DataStax 6.7
Installation Guide
Latest DSE version
Latest 6.7 patch: 6.7.7

Updated: 2020-02-15Z
## Contents

### Chapter 1. Installing DSE 6.7

- Which install method should I use? ................................................................. 5
- Basic install from the tarball on a single node ........................................... 5
- Installing supporting software ................................................................. 8
  - Installing OpenJDK 8 on RHEL-based Systems ........................................... 8
  - Installing OpenJDK 8 on Debian-based Systems .......................................... 8
  - Installing Oracle JRE or JDK on RHEL-based Systems .................................. 9
  - Installing Oracle JRE or JDK on Debian or Ubuntu systems ......................... 10
- Installing Python 2.7 on older RHEL installs ............................................ 11
- Installing a cluster using Lifecycle Manager 6.7 ........................................ 13
- Installing from the Yum package ........................................................... 17
- Installing from the Debian package ...................................................... 20
- Install from the tarball on any Linux distribution ........................................ 23
- Installing patch releases ........................................................................ 28
  - Installing the latest patch release .......................................................... 28
  - Installing any patch release .................................................................. 29
- Installing on cloud providers ................................................................... 30
- Installing on Docker .................................................................................. 31
- Uninstalling DSE ....................................................................................... 31
- Default DSE file locations ......................................................................... 32
  - Package installations .............................................................................. 32
  - Tarball installations ................................................................................. 35

### Chapter 2. Installing DSE 6.7 Tools

- Installing CQLSH ..................................................................................... 38
- Installing DataStax Studio 6.7 ................................................................. 38
- Installing DSE Graph Loader .................................................................... 39
- Installing DataStax Bulk Loader 1.3.4 ..................................................... 40
- Installing DataStax Apache Kafka Connector 1.1.1 ................................... 40

### Chapter 3. Installing DSE OpsCenter 6.7

- Installing from the RPM package ............................................................. 41
- Installing from the Debian package ......................................................... 42
- Installing from the tarball on any Linux distribution ............................... 43
Installing on Docker.................................................................................................................................................. 45
Uninstalling OpsCenter........................................................................................................................................... 45
Installing DataStax Agents 6.7.................................................................................................................................. 46
  Installing DataStax Agents automatically........................................................................................................... 46
  Installing DataStax Agents manually.................................................................................................................. 49
Setting Agent permissions to run as the DSE user..................................................................................................... 52
Configuring JAVA_HOME....................................................................................................................................... 53
Chapter 1. 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, and the type of operating system. Be sure to install on a supported platform.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

<table>
<thead>
<tr>
<th>Installation method</th>
<th>About</th>
</tr>
</thead>
</table>
| **DSE OpsCenter Lifecycle Manager** | The Lifecycle Manager (LCM) can install DSE. It is fully integrated with DSE OpsCenter. LCM provides:  
  • 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.  
  Installing DSE using LCM requires installing DSE OpsCenter, bringing your own instances on a supported platform with SSH and Python installed, and root permissions on the target nodes. |
| **Installing DataStax Enterprise 6.7 on RHEL-based systems using Yum** | 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** | Installs DSE on Debian-based systems using APT. Requires root permissions. Typically used in production environments. |
| **Binary tarball installer** | Installs DSE on any supported Linux-based platform. Does not require root permissions. |
| **Installing and deploying DSE on cloud platforms** | Instructions for installing and deploying DSE on Google Compute Engine, Microsoft Azure, and Amazon EC2. |
| **Installing DSE 6.7 on Docker** | Instructions for using DataStax Docker images to create DSE server, DSE OpsCenter, and DataStax Studio containers in non-production environments. |

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 and installation information, see the DataStax drivers page.

Installing basic DataStax Enterprise 6.7 on a single node

Use these instructions to install a single instance of DataStax Enterprise (DSE) on Linux-based platforms using a binary tarball. This single node can introduce you to DSE, which includes DSE Security, DSE Graph, DSE Analytics, and DSE Search.
Installing DataStax Enterprise 6.7

This installation is intended only for demonstration purposes. For other installations, see Which install method should I use?

Some things to know about installing DSE:

• The latest version of DataStax Enterprise 6.7 is 6.7.7.
• When installed from the binary tarball, DataStax Enterprise runs as a stand-alone process.
• 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:
Java 8. If you don’t have a recent version of Java 8, download and install OpenJDK 8.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

In a terminal window:

1. Verify that a required version of 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-b17)
   Java HotSpot(TM) 64-Bit Server VM (build 25.181-b01, mixed mode)

   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 not Oracle Java 8, or OpenJDK 8, see Installing supporting software on DataStax Enterprise 6.7.

   Package management tools do not install Oracle Java.

2. Install the libaio package. For example:

   • RHEL platforms:

     $ sudo yum install libaio

   • Debian platforms:

     $ sudo apt-get install libaio1

3. Use curl to download and install DSE:
Installing DataStax Enterprise 6.7

Your password is retained in the shell history. To avoid this security issue, DataStax recommends using curl with the --netrc or --netrc-file option.

