another window.
4. Click **SSH Credentials** in Lifecycle Manager, then click **Add Credential**. Give a Name to each individual user or key to keep track of the users or keys used in your cluster.
 - a. If you are using password authentication, add the user credentials for each node in the cluster under Login User and Login Password, and click **Save**.
 - b. If you are using a private key, enter the username under Login User, select Private Key and paste in the key, and click **Save**.
5. Click **Config Profiles** in Lifecycle Manager. then **Add Config Profile**.
 - a. Enter a Name for this configuration profile. For example, `dse60`.
 - b. Select the latest version of DataStax Enterprise under DataStax Enterprise Version. For example, `dse v6.0.0`.
6. In **Config Profiles**, select **cassandra.yaml** in the left pane under Cassandra.
 - a. Under **server_encryption_options**, set **internode_encryption** to `all`. Set the **keystore_password** and **truststore_password** to new values.
 - b. Under **client_encryption_options** select `enabled`. Set the **keystore_password** and **truststore_password** to new values.
7. In **Config Profiles** select **dse.yaml**.
 - a. In the **DSE Authenticator Options** section under **authentication_options** select `enabled`.
 - b. In the **DSE Authorizer Options** section under **authorizion_options**, select `enabled`.
8. Select **Save** to save your named Config Profile.

9. Click **Repositories** in Lifecycle Manager, then **Add Repository** to add a DSE repository for installing and updating DSE in your cluster.
 - a. Give a **Name** to this repository. For example, `dse`.
 - b. If you are using an internal repository, click **Access Private Repo** and enter the repository URL and URL key.
 - c. Enter the repository credentials for your repository under **Username** and **Password**.

10. Click **Clusters** in Lifecycle Manager, then click **Adding a Cluster**. This will bring up the **Add Cluster** dialog.
 - a. Give a **Name** to the cluster.
 - b. If you use the same SSH credentials (password or key) across all the machines in your cluster, select the name of the credentials you configured under **SSH Credentials**.
 - c. Select the name of the configuration profile you entered under **Config Profile**. You are prompted to enter passwords. Enter `cassandra` as the **Old Password** for an initial install and enter a new password for the cassandra user.
 - d. Select the name of the repository you entered under **Repository**.
 - e. Click **Save**.

11. Click the name of the cluster you entered, then click the **Add** icon by **Datacenters**. This will bring up the **Add Datacenter** dialog.

Figure 1: Adding a datacenter



12. Configure the transactional datacenter.

- a. Enter the name of your datacenter under **Name**.
- b. If you have different SSH credentials in each datacenter, select the name of the SSH configuration that applies to this datacenter.
- c. Select the workload for this datacenter under **Workload**. For the first datacenter, leave **Workload** set to `Cassandra` to create a transactional workload.
- d. Select **DSE Graph**.
- e. Click **Save**.

13. Click the Add icon by Datacenters to add and configure the second analytics datacenter.

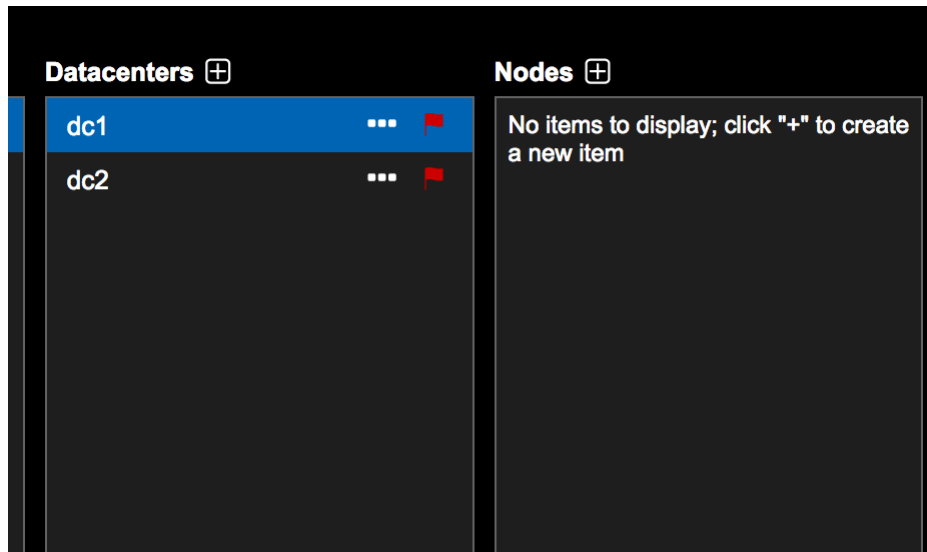
- a. Enter the name of your datacenter under **Name**.
- b. If you have different SSH credentials in each datacenter, select the name of the SSH configuration that applies to this datacenter.
- c. Select the workload for this datacenter under **Workload**. For the second datacenter, set **Workload** to `Spark` to create an analytics workload.
- d. Select **DSE Graph**.

- e. Click **Save**.

14. Add nodes to the datacenters.

- a. Select the datacenter name, then click the **Add** icon next to **Nodes**.

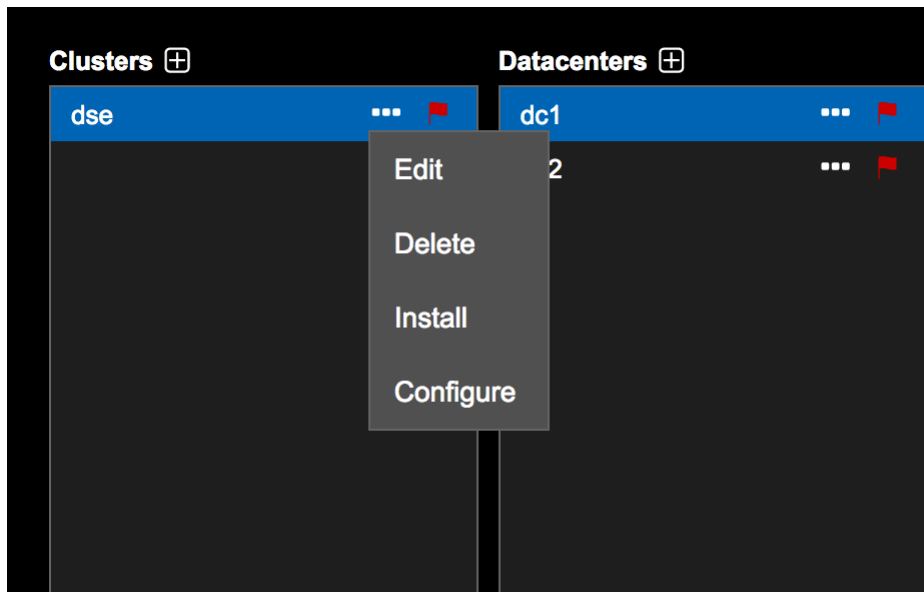
Figure 2: Adding nodes to a datacenter



- b. Enter a **Name** for the node. For example, `dse-transactional-1`.
- c. Enter the IP address for the node under **SSH IP Address**.
- d. If you have different SSH credentials for each node, select the name of the SSH configuration under **SSH Credentials**.
- e. Enter the IP Address in **Native Transport (RPC) Address** if you want to override the default, which is the **SSH IP Address**.
- f. Select **Save**.
- g. Repeat these steps for each node in each datacenter.

15. Select the cluster name under **Clusters**, then click the drop menu to the right of the cluster name, then click **Install** to open the **Run Installation Job** dialog.

Figure 3: Running the Install job



- a. Click **Submit**. The installation job has been queued by the Lifecycle Manager.
- b. Click **View Job Summary** to track the progress of the installation.

What's next:

Go the main OpsCenter interface by clicking **OpsCenter Monitoring**.

Installing DataStax Enterprise 6.7 on RHEL-based systems using Yum

Instructions for installing DataStax Enterprise (DSE) 6.7 on RHEL-based systems using Yum.

To install on SUSE, use the [binary tarball installation \(page 25\)](#).

Some things to know about installing DSE:

- The latest version of DataStax Enterprise 6.7 is 6.7.0.
- When installed from a package (Yum or APT), DataStax Enterprise runs as a service. The service initialization script is located in `/etc/init.d/dse`. Run levels are not set by the package.
- This procedure installs DSE 6.7 and the DataStax Agent. It does not install [OpsCenter \(page 47\)](#), [DataStax Studio](#), [Graph Loader](#), or [DataStax Bulk Loader](#).
- When connecting to DSE 6.7 from OpsCenter, use version OpsCenter 6.7; earlier versions are not supported. See [DataStax OpsCenter compatibility with DSE](#).

Warning: When DSE is installed, it creates a `cassandra` user in the database. Do not use the `cassandra` user in production. Failing to do so is a security risk. See [Adding a superuser login](#).

Prerequisites:

- Root or sudo access.
- A [supported platform](#).
- Your DataStax Academy [registration](#) email address and Downloads Key or Profile Name and password.
- Yum Package Management application.
- Configure your operating system to use the latest version of **Java 8**:
 - # Recommended. [OpenJDK 8](#) (1.8.0_151 minimum)
 - Note:** Recommendation changed due to the end of public updates for Oracle JRE/JDK 8. See [Oracle Java SE Support Roadmap](#).
 - # Supported. [Oracle Java SE 8 \(JRE or JDK\)](#) (1.8.0_151 minimum)
- [RedHat-compatible distributions](#) require EPEL (Extra Packages for Enterprise Linux).
- Python 2.7.x

For older RHEL distributions, see [Installing Python 2.7 on older RHEL-based package installations \(page 11\)](#).

Note: For more information about managing Java, see [Managing Java installs](#). For more information about choosing a Java vendor, see [Choosing a Java vendor in Lifecycle Manager](#).

Table 2: Hardware requirements

Requirement	Minimum	Production
CPUs	2	16
Memory	8 GB	24 GB
Data directory	20 GB	200 GB
Commit log directory	20 GB	200 GB
Saved caches directory	20 GB	200 GB
Logs directory	20 GB	200 GB
Also see Recommended production settings and the DataStax Enterprise Reference Architecture white paper.		

In a terminal window:

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

```
$ java -version
```

If OpenJDK, the results should look like:

```
openjdk version "1.8.0_171"
OpenJDK Runtime Environment (build 1.8.0_171-8u171-
b11-0ubuntu0.16.04.1-b11)
OpenJDK 64-Bit Server VM (build 25.171-b11, mixed mode)
```

If Oracle Java, the results should look like:

```
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

If not OpenJDK 8 or Oracle Java 8, see [Installing supporting software on DataStax Enterprise 6.7 \(page 6\)](#).

2. Install the libaio package. For example:

```
$ sudo yum install libaio
```

3. Add the DataStax Yum repository to a file called `/etc/yum.repos.d/datastax.repo`:

Note: Set the `gpgcheck=1` to perform a GPG signature check.

```
[datastax]
name = DataStax Repo for DataStax Enterprise
baseurl=https://DSA_email_address:downloads_key@rpm.datastax.com/
enterprise/
enabled=1
gpgcheck=0
```

where `DSA_email_address` and `downloads_key` are your [DataStax Academy email address](#) and [My Downloads Key](#). Depending on your environment, you might need to replace `@` in your email address with `%40` and escape any character in your password that is used in your operating system's command line. Examples: `\!` and `\|`.

For backward compatibility, you can use your DataStax Academy Profile Name and password instead of your email address and Downloads Key.

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

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

5. Install the DataStax Enterprise package:

- Install the latest version (6.7.0):

```
$ sudo yum install dse-full
```

- **Optional:** Install the demos:

Attention: Installing the DSE demos is not recommended for production. Only install the demos in development environments to run tutorials.

```
$ sudo yum install dse-demos
```

DataStax Enterprise is ready for additional configuration. See [What's next \(page 21\)](#).

6. Single-node cluster installations only:

- a. Start DataStax Enterprise:

```
$ sudo service dse start
```

For more start options, see [Starting DataStax Enterprise as a service](#).

- b. Verify that DataStax Enterprise is running:

```
$ nodetool status
```

Results using vnodes:

```
Datacenter: Cassandra
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens   Owns    Host ID
      Rack
UN 127.0.0.1    82.43 KB     128     ?
    40725dc8-7843-43ae-9c98-7c532b1f517e rack1
```

Results not using vnodes:

```
Datacenter: Analytics
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Owns    Host ID
      Token          Rack
```

```
UN 172.16.222.136 103.24 KB ? 3c1d0657-0990-4f78-
a3c0-3e0c37fc3a06 1647352612226902707 rack1
```

What's next:

- You must change or delete the `cassandra` user created on installation. See [Adding a superuser login](#).
- If performing an upgrade, go to the next step in the [Upgrade Guide](#).
- [Configuring DataStax Enterprise](#) - Settings for DSE Advanced Security, In-Memory, DSE Advanced Replication, DSE Multi-Instance, DSE Tiered Storage, and more.
- [Default file locations for package installations \(page 34\)](#)
- [Default file locations for tarball installations \(page 39\)](#)
- [Changing logging locations after installation](#).
- [Starting and stopping DataStax Enterprise](#).
- [Preparing DataStax Enterprise for production](#).
- [Recommended production settings](#).
- [Planning and testing DSE cluster deployments](#).
- [Configuring the heap dump directory to avoid server crashes](#).
- [DataStax Studio documentation](#).
- [Installing DataStax Enterprise drivers \(page 6\)](#).

Installing DataStax Enterprise 6.7 on Debian-based systems using APT

Use these instructions for installing DataStax Enterprise (DSE) 6.7 on Debian-based systems using APT.

Some things to know about installing DSE:

- The latest version of DataStax Enterprise 6.7 is 6.7.0.
- When installed from a package (Yum or APT), DataStax Enterprise runs as a service. The service initialization script is located in `/etc/init.d/dse`. Run levels are not set by the package.
- This procedure installs DSE 6.7 and the DataStax Agent. It does not install [OpsCenter \(page 47\)](#), [DataStax Studio](#), [Graph Loader](#), or [DataStax Bulk Loader](#).
- When connecting to DSE 6.7 from OpsCenter, use version OpsCenter 6.7; earlier versions are not supported. See [DataStax OpsCenter compatibility with DSE](#).

Warning: When DSE is installed, it creates a `cassandra` user in the database. Do not use the `cassandra` user in production. Failing to do so is a security risk. See [Adding a superuser login](#).

Prerequisites:

- Root or sudo access.

- A [supported platform](#).
- Your DataStax Academy [registration](#) email address and Downloads Key or Profile Name and password.
- Aptitude Package Management (APT) application.
- Configure your operating system to use the latest version of **Java 8**:
 - # Recommended. [OpenJDK 8](#) (1.8.0_151 minimum)

Note: Recommendation changed due to the end of public updates for Oracle JRE/JDK 8. See [Oracle Java SE Support Roadmap](#).
 - # Supported. [Oracle Java SE 8 \(JRE or JDK\)](#) (1.8.0_151 minimum)
- Python 2.7.x

Table 3: Hardware requirements

Requirement	Minimum	Production
CPUs	2	16
Memory	8 GB	24 GB
Data directory	20 GB	200 GB
Commit log directory	20 GB	200 GB
Saved caches directory	20 GB	200 GB
Logs directory	20 GB	200 GB
Also see Recommended production settings and the DataStax Enterprise Reference Architecture white paper.		

In a terminal window:

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

```
$ java -version
```

If OpenJDK, the results should look like:

```
openjdk version "1.8.0_171"
OpenJDK Runtime Environment (build 1.8.0_171-8u171-
b11-0ubuntu0.16.04.1-b11)
OpenJDK 64-Bit Server VM (build 25.171-b11, mixed mode)
```

If Oracle Java, the results should look like:

```
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

If not OpenJDK 8 or Oracle Java 8, see [Installing supporting software on DataStax Enterprise 6.7 \(page 6\)](#).

2. Install the libaio package. For example:

```
$ sudo apt-get install libaio1
```

3. Add a DataStax repository file called `/etc/apt/sources.list.d/datastax.sources.list`:

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

where `DSA_email_address` and `downloads_key` are your [DataStax Academy email address](#) and [My Downloads Key](#). Depending on your environment, you might need to replace `@` in your email address with `%40` and escape any character in your password that is used in your operating system's command line. Examples: `\!` and `\|`.

For backward compatibility, you can use your DataStax Academy Profile Name and password instead of your email address and Downloads Key.

4. Add the DataStax repository key:

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

5. Update apt-get:

```
$ sudo apt-get update
```

6. Install the DataStax Enterprise package:

- Install the latest version (6.7.0):

```
$ sudo apt-get install dse-full
```

- **Optional:** Install the demos:

Attention: Installing the DSE demos is not recommended for production. Only install the demos in development environments to run tutorials.

```
$ sudo apt-get install dse-demos
```

DataStax Enterprise is ready for additional configuration. See [What's next \(page 24\)](#).

7. Single-node cluster installations only:

a. Start DataStax Enterprise:

```
$ sudo service dse start
```

For more start options, see [Starting DataStax Enterprise as a service](#).

b. Verify that DataStax Enterprise is running:

```
$ nodetool status
```

Results using vnodes:

```
Datacenter: Cassandra
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens   Owns    Host ID
      Rack
UN  127.0.0.1    82.43 KB     128      ?
    40725dc8-7843-43ae-9c98-7c532b1f517e rack1
```

Results not using vnodes:

```
Datacenter: Analytics
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Owns    Host ID
      Token          Rack
UN  172.16.222.136 103.24 KB     ?       3c1d0657-0990-4f78-
a3c0-3e0c37fc3a06 1647352612226902707 rack1
```

What's next:

- You must change or delete the `cassandra` user created on installation. See [Adding a superuser login](#).
- If performing an upgrade, go to the next step in the [Upgrade Guide](#).
- [Configuring DataStax Enterprise](#) - Settings for DSE Advanced Security, In-Memory, DSE Advanced Replication, DSE Multi-Instance, DSE Tiered Storage, and more.
- [Default file locations for package installations \(page 34\)](#)
- [Default file locations for tarball installations \(page 39\)](#)
- [Changing logging locations](#) after installation.