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

The files are downloaded and extracted into the 6.7 directory.

4. Create the default data and logging directories:

   - /var/lib/cassandra
   - /var/log/cassandra

```
$ 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
```

5. Start DataStax Enterprise from the installation directory:

```
$ bin/dse cassandra
```

6. Verify that DataStax Enterprise is running from the installation directory:

```
$ bin/nodetool status
```

Results:

```
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
```

7. Start cqlsh from the installation directory:

```
$ bin/cqlsh
```

For more, see Accessing data using CQL.

What’s next:
See Getting started with DataStax Enterprise 6.7.
For help:
Installing DataStax Enterprise 6.7

Installing supporting software on DataStax Enterprise 6.7

Installing OpenJDK 8 on RHEL-based Systems

Configure your operating system to use the latest build of a Technology Compatibility Kit (TCK) Certified OpenJDK version 8. For example, OpenJDK 8 (1.8.0_151 minimum).

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

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 latest build of a Technology Compatibility Kit (TCK) Certified OpenJDK version 8. For example, OpenJDK 8 (1.8.0_151 minimum).

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

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.

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.

   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 java
```

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**

Configure your operating system to use the latest build of a Technology Compatibility Kit (TCK) Certified OpenJDK version 8. For example, OpenJDK 8 (1.8.0.151 minimum).

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.

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.
Installing DataStax Enterprise 6.7

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:

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

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

   ```bash
   $ 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/jdk8u_version`.

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

   ```bash
   $ 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 many 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:

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

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

   ```bash
   $ 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.

DataStax Enterprise 6.7 does not support all older RHEL-based platforms. See Supported platforms.

1. Verify your Python version:

   ```bash
   $ python -V
   ``

2. If Python is not version 2.7.x, install it from the operating system software collection:
• For RHEL **Software Collections**: 

```bash
$ 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**: 

```bash
$ 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**: 

```bash
$ python -V
```

**Python 2.7.8**

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

```bash
$ 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.
- All DSE nodes are accessible by OpsCenter.
- 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.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

1. **Install and start OpsCenter** on a separate machine using the installation method of your choice.

2. In a web browser, go to [http://OpsCenter host:8888](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, dse67.
   
   b. Select the version of DataStax Enterprise under DataStax Enterprise Version. For example, dse v6.7.0.

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.
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 + 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.
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 every datacenter.

15. Select the cluster name under Clusters, then click the drop menu to the right of the cluster name.

16. Click Install to open the Run Installation Job dialog.
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.

Some things to know about installing DSE:

- The latest version of DataStax Enterprise 6.7 is 6.7.7.
- 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, 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.

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.
- Yum Package Management application.
Installing DataStax Enterprise 6.7

- Configure your operating system to use the latest version of Java 8:
  - Recommended. The latest build of a TCK (Technology Compatibility Kit) Certified OpenJDK version 8. For example, OpenJDK 8 (1.8.0_151 minimum).
  - 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.

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.

Hardware requirements

See Recommended production settings.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

In a terminal window:

1. Verify that a required version of Java is installed:
   
   ```sh
   $ java -version
   ```

   DataStax recommends the latest build of a Technology Compatibility Kit (TCK) Certified OpenJDK version 8.
   
   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.

2. Install the libaio package. For example:

   ```sh
   $ sudo yum install libaio
   ```

3. Add the DataStax Yum repository to a file called /etc/yum.repos.d/datastax.repo:
Set the `gpgcheck=1` to perform a GPG signature check.

```
[dastax]
name=DataStax Repo for DataStax Enterprise
dir=https://rpm.datastax.com/enterprise/
enabled=1
gpgcheck=0
```

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 (6.7.7):
     ```
     $ sudo yum install dse-full
     ```
   - Install an earlier 6.7.x version:
     To view the available 6.7.x versions, see the Release notes.
     ```
     $ sudo yum install dse-full-version_number-1
     ```
     For example:
     ```
     $ sudo yum install dse-full-6.7.0-1
     ```
   - Optional: Install the demos:
     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.

6. Single-node cluster installations only:
   - Start DataStax Enterprise:
     ```
     $ sudo service dse start
     ```
   - Verify that DataStax Enterprise is running:
     ```
     $ nodetool status
     ```

   ```
   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
   ```

What’s next:
Installing DataStax Enterprise 6.7

• 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.
• Configure DSE service startup options. See Starting DataStax Enterprise as a service.
• Default file locations for package installations
• Default file locations for tarball installations
• 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.

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 6.7.7.
• 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, 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.

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.
• APTitude Package Management (APT) application.
• Configure your operating system to use the latest version of Java 8:

  # Recommended. The latest build of a TCK (Technology Compatibility Kit) Certified OpenJDK version 8. For example, OpenJDK 8 (1.8.0_151 minimum).

# Supported. Oracle Java SE 8 (JRE or JDK) (1.8.0_151 minimum)

- Python 2.7.x

## Hardware requirements

See Recommended production settings.