- [Starting and stopping DataStax Enterprise.](#)
- [Preparing DataStax Enterprise for production.](#)
- [Recommended production settings.](#)
- [Planning and testing DSE cluster deployments.](#)
- [Configuring the heap dump directory to avoid server crashes.](#)
- [DataStax Studio documentation.](#)
- [Installing DataStax Enterprise drivers \(page 6\).](#)

Installing DataStax Enterprise 6.7 using the binary tarball

Use these instructions for installing DataStax Enterprise (DSE) on supported Linux-based platforms using a binary tarball.

Some things to know about installing DSE:

- The latest version of DataStax Enterprise 6.7 is 6.7.0.
- When installed from the binary tarball:
 - # DataStax Enterprise runs as a stand-alone process.
 - # You can install DSE with or without root permissions.
- When DSE is installed, it creates a `cassandra` user in the database and runs as this user. Do not use the `cassandra` user in production. Using the `cassandra` user is a security risk. See [Adding a superuser login](#).

Prerequisites:

- A [supported platform](#).
- Your DataStax Academy [registration](#) email address and Downloads Key or Profile Name and password.
- Configure your operating system to use the latest version of **Java 8**:
 - # Recommended. [OpenJDK 8](#) (1.8.0_151 minimum)
 - Note:** Recommendation changed due to the end of public updates for Oracle JRE/JDK 8. See [Oracle Java SE Support Roadmap](#).
 - # Supported. [Oracle Java SE 8 \(JRE or JDK\)](#) (1.8.0_151 minimum)
- RedHat-compatible distributions require EPEL (Extra Packages for Enterprise Linux).
- Python 2.7.x

For older RHEL distributions, see [Installing Python 2.7 on older RHEL-based package installations \(page 11\)](#).

Table 4: Hardware requirements

Requirement	Minimum	Production
CPUs	2	16
Memory	8 GB	24 GB
Data directory	20 GB	200 GB
Commit log directory	20 GB	200 GB
Saved caches directory	20 GB	200 GB
Logs directory	20 GB	200 GB

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

In a terminal window:

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

```
$ java -version
```

If OpenJDK, the results should look like:

```
openjdk version "1.8.0_171"
OpenJDK Runtime Environment (build 1.8.0_171-8u171-
b11-0ubuntu0.16.04.1-b11)
OpenJDK 64-Bit Server VM (build 25.171-b11, mixed mode)
```

If Oracle Java, the results should look like:

```
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

If not OpenJDK 8 or Oracle Java 8, see [Installing supporting software on DataStax Enterprise 6.7 \(page 6\)](#).

2. Install the libaio package. For example:

- RHEL platforms:

```
$ sudo yum install libaio
```

- Debian platforms:

```
$ sudo apt-get install libaiol
```

Installing the latest version (6.7.0)

3. When installing from the binary tarball, you can either download the tarball and then extract the files, or use curl.

- Download and extract the latest version tarball (6.7.0):
 - a. Using your DataStax Academy [registration](#) email address and Downloads Key or Profile Name and password, download the tarball from [Download DataStax Enterprise](#).
 - b. Extract the files:

```
$ tar -xzvf dse-6.7.0.tar.gz
```

- Use curl to install the latest version (6.7.0):

Caution: If you choose this method, your password is retained in the shell history. To avoid this security issue, DataStax recommends using curl with the `--netrc` or `--netrc-file` option.

Download and extract the tarball using curl:

```
$ curl --user DSA_email_address:downloads_key -L \
https://downloads.datastax.com/enterprise/dse.tar.gz | tar xz
```

where `DSA_email_address` and `downloads_key` are your [DataStax Academy email address](#) and [My Downloads Key](#). Depending on your environment, you might need to replace `@` in your email address with `%40` and escape any character in your password that is used in your operating system's command line. Examples: `!\` and `\\`.

For backward compatibility, you can use your DataStax Academy Profile Name and password instead of your email address and Downloads Key.

The files are downloaded and extracted into the 6.7 directory.

4. You can use the default data and logging directory locations or define your locations:

- **Default directory locations:** If you want to use the default data and logging directory locations, create and change ownership for the following:

```
# /var/lib/cassandra
# /var/log/cassandra (includes audit directory, debug.log, gremlin.log,
solvalidation.log, system.log)
# /var/lib/dsefs
# /var/lib/spark
```

```
# /var/log/spark

$ sudo mkdir -p /var/lib/cassandra; sudo chown -R $USER:$GROUP /
var/lib/cassandra &&
  sudo mkdir -p /var/log/cassandra; sudo chown -R $USER:$GROUP /
var/log/cassandra &&
  sudo mkdir -p /var/lib/dsefs; sudo chown -R $USER:$GROUP /var/
lib/dsefs &&
  sudo mkdir -p /var/lib/spark; sudo chown -R $USER:$GROUP /var/
lib/spark &&
  sudo mkdir -p /var/log/spark; sudo chown -R $USER:$GROUP /var/
log/spark &&
  sudo mkdir -p /var/lib/spark/rdd; sudo chown -R $USER:$GROUP /
var/lib/spark/rdd &&
  sudo mkdir -p /var/lib/spark/worker; sudo chown -R $USER:
$GROUP /var/lib/spark/worker
```

- **Define your own directory locations:** If you want to define your own data and logging directory locations:
 - a. In the *installation_location*, make the directories for data and logging directories. For example:

```
$ mkdir dse-data &&
  cd dse-data &&
  mkdir commitlog &&
  mkdir saved_caches &&
  mkdir hints &&
  mkdir cdc_raw
```

- b. Go the directory containing the `cassandra.yaml` file:

```
$ cd installation_location/resources/cassandra/conf
```

- c. Edit the following lines in the `cassandra.yaml` file:

```
data_file_directories: full_path_to_installation_location/dse-
data
commitlog_directory: full_path_to_installation_location/dse-
data/commitlog
saved_caches_directory: full_path_to_installation_location/dse-
data/saved_caches
hints_directory: full_path_to_installation_location/dse-data/
hints
cdc_raw_directory: full_path_to_installation_location/cdc_raw
```

5. You can use either the default Spark data and logging directory locations or define your locations:

- **Default directory locations:** If you want to use the default Spark directory locations, create and change ownership for the following:

```
# /var/lib/dsefs
# /var/lib/spark
# /var/log/spark
```

```
$ sudo mkdir -p /var/lib/dsefs; sudo chown -R $USER:$GROUP /var/
lib/dsefs &&
  sudo mkdir -p /var/lib/spark; sudo chown -R $USER:$GROUP /var/
lib/spark &&
  sudo mkdir -p /var/log/spark; sudo chown -R $USER:$GROUP /var/
log/spark &&
  sudo mkdir -p /var/lib/spark/rdd; sudo chown -R $USER:$GROUP /
var/lib/spark/rdd &&
  sudo mkdir -p /var/log/spark/master; sudo chown -R $USER:
$GROUP /var/log/spark/master &&
  sudo mkdir -p /var/log/spark/alwayson_sql; sudo chown -R $USER:
$GROUP /var/log/spark/alwayson_sql &&
  sudo mkdir -p /var/lib/spark/worker; sudo chown -R $USER:
$GROUP /var/lib/spark/worker
```

- **Define your own directory locations:** If you want to define your own Spark directory locations:
 - a. In the *installation_location*, make the directories for data and logging directories. For example:

```
$ mkdir dsefs; chown -R $USER:$GROUP dsefs &&
mkdir spark; chown -R $USER:$GROUP spark &&
cd spark &&
mkdir log; chown -R $USER:$GROUP log &&
mkdir rdd; chown -R $USER:$GROUP rdd &&
mkdir worker; chown -R $USER:$GROUP worker &&
cd log &&
mkdir worker; chown -R $USER:$GROUP worker &&
mkdir master; chown -R $USER:$GROUP master &&
mkdir alwayson_sql; chown -R $USER:$GROUP alwayson_sql
```

- b. Go the directory containing the spark-env.sh file:

```
$ cd installation_location/resources/spark/conf
```

- c. Uncomment and update the following lines in the spark-env.sh file:

```
export SPARK_WORKER_DIR="full_path_to_installation_location/
spark/worker"
export SPARK_EXECUTOR_DIRS="full_path_to_installation_location/
spark/rdd"
```