**End User License Agreement (EULA).** By downloading this DataStax product, you agree to the terms of the EULA.

In a terminal window:

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

   ```
   $ java -version
   ``

   DataStax recommends the latest build of a Technology Compatibility Kit (TCK) Certified OpenJDK version 8.

   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.
1. Install the libaio package. For example:

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

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

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

3. Add the DataStax repository key:

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

4. Update apt-get:

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

5. Install the DataStax Enterprise package:

- Install the latest version (6.7.7):
  
  ```
  $ sudo apt-get install dse-full
  ```

- Install an earlier 6.7.x version:
  
  To view the available 6.7.x versions, see the Release notes.

  ```
  $ sudo apt-get install dse=version_number-1 dse-full=version_number-1 dse-libcassandra=version_number-1 dse-libgraph=version_number-1 dse-libhadoop2-client-native=version_number-1 dse-libhadoop2-client=version_number-1 dse-liblog4j=version_number-1 dse-libsolr=version_number-1 dse-libspark=version_number-1 dse-libtomcat=version_number-1
  ```

  For example:

  ```
  $ sudo apt-get install dse=6.7.0-1 dse-full=6.7.0-1 dse-libcassandra=6.7.0-1 dse-libgraph=6.7.0-1 dse-libhadoop2-client-native=6.7.0-1 dse-libhadoop2-client=6.7.0-1 dse-liblog4j=6.7.0-1 dse-libsolr=6.7.0-1 dse-libspark=6.7.0-1 dse-libtomcat=6.7.0-1
  ```

  **Optional:** Install the demos:

  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.

6. **Single-node cluster installations only:**

```
Installing DataStax Enterprise 6.7 using the binary tarball

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

Some things to know about installing DSE:

- The latest version of DataStax Enterprise 6.7 is 6.7.7.
- When installed from the binary tarball, DataStax Enterprise runs as a stand-alone process.
- 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.
Installing DataStax Enterprise 6.7

Prerequisites:

• A supported platform.

• Configure your operating system to use the latest version of Java 8:

  # Recommended. The latest build of a TCK (Technology Compatibility Kit) Certified OpenJDK version 8.
  
  For example, OpenJDK 8 (1.8.0_151 minimum).


  # 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.

Hardware requirements

See Recommended production settings.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

In a terminal window:

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

   $ java -version

   DataStax recommends the latest build of a Technology Compatibility Kit (TCK) Certified OpenJDK version 8.

   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.

2. Install the libaio package. For example:
• RHEL platforms:

    $ sudo yum install libaio

• Debian platforms:

    $ sudo apt-get install libaio1

Installing the latest version (6.7.7)

To download a specific version of DSE 6.7.x, go to 4.

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 (6.7.7):

     a. Download the tarball from Download DataStax Enterprise.

     b. Extract the files:

        $ tar -xzvf dse-6.7.7-bin.tar.gz

   • Use curl to install the latest version (6.7.7):

     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 -L https://downloads.datastax.com/enterprise/dse.tar.gz | tar xz

     The files are downloaded and extracted into the 6.7 directory.

Installing specific 6.7.x versions

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

   • Download and extract specific 6.7.x tarballs into the current directory:

     a. Download the tarball from Download DataStax Enterprise.

     b. Extract the files:

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

   • Use curl to install specific 6.7.x versions:

     a. Download and extract:

        $ curl -L https://downloads.datastax.com/enterprise/dse-version_number-bin.tar.gz | tar xz

     The files are downloaded and extracted into the 6.7 directory.

5. You can use either the default data and logging directory locations or define your locations:
Installing DataStax Enterprise 6.7

- **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
  $$ 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
  ```

- **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 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. Update the following lines in the `cassandra.yaml` file to match the custom locations:

  ```
  data_file_directories:
  - full_path_to_installation_location/dse-data/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
  ```

6. If using DSE analytics, 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/log/spark/rrdd; sudo chown -R $USER:$GROUP /var/lib/spark/rrdd &&
  sudo mkdir -p /var/log/spark/master; sudo chown -R $USER:$GROUP /var/log/spark/master &&
  sudo mkdir -p /var/log/spark/alwaysson_sql; sudo chown -R $USER:$GROUP /var/log/spark/
  alwaysson_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 && mkdir spark && cd spark && mkdir log && mkdir rdd && mkdir worker && cd log && mkdir worker && mkdir master && mkdir 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 ALWAYSON_SQL_LOG_DIR="full_path_to_installation_location/spark/log/alwayson_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.

7. 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

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
```

```
Datacenter: Cassandra
---------------------
Status=Up/Down
| State=Normal/Leaving/Joining/Moving
-- Address    Load    Tokens    Owns    Host ID
Rack
```
Results not using vnodes:

<table>
<thead>
<tr>
<th>Datacenter: Analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status=Up/Down</td>
</tr>
<tr>
<td><img src="image" alt="State-Normal/Leaving/Joining/Moving" /></td>
</tr>
<tr>
<td><img src="image" alt="Address" /></td>
</tr>
<tr>
<td><img src="image" alt="Token" /></td>
</tr>
</tbody>
</table>

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.
- Configure DSE service startup options. See Starting DataStax Enterprise as a service.
- Default file locations for package installations
- Default file locations for tarball installations
- 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.

Installing DataStax Enterprise 6.7.x patch releases

The latest version of DataStax Enterprise is 6.7.7.

To view the available patch releases for DataStax Enterprise 6.7.x, see the 6.7 Release Notes.

To install patches for earlier versions of DataStax Enterprise, see 6.0 or 5.1

Installing the latest DataStax Enterprise 6.7.x patch release

The latest version of DataStax Enterprise is 6.7.7.

To view the available patch releases for DataStax Enterprise 6.7, see the 6.7 Release Notes.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.
Installing DataStax Enterprise 6.7

Installing the latest 6.7.x patch release (6.7.7) from the RHEL packages

1. If needed, add the DataStax Yum repository and enable signature verification and enable signature verification.

2. Install DataStax Enterprise package.
   For example:
   
   ```
   $ sudo yum install dse-full
   ```

Installing the latest 6.7.x patch release (6.7.7) from the Debian packages

1. If needed, add the DataStax repository file and key.

2. Install the latest version (6.7.7):
   
   ```
   $ sudo apt-get install dse-full
   ```

Installing the latest 6.7.x patch release (6.7.7) from the binary tarball

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 DSE patch release (6.7.7):
  
  1. Download the tarball from Download DataStax Enterprise into the current directory.
  2. Extract the files:
     
     ```
     $ tar -xzvf dse-6.7.7-bin.tar.gz
     ```

- Use curl to install the latest DSE patch release (6.7.7):
  
  Use the following curl command to download and install the latest DSE patch release (6.7.7):
  
  ```
  $ curl -L https://downloads.datastax.com/enterprise/dse.tar.gz | tar xz
  ```

Installing Previous DataStax Enterprise 6.7.x patch releases

The latest version of DataStax Enterprise is 6.7.7.

To view the available patch releases for DataStax Enterprise 6.7.x, see the 6.7 Release Notes.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

Installing previous 6.7.x patch releases from the RHEL packages

1. If needed, add the DataStax Yum repository and enable signature verification.

2. Install DataStax Enterprise specifying the version.
   
   ```
   $ sudo yum install dse-full-version_number-1
   ```
   
   For example:
   
   ```
   $ sudo yum install dse-full-6.7.0-1
   ```

Installing previous 6.7.x patch releases from the Debian packages

1. If needed, add the DataStax repository file and key.
2. Install DataStax Enterprise specifying the version. In addition to specifying the DSE version, you must also specify the exact dependent packages, including the package version.

```bash
$ sudo apt-get install dse=version_number-1 dse-full=version_number-1 dse-libcassandra=version_number-1 dse-libgraph=version_number-1 dse-libhadoop2-client-native=version_number-1 dse-libhadoop2-client=version_number-1 dse-liblog4j=version_number-1 dse-libsolr=version_number-1 dse-libspark=version_number-1 dse-libtomcat=version_number-1
```

For example:

Specify all packages; otherwise, the installation fails.

```bash
$ sudo apt-get install dse=6.7.0-1 dse-full=6.7.0-1 dse-libcassandra=6.7.0-1 dse-libgraph=6.7.0-1 dse-libhadoop2-client-native=6.7.0-1 dse-libhadoop2-client=6.7.0-1 dse-liblog4j=6.7.0-1 dse-libsolr=6.7.0-1 dse-libspark=6.7.0-1 dse-libtomcat=6.7.0-1
```

### Installing previous 6.7.x patch releases from the binary tarball

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

- **Download and extract specific 6.7.x tarballs:**
  1. Download the tarball from [DataStax Enterprise directory](https://www.datastax.com) into the current directory.
  2. Extract the files:

```bash
$ tar -xzvf dse-version_number-bin.tar.gz
```

For example:

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

- **Use curl to install specific 6.7.x versions:**
  Download and extract:

```bash
$ curl -L https://downloads.datastax.com/enterprise/dse-version_number-bin.tar.gz | tar xz
```

The files are downloaded and extracted into the 6.7 directory.

### Installing and deploying DataStax Enterprise 6.7 on cloud platforms

#### Installing a DataStax Enterprise cluster on various cloud providers

**End User License Agreement (EULA).** By downloading this DataStax product, you agree to the terms of the EULA.

The [DataStax Partner Network](https://www.datastax.com/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](https://aws.amazon.com/datastax-enterprise/) and [Installing a DataStax Enterprise cluster on Amazon EC2](https://aws.amazon.com/datastax-enterprise/ec2/) 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 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.

Get the full installation instructions in the **DataStax Docker guide**.

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 Docker docs** for information on downloading and using the DataStax images for Docker.

**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 or Yum.

1. Drain and stop the DataStax Enterprise service:

   ```
   nodetool drain &
   ```
Installing DataStax Enterprise 6.7

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 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.