```
export
  SPARK_WORKER_LOG_DIR="full_path_to_installation_location/
  spark/log/worker"
export
  SPARK_MASTER_LOG_DIR="full_path_to_installation_location/
  spark/log/master"
export
  ALWAYS_ON_SQL_LOG_DIR="full_path_to_installation_location/
  spark/log/always_on_sql"
```

- d. Go to the directory containing the `dsefs_options` file:

```
cd installation_location/resources/dse/conf
```

- e. Uncomment and update the DSEFS directory in `dse.yaml`:

```
work_dir: full_path_to_installation_location/dsefs
```

DataStax Enterprise is ready for additional configuration. See [What's next \(page 31\)](#).

6. Single-node cluster installations only:

- a. Start DataStax Enterprise from the installation directory:

```
$ bin/dse cassandra
```

where the installation directory is either:

- `/usr/share/dse`
- DataStax Enterprise installation directory

Note: For other start options, see [Starting DataStax Enterprise as a stand-alone process](#).

- b. Verify that DataStax Enterprise is running from the *installation directory*:

```
$ bin/nodetool status
```

Results using `vnodes`:

```
Datacenter: Cassandra
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens   Owns    Host ID
      Rack
```

```
UN 127.0.0.1 82.43 KB 128 ?
40725dc8-7843-43ae-9c98-7c532b1f517e rack1
```

Results not using vnodes:

```
Datacenter: Analytics
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address          Load           Owns           Host ID
      Token
UN 172.16.222.136 103.24 KB ?           3c1d0657-0990-4f78-
a3c0-3e0c37fc3a06 1647352612226902707 rack1
```

What's next:

- You must change or delete the `cassandra` user created on installation. See [Adding a superuser login](#).
- If performing an upgrade, go to the next step in the [Upgrade Guide](#).
- [Configuring DataStax Enterprise](#) - Settings for DSE Advanced Security, In-Memory, DSE Advanced Replication, DSE Multi-Instance, DSE Tiered Storage, and more.
- [Default file locations for package installations \(page 34\)](#)
- [Default file locations for tarball installations \(page 39\)](#)
- [Changing logging locations after installation](#).
- [Starting and stopping DataStax Enterprise](#).
- [Preparing DataStax Enterprise for production](#).
- [Recommended production settings](#).
- [Planning and testing DSE cluster deployments](#).
- [Configuring the heap dump directory to avoid server crashes](#).
- [DataStax Studio documentation](#).
- [Installing DataStax Enterprise drivers \(page 6\)](#).

Installing and deploying DataStax Enterprise 6.7 on cloud platforms

Installing a DataStax Enterprise cluster on various cloud providers

The [DataStax Partner Network](#) provides information on installing and deploying DataStax Enterprise on the following cloud platforms:

- Amazon Web Services (AWS)

Also see AWS Quick Start: [DataStax Enterprise on AWS](#) and [Installing a DataStax Enterprise cluster on Amazon EC2 \(page 32\)](#) below.

- Google Compute Engine

- Microsoft Azure

Installing a DataStax Enterprise cluster on Amazon EC2

DataStax no longer hosts the DataStax ComboAMI. You can install DataStax Enterprise in two ways:

- Create your instances using an AMI for a [supported platform](#) and from a [trusted source](#). Then use the appropriate [install method \(page 5\)](#) for your platform.
- Use the Lifecycle Manager in OpsCenter to easily provision a DataStax Enterprise cluster:
 1. Create your instances using an AMI for a [supported platform](#) and from a [trusted source](#).
 2. Use [DSE OpsCenter Lifecycle Manager](#) to provision and configure your cluster.

Installing DSE 6.7 on Docker

About DataStax-provided Docker images

Use DataStax-provided Docker images to learn DataStax Enterprise (DSE), DSE OpsCenter, and DataStax Studio, try new ideas, and test and demonstrate an application.

DataStax Docker images are hosted on [Docker Hub](#).

Learn more

- Get the full installation instructions in the [DataStax Docker guide](#).
- Contact us to learn more about DSE by [completing this form](#).
- Get sample Docker Compose scripts from the DataStax Docker repository [on Github](#).
- Ask questions in the [DataStax Slack](#) channel to get help from experts.
- Report issues [on Github](#) to help improve the experience with DataStax Docker containers.
- The [It's Here! DataStax Docker Images for DSE](#) blog provides information about how DataStax uses Docker for testing and how to provide feedback.

Use DataStax Docker images to create DataStax Enterprise (DSE) server, DSE OpsCenter, and DataStax Studio containers in non-production environments.

See the [DataStax Academy Quick Downloads](#) page for information on downloading and using the DataStax images for Docker.

Note: A valid DataStax Academy account is required to access the images and documentation.

Uninstalling DataStax Enterprise 6.7

Select the uninstall method for your type of installation.

Uninstalling Debian- and RHEL-based packages

Use this method when you have installed DataStax Enterprise using [APT \(page 21\)](#) or [Yum \(page 17\)](#).

1. Drain and stop the DataStax Enterprise service:

```
nodetool drain &&
sudo service dse stop
```

2. Make sure all services are stopped:

```
$ ps auwx | grep dse
```

3. If services are still running, use the PID to kill the service:

```
$ bin/dse cassandra-stop -p dse_pid
```

4. Remove the installation directories:

RHEL-based packages:

```
$ sudo yum remove "dse-*" "datastax-*"
```

Debian-based packages:

```
$ sudo apt-get purge "dse-*" "datastax-*"
```

Uninstalling the binary tarball

Use this method when you have installed DataStax Enterprise using the [binary tarball \(page 25\)](#).

1. Stop the node:

```
$ bin/dse cassandra-stop
```

2. Make sure all services are stopped:

```
$ ps auwx | grep dse
```

3. If services are still running, use the PID to kill the service:

```
$ bin/dse cassandra-stop -p dse_pid
```

4. Remove the installation directory.

Default DSE file locations

Default file locations for package installations

The default location of the files depends on how DataStax Enterprise is installed.

Default directories for `cassandra.yaml` and `dse.yaml`

Directories	Description
<code>/etc/dse/cassandra/cassandra.yaml</code>	<code>cassandra.yaml</code> is the main configuration file for the DataStax Enterprise database with default configuration for all nodes.
<code>/etc/dse/dse.yaml</code>	<code>dse.yaml</code> is the main configuration file for DataStax Enterprise.

Default database directories

Directories	Description
<code>/var/lib/cassandra/data</code>	commitlog, data, hints, saved_caches directories
<code>/var/log/cassandra</code>	Log files, including: <ul style="list-style-type: none"> • audit directory • debug.log • gremlin.log • solrvalidation.log • system.log You can change logging locations .
<code>/var/run/cassandra</code>	Database process ID (pid) directory
<code>/usr/share/dse/cassandra</code>	Environment settings
<code>/usr/share/dse/cassandra/tools</code>	Tools for testing, starting, using SSTables, plus YAML examples.

Directories	Description
/etc/dse/cassandra	Property files and cqlshrc samples including: <ul style="list-style-type: none"> • <code>cassandra-env.sh</code> • <code>cassandra-rackdc.properties</code> • <code>cassandra-topology.properties</code> • <code>cassandra-topology.yaml</code> • <code>commitlog_archiving.properties</code> • <code>cqlshrc.sample</code> • <code>logback.xml</code>
/etc/init.d	Set node type and other server configuration

Default DSEFS data directory

The default location for the DSEFS data directory is `/var/lib/dsefs`.

Default DSE Graph directories

Directories	Description
/etc/dse/graph/gremlin-console/conf/remote.yaml	Gremlin console configuration for connection to the Gremlin Server, including Kerberos authentication and SSL encryption.
/etc/dse/graph/logback-gremlin-server.xml	GremlinServerFileAppender

Default DSE Search directories

Directories	Description
/usr/share/dse/resources/solr/conf	Solr configuration
/usr/share/dse/demos/wikipedia	Search - Wikipedia demo
/usr/share/dse/solr/web/demos/wikipedia	Search - Wikipedia demo with Tomcat
/var/log/cassandra	Search log messages are in the <code>system.log</code> file

Default Spark directories

Directories	Description
/etc/dse/spark/	<code>spark-env.sh</code> , <code>spark-defaults.conf</code> , <code>spark-daemon-defaults.conf</code>
/usr/share/dse/spark/lib	Spark library
/var/log/spark	Spark Master and Worker logs

Directories	Description
<code>/usr/share/dse/spark/spark-jobserver</code>	Spark Jobserver
<code>/usr/share/dse/demos/portfolio_manager</code>	Spark Portfolio Manager demo
<code>/var/lib/dsefs</code>	The default directory to store the DSE File System data.

Default location for the logback configuration file

Directories	Description
<code>/etc/dse/cassandra/logback.xml</code>	logback.xml is the logback configuration file

Default location audit logs

Directories	Description
<code>/etc/dse/tomcat/conf/server.xml</code> / <code>/etc/dse/tomcat/conf/web.xml</code>	Default location for Tomcat server logs for DSE Search.
<code>/var/log/cassandra/dropped_audit_events.log</code>	Default location for dropped events logs.

Default DSE OpsCenter directories

See the [DSE OpsCenter documentation](#).

Default DSE Multi-Instance configuration files

With DSE Multi-Instance, multiple DataStax Enterprise nodes reside on a single host machine. To segregate the configuration for each DataStax Enterprise node, node-specific directory structures are used to store configuration and operational files. For example, in addition to `/etc/dse/dse.yaml`, the DSE Multi-Instance `dse.yaml` files are stored in `/etc/dse-nodeId/dse.yaml` locations. The `server_id` option is generated in DSE Multi-Instance `/etc/dse-nodeId/dse.yaml` files to uniquely identify the physical server on which multiple instances are running and is unique for each database instance. See [DSE Multi-Instance server_id](#).

Directories	Description
<code>/etc/dse</code>	<code>/etc/dse/dse.yaml</code> is the primary configuration file for DataStax Enterprise
<code>/etc/dse-node1</code>	<code>/etc/dse-node1/dse.yaml</code> is the configuration file for the DataStax Enterprise node in the <code>dse-node1</code> directory
<code>/etc/dse-node2</code>	<code>/etc/dse-node2/dse.yaml</code> is the configuration file for the DataStax Enterprise node in the <code>dse-node2</code> directory

For a comprehensive list of file locations in a DSE Multi-Instance cluster, see [directories for DSE Multi-Instance \(page 37\)](#).

Default DSE Multi-Instance generated directories

With [DSE Multi-Instance](#), these directories are created on the host machine for each node.

Purpose	Directories	Description
DSE Multi-Instance root directory	<code>/etc/defaults</code>	Each DSE Multi-Instance host machine has the <code>/etc/defaults</code> root directory. This default location is not configurable.
DataStax Enterprise node type	<code>/etc/defaults/dse-nodeId</code>	Defines the node type (transactional, search, analytics, graph, and so on).
DataStax Enterprise configuration file	<code>/etc/dse-nodeId/dse.yaml</code>	A configuration file for each node.
DataStax Enterprise process ID (pid) directory	<code>/var/run/dse-nodeId.dse-nodeId.pid</code>	The default DataStax Enterprise process ID (pid) directory for each node.
Database configuration	<code>/etc/dse-nodeId/cassandra/cassandra.yaml</code>	A configuration file for each node.
Database data directory	<code>/var/lib/dse-nodeId/data</code>	The root directory for storing data on each node. Define with <code>dse add-node --data-directory=directory ...</code>
Database log files	<code>/var/log/dse-nodeId/cassandra</code>	The default directory where the <code>audit.log</code> , <code>output.log</code> , <code>solrvalidation.log</code> , and <code>system.log</code> log files are stored for each node. Define with <code>dse add-node --logs-directory ...</code>
Database pid directory	<code>/var/run/dse-nodeId</code>	Database process ID (pid) directory for each node.
Caches directory	<code>/var/lib/dse-nodeId/saved_caches</code>	The table key and row caches directory for each node. Define with <code>dse add-node --saved-caches-directory=directory ...</code>

Purpose	Directories	Description
Commit log files	<code>/var/lib/dse-nodeId/commitlog</code>	The commit log directory for each node. Define with <code>dse add-node --commit-directory=directory ...</code>
Hints directory	<code>/var/lib/dse-nodeId/hints</code>	The hints directory for each node. Define with <code>dse add-node --hints-directory=directory ...</code>
Spark configuration file	<code>/etc/dse-nodeId/spark/spark-env.sh</code>	Spark configuration file <code>spark-env.sh</code> for each node.
Spark Worker data directory	<code>/var/lib/dse-nodeId/spark/worker</code>	The data directory for Spark Worker for each node. Define with <code>dse add-node --spark-worker-directory=directory ...</code>
Spark Worker local node directory	<code>/var/lib/dse-nodeId/spark/rdd</code>	The local directory for Spark Worker for each node. Define with <code>dse add-node --spark-local-directory=directory ...</code>
Spark logs directory	<code>/var/log/dse-nodeId/spark</code>	The Spark logs directory for each node. Define with <code>dse add-node --spark-log-directory=directory ...</code>
Logback configuration	<code>/etc/dse-nodeId/cassandra/logback.xml</code>	Logback configuration file for each node.
Solr configuration	<code>/etc/dse-nodeId/solr</code>	Solr configuration files for each node.
Tomcat log files	<code>/var/log/dse-nodeId/tomcat</code>	The directory for Tomcat server logs. Define with <code>dse add-node --tomcat-logs=directory ...</code>

Licenses and other documents

The default location is *installation_location*. Also see [DataStax Enterprise third-party software](#).

Default file locations for tarball installations

The default location of the files depends on how DataStax Enterprise is installed.

Default installation location

The default *installation_location* depends on where you extracted DataStax Enterprise:

Default directories for cassandra.yaml and dse.yaml

Directories	Description
<i>installation_location</i> / resources/cassandra/conf/ cassandra.yaml	cassandra.yaml is the main configuration file for the database.
<i>installation_location</i> / resources/cassandra/conf/ cassandra.yaml <i>installation_location</i> / resources/dse/conf/dse.yaml	dse.yaml is the main configuration file for DataStax Enterprise.

Default database directories

Directories	Description
<i>installation_location</i> /resources/ cassandra/bin	Commands and utilities, such as nodetool, cqlsh, sstabledump, and sstableloader
<i>installation_location</i> /resources/ cassandra/conf/cassandra.yaml	Property files and cqlshrc samples including: <ul style="list-style-type: none"> cassandra-env.sh cassandra-rackdc.properties cassandra-topology.properties cassandra-topology.yaml commitlog_archiving.properties cqlshrc.sample logback.xml
/var/lib/cassandra or <i>installation_location</i>	commitlog, data, hints, saved_caches directories

Directories	Description
<code>/var/log/cassandra</code>	Log files, including: <ul style="list-style-type: none"> • <code>audit</code> directory • <code>debug.log</code> • <code>gremlin.log</code> • <code>solrvalidation.log</code> • <code>system.log</code> You can change logging locations .

Default DSEFS data directory

The default location for the DSEFS data directory is `/var/lib/dsefs`.

Default DSE Graph directories

Directories	Description
<code>installation_location/resources/graph/gremlin-console/conf/remote.yaml</code>	Gremlin console configuration for connection to the Gremlin Server, including Kerberos authentication and SSL encryption.
<code>installation_location/resources/graph/conf/logback-gremlin-server.xml</code>	GremlinServerFileAppender

Default DSE Search directories

Directories	Description
<code>installation_location/resources/solr/conf</code>	Solr configuration
<code>installation_location/resources/solr/lib</code>	Solr driver
<code>installation_location/demos/wikipedia</code>	Search - Wikipedia demo
<code>/var/log/cassandra</code>	Solr log messages are in the <code>system.log</code> file.

Default Spark directories

Directories	Description
<code>installation_location/resources/spark/conf</code>	<code>spark-env.sh</code> , <code>spark-defaults.conf</code> , <code>spark-daemon-defaults.conf</code>
<code>/var/lib/spark</code>	Spark library
<code>/var/log/spark</code>	Spark Master and Worker logs

Directories	Description
<code>installation_location/resources/spark/spark-jobserver</code>	Spark Jobserver
<code>installation_location/demos/portfolio_manager</code>	Spark Portfolio Manager demo
<code>/var/lib/dsefs</code>	The default directory to store the DSE File System data.

Default Logback-appender directories

Directories	Description
<code>installation_location/resources/cassandra/conf/logback.xml</code>	logback.xml is the logback configuration file

Default location audit logs

Directories	Description
<code>/etc/dse/tomcat/conf/server.xml</code> / <code>/etc/dse/tomcat/conf/web.xml</code>	Default location for Tomcat server logs for DSE Search.
<code>/var/log/cassandra/dropped_audit_events.log</code>	Default location for dropped events logs.

Default DSE OpsCenter directories

See the [OpsCenter documentation](#).

Default directory for Token-generator tool

Location	Description
<code>installation_location/resources/cassandra/tools/bin/token-generator</code>	For manually Generating tokens .

Licenses and other documents

The default location is `installation_location`. Also see [DataStax Enterprise third-party software](#).

Installing DataStax Enterprise 6.7 Tools

Installing DataStax Studio 6.7

Tip: You can install DataStax Studio in a Docker container. DataStax maintains Studio images for Docker. For information on downloading and using the Studio image, see the [DataStax Docker docs](#).

DataStax Studio 6.7 is compatible with DataStax Enterprise (DSE) 6.7. The latest version of Studio is 6.7.0.

Notebooks created in earlier versions of DataStax Studio can be used with new Studio versions:

- When the new version of Studio starts, all notebooks created in an earlier version are automatically upgraded.
- Notebooks are not impacted, lost, or corrupted with the upgrade.
- After the notebooks are upgraded, they are no longer compatible with the earlier version.

For upgrade information, see [Upgrading DataStax Studio](#).

Prerequisites:

To install and run Studio:

- A [supported web browser](#).
- All DataStax Enterprise 6.7 [prerequisites \(page 5\)](#) for your platform.
- Your DataStax Academy [registration](#) email address and Downloads Key or Profile Name and password.
- Like DSE 6.7, which supports the latest version of Java 8, DataStax Studio does not work with Java 9 or later:
 - # Recommended. [OpenJDK 8](#) (1.8u151 minimum)
 - # Supported. [Oracle Java SE 8 \(JRE or JDK\)](#) (1.8u151 minimum)

Verify that the required Java version is used. See [Studio will not start with wrong Java version](#).

- Windows platforms: Windows 7 and 10 with Java 8.

Tip: Ensure Microsoft Windows 7 [libraries](#) are current.

1. If an earlier version of DataStax Studio is installed, before you install the new version, back up the user data directory: .