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 DataStax Enterprise 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**

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/dse/cassandra/cassandra.yaml</td>
<td>cassandra.yaml is the main configuration file for the DataStax Enterprise database with default configuration for all nodes.</td>
</tr>
<tr>
<td>/etc/dse/dse.yaml</td>
<td>dse.yaml is the main configuration file for DataStax Enterprise.</td>
</tr>
</tbody>
</table>

**Default database directories**

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/lib/cassandra/data</td>
<td>commitlog, data, hints, saved_caches directories</td>
</tr>
</tbody>
</table>
### Directories

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/var/log/cassandra</code></td>
<td>Log files, including:</td>
</tr>
<tr>
<td></td>
<td>• audit directory</td>
</tr>
<tr>
<td></td>
<td>• debug.log</td>
</tr>
<tr>
<td></td>
<td>• gremlin.log</td>
</tr>
<tr>
<td></td>
<td>• solrvalidation.log</td>
</tr>
<tr>
<td></td>
<td>• system.log</td>
</tr>
<tr>
<td></td>
<td>You can change logging locations.</td>
</tr>
<tr>
<td><code>/var/run/cassandra</code></td>
<td>Database process ID (pid) directory</td>
</tr>
<tr>
<td><code>/usr/share/dse/cassandra</code></td>
<td>Environment settings</td>
</tr>
<tr>
<td><code>/usr/share/dse/cassandra/tools</code></td>
<td>Tools for testing, starting, using SSTables, plus YAML examples.</td>
</tr>
<tr>
<td><code>/etc/dse/cassandra</code></td>
<td>Property files and <code>cqlshrc</code> samples including:</td>
</tr>
<tr>
<td></td>
<td>• <code>cassandra-env.sh</code></td>
</tr>
<tr>
<td></td>
<td>• <code>cassandra-rackdc.properties</code></td>
</tr>
<tr>
<td></td>
<td>• <code>cassandra-topology.properties</code></td>
</tr>
<tr>
<td></td>
<td>• <code>cassandra-topology.yaml</code></td>
</tr>
<tr>
<td></td>
<td>• <code>commitlog_archiving.properties</code></td>
</tr>
<tr>
<td></td>
<td>• <code>cqlshrc.sample</code></td>
</tr>
<tr>
<td></td>
<td>• <code>logback.xml</code></td>
</tr>
<tr>
<td><code>/etc/init.d</code></td>
<td>Set node type and other server configuration</td>
</tr>
</tbody>
</table>

#### Default DSEFS data directory

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

#### Default DSE Graph directories

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/dse/graph/gremlin-console/conf/remote.yaml</code></td>
<td>Gremlin console configuration for connection to the Gremlin Server, including Kerberos authentication and SSL encryption.</td>
</tr>
<tr>
<td><code>/etc/dse/graph/logback-gremlin-server.xml</code></td>
<td>GremlinServerFileAppender</td>
</tr>
</tbody>
</table>

#### Default DSE Search directories

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

#### Default Spark directories

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/dse/spark/</code></td>
<td><code>spark-env.sh</code>, <code>spark-defaults.conf</code>, <code>spark-daemon-defaults.conf</code></td>
</tr>
<tr>
<td><code>/usr/share/dse/spark/lib</code></td>
<td>Spark library</td>
</tr>
</tbody>
</table>
## Installing DataStax Enterprise 6.7

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/log/spark</td>
<td>Spark Master and Worker logs</td>
</tr>
<tr>
<td>/usr/share/dse/spark/spark-jobserver</td>
<td>Spark Jobserver</td>
</tr>
<tr>
<td>/usr/share/dse/demos/portfolio_manager</td>
<td>Spark Portfolio Manager demo</td>
</tr>
<tr>
<td>/var/lib/dsefs</td>
<td>The default directory to store the DSE File System data.</td>
</tr>
</tbody>
</table>

### Default location for the logback configuration file

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/dse/cassandra/logback.xml</td>
<td>logback.xml is the logback configuration file</td>
</tr>
</tbody>
</table>

### Default location audit logs

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf/web.xml</td>
<td></td>
</tr>
<tr>
<td>/var/log/cassandra/dropped_audit_events.log</td>
<td>Default location for dropped events logs.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/dse</td>
<td>/etc/dse/dse.yaml is the primary configuration file for DataStax Enterprise</td>
</tr>
<tr>
<td>/etc/dse-node1</td>
<td>/etc/dse-node1/dse.yaml is the configuration file for the DataStax Enterprise node in the dse-node1 directory</td>
</tr>
<tr>
<td>/etc/dse-node2</td>
<td>/etc/dse-node2/dse.yaml is the configuration file for the DataStax Enterprise node in the dse-node2 directory</td>
</tr>
</tbody>
</table>

For a comprehensive list of file locations in a DSE Multi-Instance cluster, see directories for DSE Multi-Instance.

### Default DSE Multi-Instance generated directories

With DSE Multi-Instance, these directories are created on the host machine for each node.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSE Multi-Instance root directory</td>
<td>/etc/defaults</td>
<td>Each DSE Multi-Instance host machine has the /etc/defaults root directory. This default location is not configurable.</td>
</tr>
<tr>
<td>DataStax Enterprise node type</td>
<td>/etc/defaults/dse-nodeId</td>
<td>Defines the node type (transactional, search, analytics, graph, and so on).</td>
</tr>
<tr>
<td>DataStax Enterprise configuration file</td>
<td>/etc/dse-nodeId/dse.yaml</td>
<td>A configuration file for each node.</td>
</tr>
<tr>
<td>DataStax Enterprise process ID (pid) directory</td>
<td>/var/run/dse-nodeId.dse-nodeId.pid</td>
<td>The default DataStax Enterprise process ID (pid) directory for each node.</td>
</tr>
<tr>
<td>Purpose</td>
<td>Directories</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Database configuration</td>
<td>/etc/dse-nodeId/cassandra/</td>
<td>A configuration file for each node.</td>
</tr>
<tr>
<td></td>
<td>cassandra.yaml</td>
<td></td>
</tr>
<tr>
<td>Database data directory</td>
<td>/var/lib/dse-nodeId/data</td>
<td>The root directory for storing data on each node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define with <code>dse add-node --data-directory=directory...</code></td>
</tr>
<tr>
<td>Database log files</td>
<td>/var/log/dse-nodeId/cassandra</td>
<td>The default directory where the audit.log, output.log, solrvalidation.log, and system.log log files are stored for each node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define with <code>dse add-node --logs-directory=directory...</code></td>
</tr>
<tr>
<td>Database pid directory</td>
<td>/var/run/dse-nodeId</td>
<td>Database process ID (pid) directory for each node.</td>
</tr>
<tr>
<td>Caches directory</td>
<td>/var/lib/dse-nodeId/saved_caches</td>
<td>The table key and row caches directory for each node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define with <code>dse add-node --saved-caches-directory=directory...</code></td>
</tr>
<tr>
<td>Commit log files</td>
<td>/var/lib/dse-nodeId/commitlog</td>
<td>The commit log directory for each node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define with <code>dse add-node --commit-directory=directory...</code></td>
</tr>
<tr>
<td>Hints directory</td>
<td>/var/lib/dse-nodeId/hints</td>
<td>The hints directory for each node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define with <code>dse add-node --hints-directory=directory...</code></td>
</tr>
<tr>
<td>Spark configuration file</td>
<td>/etc/dse-nodeId/spark/spark-env.sh</td>
<td>Spark configuration file <code>spark-env.sh</code> for each node.</td>
</tr>
<tr>
<td>Spark Worker data directory</td>
<td>/var/lib/dse-nodeId/spark/worker</td>
<td>The data directory for Spark Worker for each node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define with <code>dse add-node --spark-worker-directory=directory...</code></td>
</tr>
<tr>
<td>Spark Worker local node directory</td>
<td>/var/lib/dse-nodeId/spark/rdd</td>
<td>The local directory for Spark Worker for each node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define with <code>dse add-node --spark-local-directory=directory...</code></td>
</tr>
<tr>
<td>Spark logs directory</td>
<td>/var/log/dse-nodeId/spark</td>
<td>The Spark logs directory for each node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define with <code>dse add-node --spark-log-directory=directory...</code></td>
</tr>
<tr>
<td>Logback configuration</td>
<td>/etc/dse-nodeId/cassandra/logback.xml</td>
<td>Logback configuration file for each node.</td>
</tr>
<tr>
<td>Solr configuration</td>
<td>/etc/dse-nodeId/solr</td>
<td>Solr configuration files for each node.</td>
</tr>
<tr>
<td>Tomcat log files</td>
<td>/var/log/dse-nodeId/tomcat</td>
<td>The directory for Tomcat server logs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define with <code>dse add-node --tomcat-logs=directory...</code></td>
</tr>
</tbody>
</table>

**Licenses and other documents**

The default location is `installation_location`. Also see [DataStax Enterprise third-party software](https://www.datastax.com/products/datastax-enterprise-support).  

**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**

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>installation_location/resources/cassandra/conf/cassandra.yaml</code></td>
<td><code>cassandra.yaml</code> is the main configuration file for the database.</td>
</tr>
<tr>
<td><code>installation_location/resources/cassandra/conf/cassandra.yaml/installation_location/resources/dse/conf/dse.yaml</code></td>
<td><code>dse.yaml</code> is the main configuration file for DataStax Enterprise.</td>
</tr>
</tbody>
</table>
### Default database directories

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>installation_location/resources/cassandra/bin</code></td>
<td>Commands and utilities, such as nodetool, cqlsh, sstabledump, and sstableloader</td>
</tr>
</tbody>
</table>
| `installation_location/resources/cassandra/conf/cassandra.yaml` | Property files and cqlshrc samples including:  
  - cassandra-env.sh  
  - cassandra-rackdc.properties  
  - cassandra-topology.properties  
  - cassandra-topology.yaml  
  - commitlog_archiving.properties  
  - cqlshrc.sample  
  - logback.xml |
| `/var/lib/cassandra` or `installation_location` | Commitlog, data, hints, saved_caches directories |
| `/var/log/cassandra` | Log files, including:  
  - audit directory  
  - debug.log  
  - gremlin.log  
  - solrvalidation.log  
  - system.log  
  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