```
user_home_directory/.datastax_studio
```

See [User data in DataStax Studio](#).

2. Download the Studio tarball from the [DataStax Download page](#) using your DataStax Academy [registration](#) credentials.
3. Unpack Studio using the appropriate method for the file type:

- Linux:

```
tar zxvf datastax-studio-6.7.0.tar.gz
```

The files are downloaded and extracted into the current directory.

- Windows:

- a. From File Explorer, right-click `user_home_directory\datastax-studio-6.7.0.zip`.
- b. Enter the destination folder for Studio.
- c. Click **Extract**.

What's next:

[Starting and stopping DataStax Studio](#).

[Configuring DataStax Studio](#) .

Installing DSE Graph Loader

DSE Graph Loader is not included as part of DataStax Enterprise (DSE) installations. Use these instructions for installing on Linux-based platforms using the binary tarball.

Prerequisites:

- A sufficiently powerful machine should be used to run DSE Graph Loader. The memory requirements must account for caching the serialized vertices during the loading process, up to ten times (10X) the size of the original data. Ensure that enough shared memory (mmap, or buffer cache) is allocated; properties and edges are bound by the speed of the available I/O. The network connection between the DSE Graph Loader machine and the DSE Graph cluster must have sufficient bandwidth.

Note: Do not run DSE Graph Loader on a machine that hosts a DSE Graph node for larger scale datasets.

- Your DataStax Academy [registration](#) email address and Downloads Key or Profile Name and password.
- DataStax Enterprise is installed and configured for DSE Graph.

1. Download the DSE Graph Loader tarball from the [DataStax Download page](#) using your DataStax Academy [registration](#) credentials.
2. Unpack the DataStax Enterprise tarball:

```
$ tar -xzvf dse-graph-loader.tar.gz
```

The files are downloaded and extracted into the `dse-graph-loader` directory.

What's next: [Configure DSE Graph Loader](#)

Installing DataStax Bulk Loader 1.2

The DataStax Bulk Loader tool (`dsbulk`) is designed to provide users with the ability to both load and unload data in and out of DataStax Enterprise (DSE) efficiently and reliably. DataStax recommends using the latest DataStax Bulk Loader 1.2.0. For details, see the [DataStax Bulk Loader guide](#).

The `dsbulk` tool can be run as a standalone tool that remotely connects to a cluster. The tool is not required to run locally on a cluster node, but can be used in this configuration.

1. Download the tarball from the [DataStax Download](#) page using your DataStax Academy [registration](#) credentials.
2. Unpack the distribution:

```
$ tar -xzvf dsbulk.tar.gz
```

The files are downloaded and extracted into the current directory.

What's next: Learn how to [get started](#) with `dsbulk`.

Installing the DataStax Kafka Connector 1.0

Install the connector from the DataStax distribution binary tarball using an account that has write access to the Kafka configuration directory.

Tip: See [About the DataStax Apache Kafka Connector](#).

Support Apache Kafka versions

Install the DataStax Apache Kafka™ Connector on any of following versions:

- Confluent 3.2 and higher
- Apache Kafka 0.10.2 and higher

Stream data using the connector to DataStax Enterprise (DSE) 5.0 and higher.

Prerequisites:

Sign up for an account on [DataStax Academy](#) and configure a Download Key.

Perform the following steps on a Kafka Connect node

1. Using your DataStax Academy credentials, download the tarball from [Download Kafka Connector](#).
2. Extract the files:

```
$ tar xzf kafka-connect-dse-version_number.tar.gz
```

where *version_number* is the connector version number.

The following files are unpacked into a directory such as `kafka-connect-dse-version_number`.

```
LICENSE.txt
README.md
THIRD-PARTY.txt
conf/dse-sink-distributed.json.sample
conf/dse-sink-standalone.properties.sample
kafka-connect-dse-version_number.jar
```

where *version_number* is the DSE connector version number.

3. Configured the DataStax connector JAR using one of the following methods:

- Move the DataStax connector JAR to Kafka plugins directory:

```
mv installation_location/kafka-connect-
dse-version_number.jar kafka_plugins_dir
```

- Configure the path to the JAR:

Apache Kafka 0.11.x and higher - Specify the JAR location in the `plugin.path` parameter in the `connect-standalone.properties` or `connect-distributed.properties` file that is passed to the worker start-up scripts. For example:

```
plugin.path=install_location/kafka-connect-dse-1.0.0-alpha2.jar
```

Apache Kafka versions earlier than 0.11 - To properly load the classes, add the JAR path to the worker CLASSPATH. For example:

```
$ export CLASSPATH=$CLASSPATH:install_location/kafka-connect-
dse-1.0.0-alpha2.jar
```

Note: For Confluent, reference the compatibility table.

4. Copy the sample configuration file from `kafka-connect-dse-version_number/conf/` to the Kafka configuration directory, which is typically the `config` or `etc` directory.

DataStax provides the following sample files in the `conf` directory of the connector distribution package:

- `dse-sink-standalone.properties.sample` for standalone mode. It is a Java properties file that contains all settings with descriptions. Settings with a default value are commented out.
- `dse-sink-distributed.json.sample` for distributed mode. This file is in JSON format and contains all settings, which are enumerated and active. To use the default values, remove settings from the configuration file. JSON does not support comments.

5. Rename the sample file to `dse-sink.properties` or `dse-sink.json`.
6. Update the settings as necessary. See the [DataStax Apache Kafka Connector configuration parameter reference](#).
7. Ensure that the user running Kafka has permission to access the configuration and JAR files.

Installing DSE OpsCenter 6.7

Installing DSE OpsCenter 6.7 from the RPM package

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

For a complete list of supported platforms, see [OpsCenter Supported Platforms](#).

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

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

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

Prerequisites:

Minimum hardware requirements for the machine on which OpsCenter runs:

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

Permission and software requirements:

- [Yum](#) package management utility.
- OpenJDK 8 or Oracle Java SE Runtime Environment 8 (JRE or JDK). Earlier or later versions are not supported. See installing the [OpenJDK](#) or [Oracle JDK \(page 6\)](#).

Attention: Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8. This change is due to the end of public updates for Oracle JRE/JDK 8.

Use OpsCenter Lifecycle Manager to [automatically manage Java and JCE installs](#) for DSE clusters.

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

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

```
[opscenter]
name = DataStax Repository
baseurl = https://DSA_email_address:downloads_key@rpm.datastax.com/enterprise
enabled = 1
```

```
gpgcheck = 0
```

Note: Set the `gpgcheck=1` to perform a GPG signature check.

where `DSA_email_address` and `downloads_key` are your [DataStax Academy email address](#) and [My Downloads Key](#). Depending on your environment, you might need to replace `@` in your email address with `%40` and escape any character in your password that is used in your operating system's command line. Examples: `\!` and `\|`.

For backward compatibility, you can use your DataStax Academy Profile Name and password instead of your email address and Downloads Key.

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

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

3. Install the OpsCenter package:

```
$ sudo yum install opscenter
```

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

4. Start OpsCenter:

```
$ sudo service opscenterd start
```

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

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

What's next:

[Add an existing cluster](#) or provision a new cluster in [Lifecycle Manager](#).

Installing DSE OpsCenter 6.7 from the Debian package

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

For a complete list of supported platforms, see [OpsCenter Supported Platforms](#).

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

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


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

Prerequisites:

Minimum hardware requirements for the machine on which OpsCenter runs:

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

Permission and software requirements:

- APT Package Manager is installed.
- OpenJDK 8 or Oracle Java SE Runtime Environment 8 (JRE or JDK). Earlier or later versions are not supported. See installing the [OpenJDK](#) or [Oracle JDK \(page 6\)](#).

Attention: Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8. This change is due to the end of public updates for Oracle JRE/JDK 8.

Use OpsCenter Lifecycle Manager to [automatically manage Java and JCE installs](#) for DSE clusters.

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

In a terminal window:

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

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

where `DSA_email_address` and `downloads_key` are your [DataStax Academy email address](#) and [My Downloads Key](#). Depending on your environment, you might need to replace `@` in your email address with `%40` and escape any character in your password that is used in your operating system's command line. Examples: `!\` and `\\`.

For backward compatibility, you can use your DataStax Academy Profile Name and password instead of your email address and Downloads Key.

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

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

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