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

### Default DSE Search directories

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

### Default Spark directories

<table>
<thead>
<tr>
<th>Directories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>installation_location/resources/spark/conf</code></td>
<td><code>spark-env.sh</code>, <code>spark-defaults.conf</code>, <code>spark-daemon-defaults.conf</code></td>
</tr>
<tr>
<td><code>/var/lib/spark</code></td>
<td>Spark library</td>
</tr>
</tbody>
</table>
### Directories

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/log/spark</td>
<td>Spark Master and Worker logs</td>
</tr>
<tr>
<td>installation_location/resources/spark/spark-jobserver</td>
<td>Spark Jobserver</td>
</tr>
<tr>
<td>installation_location/demos/portfolio_manager</td>
<td>Spark Portfolio Manager demo</td>
</tr>
<tr>
<td>/var/lib/dsefs</td>
<td>The default directory to store the DSE File System data.</td>
</tr>
</tbody>
</table>

### Default Logback-appender directories

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>installation_location/resources/cassandra/conf/logback.xml</td>
<td>logback.xml is the logback configuration file</td>
</tr>
</tbody>
</table>

### Default location audit logs

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/dse/tomcat/conf/server.xml</td>
<td>Default location for Tomcat server logs for DSE Search.</td>
</tr>
<tr>
<td>/var/log/cassandra/dropped_audit_events.log</td>
<td>Default location for dropped events logs.</td>
</tr>
</tbody>
</table>

### Default DSE OpsCenter directories

See the [OpsCenter documentation](#).

### Default directory for Token-generator tool

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>installation_location/resources/cassandra/tools/bin/token-generator</td>
<td>For manually generating tokens.</td>
</tr>
</tbody>
</table>

### Licenses and other documents

The default location is `installation_location`. Also see [DataStax Enterprise third-party software](#).
Chapter 2. Installing DataStax Enterprise 6.7 Tools

Installing the CQLSH tool

Install the CQLSH standalone tool using a binary tarball on any Linux-based platform. This command line shell remotely connects to a DataStax database cluster. The tool is not required to run locally on a cluster node.

Prerequisites: The cqlsh Python script that starts CQLSH requires Python 2.7.

DataStax CQLSH License Terms. By downloading this DataStax product, you agree to the terms of the EULA.

1. On the DataStax Download page, click Tools > CQLSH.
2. If you agree with the license terms, select the CQLSH Terms checkbox, choose a compatible version, and click Tarball to download the tool.
3. Unpack the distribution:

   $ tar -xzvf cqlsh-6.7.tar.gz

   The files are downloaded and extracted into the current directory.

What’s next: Start CQLSH.

Installing DataStax Studio 6.7

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 for your platform.
- 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 Troubleshooting DataStax Studio.

- Windows platforms: Windows 7 and 10 with Java 8.
  - Ensure Microsoft Windows 7 libraries are current.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

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

   ```plaintext
   user_home_directory/.datastax_studio
   ```

   See User data in DataStax Studio.

2. Download the Studio tarball from the DataStax Download page.

3. Unpack Studio using the appropriate method for the file type:

   - Linux:
     ```bash
     tar zvxf 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.

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

- DataStax Enterprise is installed and configured for DSE Graph.
Installing DataStax Enterprise 6.7 Tools

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

1. Download the DSE Graph Loader tarball from the DataStax Download page.

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 1.3.4

See Installing DataStax Bulk Loader.

Installing DataStax Apache Kafka Connector 1.1.1

See Installing DataStax Apache Kafka Connector.
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.
- Latest build of a TCK (Technology Compatibility Kit) Certified OpenJDK version 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**.
  
  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.

  **End User License Agreement (EULA).** By downloading this DataStax product, you agree to the terms of the EULA.

In a terminal window:

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

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

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

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

   $ sudo yum install -y opscenter-6.7.version_number

   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.
- Latest build of a TCK (Technology Compatibility Kit) Certified OpenJDK version 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.

  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.

**End User License Agreement (EULA).** By downloading this DataStax product, you agree to the terms of the EULA.

In a terminal window:

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

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

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

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

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

   ```bash
   $ sudo apt-get update
   $ sudo apt-get install opscenter
   ```
   
   If you are installing an earlier version of OpsCenter, add the supported version to the command:
   ```bash
   $ sudo apt-get install -y opscenter=6.7.version_number
   ```