```
$ sudo apt-get update
```

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

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

4. Start OpsCenter:

```
$ sudo service opscenterd start
```

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

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

What's next:

[Add an existing cluster](#) or provision a new cluster in [Lifecycle Manager](#).

Installing DSE OpsCenter 6.7 with a tarball on any Linux distribution

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

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

For a complete list of supported platforms, see [OpsCenter Supported Platforms](#).

Prerequisites:

Minimum hardware requirements for the machine on which OpsCenter runs:

- 2 CPU cores
- 2 GB of RAM available to OpsCenter
- Your DataStax Academy [registration](#) email address and Downloads Key or Profile Name and password.

Permission and software requirements:

- OpenJDK 8 or Oracle Java SE Runtime Environment 8 (JRE or JDK). Earlier or later versions are not supported. See installing the [OpenJDK](#) or [Oracle JDK \(page 6\)](#).

Attention: Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8. This change is due to the end of public updates for Oracle JRE/JDK 8.

Use OpsCenter Lifecycle Manager to [automatically manage Java and JCE installs](#) for DSE clusters.

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

Installing the latest version (6.7.0)

1. You can either download the tarball and then extract the files, or use curl.

- Download and extract the latest version tarball (6.7.0):
 - a. Using your DataStax Academy [registration](#) email address and Downloads Key or Profile Name and password, download the tarball from [Download DataStax OpsCenter](#).
 - b. Extract the files:

```
$ tar -xzvf opscenter-6.7.0.tar.gz
```

The files are extracted into the opscenter-6.7.0 directory.

- Use curl to install the latest version (6.7.0):

Caution: If you choose this method, your password is retained in the shell history. To avoid this security issue, DataStax recommends using curl with the [--netrc](#) or [--netrc-file](#) option.

- a. In a terminal window, download and extract the tarball using curl:

```
$ curl --user DSA_email_address:downloads_key -L \
https://downloads.datastax.com/enterprise/opscenter.tar.gz |
tar xz
```

where *DSA_email_address* and *downloads_key* are your [DataStax Academy email address](#) and [My Downloads Key](#). Depending on your environment, you might need to replace @ in your email address with %40 and escape any character in your password that is used in your operating system's command line. Examples: \! and \|.

For backward compatibility, you can use your DataStax Academy Profile Name and password instead of your email address and Downloads Key.

The files are downloaded and extracted into the opscenter-6.7.0 directory.

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

```
$ cd opscenter-version_number
```

3. Start OpsCenter from the install location:

```
$ bin/opscenter
```

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

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

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

What's next:

[Add an existing cluster](#) or provision a new cluster in [Lifecycle Manager](#).

Installing DSE OpsCenter 6.7 on Docker

Use DataStax Docker images to create DataStax Enterprise (DSE) server, DSE OpsCenter, and DataStax Studio containers in non-production environments.

See the [DataStax Academy Quick Downloads](#) page for information on downloading and using the DataStax images for Docker.

Note: A valid DataStax Academy account is required to access the images and documentation.

Uninstalling DSE OpsCenter 6.7

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

Uninstalling an OpsCenter RPM package

Use this method if OpsCenter was installed using an [RPM \(page 47\)](#) package.

1. [Stop \(page \)](#) OpsCenter.
2. Open a terminal and enter the following command:

```
$ sudo yum remove opscenter
```

Uninstalling an OpsCenter Debian package

Use this method if OpsCenter was installed using a [Debian \(page 48\)](#) package.

1. Stop (page) OpsCenter.
2. Open a terminal and enter the following command:

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

Uninstalling an OpsCenter binary tarball

Use this method if OpsCenter was installed using a [tarball \(page 50\)](#).

1. Stop (page) OpsCenter.
2. Open a terminal and enter the following command:

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

Installing DataStax Agents 6.7

DataStax agents must be installed on every managed node in a cluster and are necessary to perform most of the functionality within DSE OpsCenter. When creating a new cluster with Lifecycle Manager, the DataStax Agent is automatically installed. When adding a new cluster to manage with OpsCenter, you are given the option to automatically or manually install agents. Also see [DataStax Agents Status View](#).

Installing DataStax Agents automatically

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

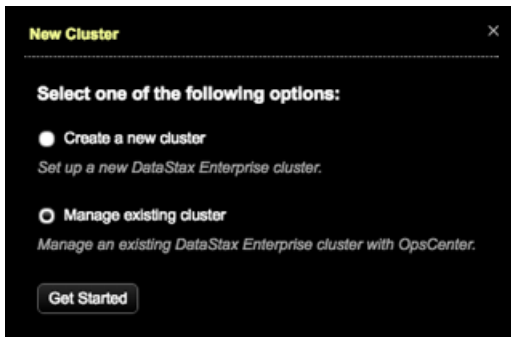
After [adding a cluster](#), OpsCenter will determine whether DataStax Agents are already properly installed and configured. If they are not, you will be prompted to check the status of the Agents and potentially install them from the [Agents](#) tab.

Prerequisites:

- [OpsCenter \(page 47\)](#) must be installed.
- Root or sudo access to the machines where the DataStax 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.
- DataStax Enterprise 6.7. See [DSE OpsCenter compatibility with DSE](#).

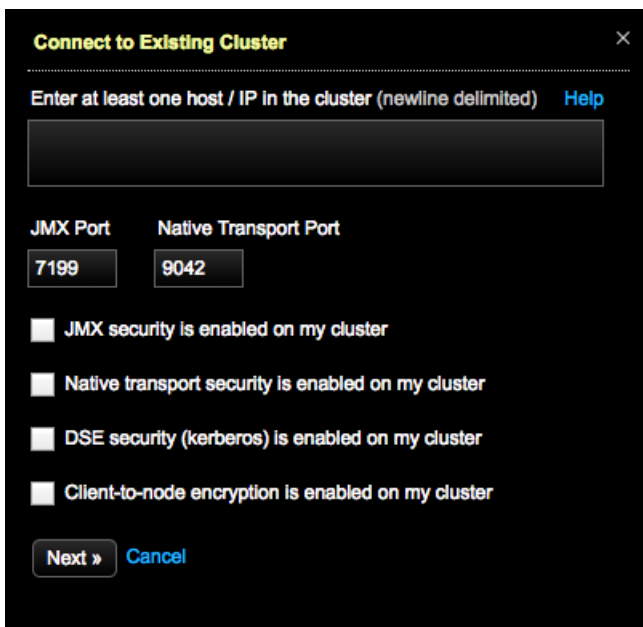
1. Start your DataStax Enterprise cluster and the OpsCenter daemon.
2. Open a browser window and go to the OpsCenter URL at `http://opscenter_host:8888`, where `opscenter_host` is the IP or hostname of the OpsCenter machine.

A dialog prompts you to select a cluster option:



3. Click **Manage Existing Cluster#Get Started**.

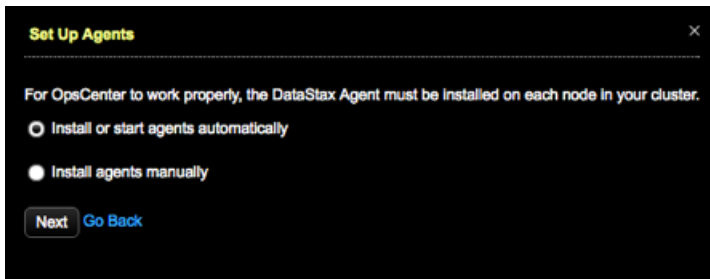
The **Connect to Existing Cluster** dialog appears.



4. Add one or more hostnames or IP addresses of the nodes in the cluster. For best results, use private IP addresses.

5. Click **Next**.

The **Set Up Agents** dialog appears.



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

If your environment requires manual installation of DataStax Agents, click **Install agents manually**. Click the link to access the instructions for [manually installing Agents \(page 57\)](#).

The **Agents Credentials** dialog appears.

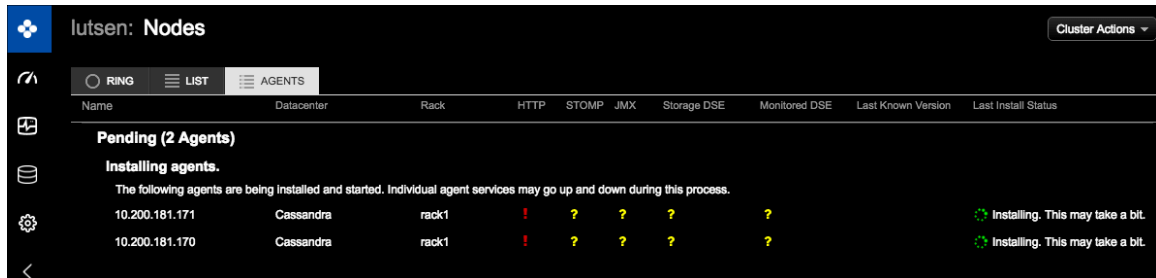
7. Enter SSH credentials to connect to the nodes:

- a. Enter a **Username**. The user must have root or sudo privileges.
- b. A sudo password can be entered in the **Password** box.
- c. Enter a **Private Key**.

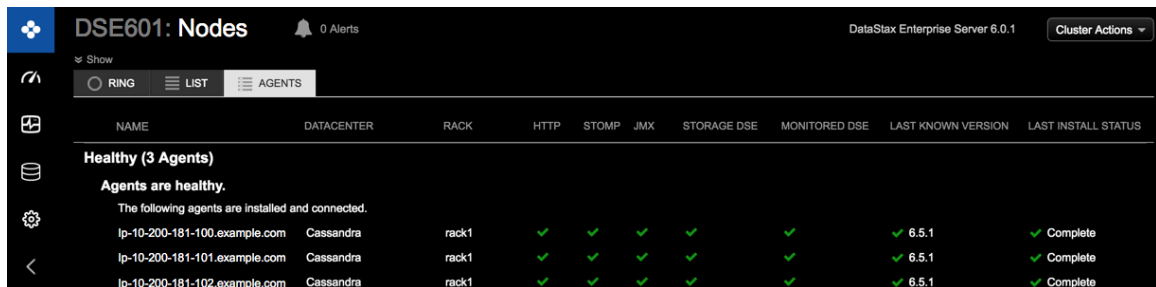
Entered credentials information is not saved or stored.

8. Click **Submit**.

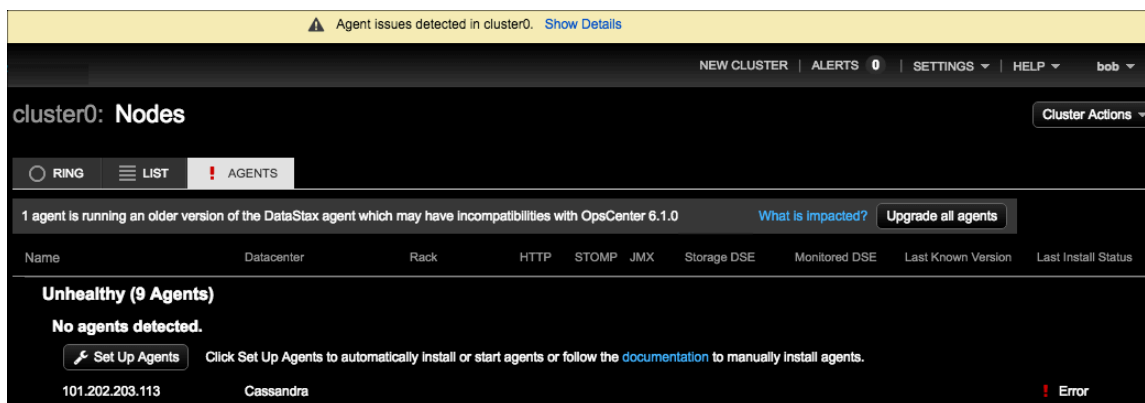
The **Agents tab** opens, starts installing the DataStax Agents, and displays the progress of the agent installation. The DataStax Agent services may go up and down during the installation process.



When the install process successfully completes, the DataStax Agents have been deployed and configured for each managed node in the DataStax Enterprise cluster.



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



If you were unable to install the Agents through the OpsCenter UI, follow the instructions to [manually install the Agents \(page 57\)](#).

Related information:

[DataStax Agents Status View](#) [*View the current installation, configuration, and connection status of DataStax Agents.*] ([page](#))

Installing DataStax Agents manually

If automatic DataStax Agent installation was unsuccessful, manually install the Agents using the procedure appropriate for your installation.

Manually deploying DataStax Agents from RPM

Prerequisites:

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

In a terminal:

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

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

Note: Set the `gpgcheck=1` to perform a GPG signature check.

where `DSA_email_address` and `downloads_key` are your [DataStax Academy email address](#) and [My Downloads Key](#). Depending on your environment, you might need to replace `@` in your email address with `%40` and escape any character in your password that is used in your operating system's command line. Examples: `\!` and `\|`.

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

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

3. Install the DataStax agent:

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

If you are using an earlier version of OpsCenter, add the supported version to the command:

```
$ sudo yum install datastax-agent-6.5.version-1
```

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

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

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

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

6. Start the DataStax agent:

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

Manually deploying DataStax Agents from Debian

Prerequisites:

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

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

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

where `DSA_email_address` and `downloads_key` are your [DataStax Academy email address](#) and [My Downloads Key](#). Depending on your environment, you might need to replace `@` in your email address with `%40` and escape any character in your password that is used in your operating system's command line. Examples: `\!` and `\|`.

For backward compatibility, you can use your DataStax Academy Profile Name and password instead of your email address and Downloads Key.

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

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

3. Install the DataStax agent:

```
$ sudo apt-get update
```

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

If you are using DSE 5.1 or earlier, you must install an earlier version of the DataStax Agent. See [DataStax OpsCenter compatibility with DSE](#).

```
$ sudo apt-get install datastax-agent datastax-agent-6.5.version-1
```

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

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

5. If SSL communication is enabled in `opscenterd.conf`, use SSL in `address.yaml` (page [address.yaml](#)):

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

6. Start the DataStax agent:

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

Manually deploying DataStax Agents from a tarball

Prerequisites:

- The DataStax Enterprise cluster is up and running.
- OpsCenter is installed and configured.
- JMX connectivity is enabled on each node in the cluster.
- SYSSTAT Utilities (needed for the collection of I/O metrics).

1. Download the DataStax agent tarball and extract it:

```
$ curl --user DSA_email_address:downloads_key\
-L http://downloads.datastax.com/enterprise/datastax-
agent-version_number.tar.gz | tar xz
```

where `DSA_email_address` and `downloads_key` are your [DataStax Academy email address](#) and [My Downloads Key](#). Depending on your environment, you might

need to replace @ in your email address with %40 and escape any character in your password that is used in your operating system's command line. Examples: \! and \|.

For backward compatibility, you can use your DataStax Academy Profile Name and password instead of your email address and Downloads Key.

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 might have to create the `address.yaml` file.

```
$ echo "stomp_interface: reachable_opscenterd_ip" >> ./conf/address.yaml
```

4. If SSL communication is enabled in `opscenterd.conf`, use SSL in `address.yaml` ([page](#)):

```
$ echo "use_ssl: 1" >> ./conf/address.yaml
```

5. Start the agent:

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

Use the `-f` flag to run in the foreground.

Setting permissions to run the DataStax Agent as the DSE user

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

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

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

Table 5: Directory and File Permissions

Feature functionality	Permissions required
General Agent functionality	Read permission to <code>cassandra.yaml</code>

Feature functionality	Permissions required
Configuring a cluster	Read/write permissions to configuration directories and files.
Backup and restore	<ul style="list-style-type: none"> Read/write permissions to configuration directories and files. Read/write permissions to Cassandra data directories. <p>Note: A <code>umask</code> (page 61) must also be set to accommodate group permissions for new tables and data.</p> <ul style="list-style-type: none"> If <code>commit log archiving</code> is enabled, the DataStax Enterprise process must also have permissions to run the Agent's archive script and write permissions to the configured backup directory.

Table 6: Directory and File Locations

Directory/File	Location
<code>cassandra.yaml</code>	See Configuration directories and files below.
Configuration directories and files	<ul style="list-style-type: none"> DataStax Enterprise Package installations: <code>/etc/dse</code> DataStax Enterprise Tarball installations: <code>install_location/conf</code>
Data directories	Default: <code>/var/lib/cassandra</code> Note: Location is user-configurable; set in <code>cassandra.yaml</code> .
Commit log archiving script	<ul style="list-style-type: none"> Agent package install: <code>/usr/share/datastax-agent/bin/archive_commitlog.sh</code> Agent tarball install: <code>install_location/bin/archive_commitlog.sh</code>

To set up the `umask`:

1. Open a terminal.
2. To give read/write permissions for new tables and data, edit the appropriate shell file for the DataStax Enterprise environment:
3. Add the command `umask 002` to the top of the file.

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

```
umask 002
```

Configuring JAVA_HOME for DataStax Agents

DataStax Agents do not find the environment variables of the currently logged in user by default. If Java is not in the machine's `PATH`, the Agent log shows an error on startup:

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

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

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