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

4. Start OpsCenter:

   ```bash
   $ sudo service opscenterd start
   ```

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

   ```bash
   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.

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
Installing DSE OpsCenter 6.7

- 2 GB of RAM available to OpsCenter

Permission and software requirements:

- Latest build of a TCK (Technology Compatibility Kit) Certified OpenJDK version 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.

  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.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

Installing the latest version (6.7.7)

To download a specific version of OpsCenter 6.7.x, go to 2.

1. You can either download the tarball and then extract the files, or use curl.
   - Download and extract the latest version tarball (6.7.7):
     a. Download the tarball from Download DataStax OpsCenter.
     b. Extract the files:

        $ tar -xzvf opscenter-6.7.7.tar.gz

        The files are extracted into the opscenter-6.7.7 directory.

   - Use curl to install the latest version (6.7.7):
     a. In a terminal window, download and extract the tarball using curl:

        $ curl -L https://downloads.datastax.com/enterprise/opscenter.tar.gz | tar xz

        The files are downloaded and extracted into the opscenter-6.7.7 directory.

Installing specific 6.7.x versions

2. When installing from the binary tarball, you can either download the tarball and then extract the files, or use curl.
   - Download and extract specific 6.7.x tarballs into the current directory:
     a. Download the tarball from Download DataStax Enterprise.
     b. Extract the files:

        $ tar -xzvf opscenter-6.7.version_number.tar.gz

     - Use curl to install specific 6.7.x versions:
Installing DSE OpsCenter 6.7

a. Download and extract:

```bash
$ curl -L https://downloads.datastax.com/enterprise/opscenter-6.7.version_number.tar.gz | tar xz
```

The files are downloaded and extracted into the opscenter-6.7.x directory.

3. Change to the `opscenter-version_number` directory:

```bash
$ cd opscenter-version_number
```

4. Start OpsCenter from the install location:

```bash
$ bin/opscenter
```

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/
```

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 Docker docs](#) for information on downloading and using the DataStax images for Docker.

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 package.

1. **Stop** OpsCenter.

2. Open a terminal and enter the following command:

```bash
$ sudo yum remove opscenter
```

Uninstalling an OpsCenter Debian package
Use this method if OpsCenter was installed using a Debian package.

1. **Stop** OpsCenter.
Uninstalling an OpsCenter binary tarball
Use this method if OpsCenter was installed using a tarball.

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

   $ sudo apt-get purge 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 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.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

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 DataStax Agents.
7. Enter SSH credentials to connect to the nodes:
   a. Enter a **Username**. The user must have root or sudo privileges.
   b. If using a sudo password for the specified user, enter the password in the **Password** field.
   c. If using an SSH private key, enter it in the **Private Key** field.

   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.
Installing DSE OpsCenter 6.7

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.

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.

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

In a terminal:

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

```bash
[opscenter]
name = DataStax Repository
baseurl = https://rpm.datastax.com/enterprise
```
Installing DSE OpsCenter 6.7

```
enabled = 1
gpgcheck = 0
```

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

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-6.7.version-1
   ```

4. In DataStax Agent configuration, 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.

**End User License Agreement (EULA).** By downloading this DataStax product, you agree to the terms of the EULA.

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

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

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

```
$ 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-6.7.version-1
```

4. In DataStax Agent configuration, 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 DataStax Agent configuration:

```
$ 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).

End User License Agreement (EULA). By downloading this DataStax product, you agree to the terms of the EULA.

In a terminal:
1. Download the DataStax agent tarball and extract it:

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

2. Change into the agent directory:

   ```
   $ cd datastax-agent-version_number
   ```

3. In `address.yaml` 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`:

   ```
   $ 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.

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 1: Directory and File Permissions

<table>
<thead>
<tr>
<th>Feature functionality</th>
<th>Permissions required</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Agent functionality</td>
<td>Read permission to <code>cassandra.yaml</code></td>
</tr>
<tr>
<td>Configuring a cluster</td>
<td>Read/write permissions to configuration directories and files.</td>
</tr>
<tr>
<td>Backup and restore</td>
<td>• Read/write permissions to configuration directories and files.</td>
</tr>
<tr>
<td></td>
<td>• Read/write permissions to Cassandra data directories.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
</tbody>
</table>

### Table 2: Directory and File Locations

<table>
<thead>
<tr>
<th>Directory/File</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cassandra.yaml</code></td>
<td>See Configuration directories and files below.</td>
</tr>
<tr>
<td>Configuration directories and files</td>
<td>• DataStax Enterprise Package installations: /etc/dse</td>
</tr>
<tr>
<td></td>
<td>• DataStax Enterprise Tarball installations: install_location/conf</td>
</tr>
<tr>
<td>Data directories</td>
<td>Default: /var/lib/cassandra</td>
</tr>
<tr>
<td></td>
<td>Location is user-configurable; set in <code>cassandra.yaml</code>.</td>
</tr>
</tbody>
</table>
To set up the umask:

1. Open a terminal.

2. To give read/write permissions for new tables and data, edit the appropriate shell file for the DataStax Enterprise environment:

<table>
<thead>
<tr>
<th>File</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>dse-env.sh</td>
<td>/etc/dse/</td>
</tr>
<tr>
<td></td>
<td>install_location/conf/</td>
</tr>
<tr>
<td>cassandra-env.sh</td>
<td>/etc/dse/cassandra</td>
</tr>
<tr>
<td></td>
<td>install_location/conf</td>
</tr>
</tbody>
</table>

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

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

   ```bash
   umask 002
   ```

**Configuring JAVA_HOME for DataStax Agents**

DataStax Agents do not 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:

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