The first step is to download the software you’ll need for your Windows machine.

DataStax, makes available the [DataStax Community Edition](http://www.PlanetCassandra.org/cassandra), which contains the latest community version of Apache Cassandra, along with the Cassandra Query Language (CQL) utility, and a free edition of DataStax OpsCenter, which is the tool you’ll want to use for managing and monitoring your Cassandra cluster on Windows.

To get Datastax Community Edition, go to the [downloads](http://www.PlanetCassandra.org/cassandra) page and select the Windows installation package for your version of Microsoft Windows. Note that 32 and 64-bit installers are offered.

### DataStax Community Edition for Apache Cassandra™ (v2.1.x – v3.0.x) Consists of Several Components:

• An “Archive Release” of Apache Cassandra™  
• DataStax OpsCenter Monitoring Tool (Does not work with Apache Cassandra™ v3.0 and beyond; [click here](http://www.planetcassandra.org/opscenter) to download & setup OpsCenter for other operating systems and previous versions)  
• Sample application and demo database  
• Smart installers for Linux, Windows, and Macintosh  
• Easy [uninstall](http://docs.datastax.com/en/cassandra/2.1/cassandra/install/installRemove.html) of DataStax Community Edition

**Download DataStax Community Edition v3.0.9 (**[**Changelog**](https://git1-us-west.apache.org/repos/asf?p=cassandra.git;a=blob_plain;f=CHANGES.txt;hb=refs/tags/cassandra-3.0.9)**)**  
[Tarball](http://downloads.datastax.com/community/dsc-cassandra-3.0.9-bin.tar.gz), [MSI Installer (32-bit)](http://downloads.datastax.com/community/datastax-community-32bit_3.0.9.msi), [MSI Installer (64-bit)](http://downloads.datastax.com/community/datastax-community-64bit_3.0.9.msi), [RPM Using Yum](http://docs.datastax.com/en/cassandra/3.0/cassandra/install/installRHEL.html), [DEB Using Apt-Get](http://docs.datastax.com/en/cassandra/3.0/cassandra/install/installDeb.html)

**Download DataStax Community Edition v2.2.8 (**[**Changelog**](http://git-wip-us.apache.org/repos/asf?p=cassandra.git;a=blob_plain;f=CHANGES.txt;hb=refs/tags/cassandra-2.2.8)**)**  
[Tarball](http://downloads.datastax.com/community/dsc-cassandra-2.2.8-bin.tar.gz), [MSI Installer (32-bit)](http://downloads.datastax.com/community/datastax-community-32bit_2.2.8.msi), [MSI Installer (64-bit)](http://downloads.datastax.com/community/datastax-community-64bit_2.2.8.msi), [RPM Using Yum](http://docs.datastax.com/en/cassandra/2.2/cassandra/install/installRHEL.html), [DEB Using Apt-Get](http://docs.datastax.com/en/cassandra/2.2/cassandra/install/installDeb.html)

**Download DataStax Community Edition v2.1.15 (**[**Changelog**](https://git1-us-west.apache.org/repos/asf?p=cassandra.git;a=blob_plain;f=CHANGES.txt;hb=refs/tags/cassandra-2.1.15)**)**  
[Tarball](http://downloads.datastax.com/community/dsc-cassandra-2.1.15-bin.tar.gz), [MSI Installer (32-bit)](http://downloads.datastax.com/community/datastax-community-32bit_2.1.15.msi), [MSI Installer (64-bit)](http://downloads.datastax.com/community/datastax-community-64bit_2.1.15.msi), [RPM Using Yum](http://docs.datastax.com/en/cassandra/2.1/cassandra/install/installRHEL_t.html), [DEB Using Apt-Get](http://docs.datastax.com/en/cassandra/2.1/cassandra/install/installDeb_t.html)

Now the next step is Cassandra drivers:

**DataStax Drivers for Apache Cassandra™**

|  |  |
| --- | --- |
| [DataStax Java Driver](https://github.com/datastax/java-driver) | [Documentation](http://docs.datastax.com/en/developer/java-driver/latest) |
| [DataStax C# Driver](https://github.com/datastax/csharp-driver) | [Documentation](http://docs.datastax.com/en/developer/csharp-driver/latest) |
| [DataStax Node.js Driver](https://github.com/datastax/nodejs-driver) | [Documentation](http://docs.datastax.com/en/developer/nodejs-driver/latest) |
| [DataStax Python Driver](https://github.com/datastax/python-driver) | [Documentation](http://docs.datastax.com/en/developer/python-driver/latest) |
| [DataStax Ruby Driver](https://github.com/datastax/ruby-driver) | [Documentation](http://docs.datastax.com/en/developer/ruby-driver/latest) |
| [DataStax C/C++ Driver](https://github.com/datastax/cpp-driver) | [Documentation](http://docs.datastax.com/en/developer/cpp-driver/latest) |
| [DataStax PHP Driver](https://github.com/datastax/php-driver) | [Documentation](http://docs.datastax.com/en/developer/php-driver/latest) |
| [DataStax Apache Cassandra™ Connector for Spark](https://github.com/datastax/spark-cassandra-connector) |  |

**Loaders and ODBC/JDBC Drivers**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| [DataStax Enterprise Graph Loader](https://portal.datastax.com/downloads.php?dsedownload=tar/enterprise/dse-graph-loader.tar.gz) (DSE 5.0 and later) |  |  | [Documentation](http://docs.datastax.com/en/latest-dse/datastax_enterprise/graph/dgl/dglOverview.html) |
| Simba ODBC Driver for Apache Spark™ (DSE 4.8 and later) | [Linux](https://portal.datastax.com/downloads.php?dsedownload=tar/odbc/1.2.2.1002/SimbaSparkODBC-1.2.2.1002-Linux.tar.gz) | Windows [32-bit](https://portal.datastax.com/downloads.php?dsedownload=tar/odbc/1.2.2.1002/SimbaSparkODBC32.msi) | [64-bit](https://portal.datastax.com/downloads.php?dsedownload=tar/odbc/1.2.2.1002/SimbaSparkODBC64.msi) | Documentation [Windows](http://docs.datastax.com/en/datastax_enterprise/latest/datastax_enterprise/spark/simbaOdbcDriverWindows.html) | [Linux](http://docs.datastax.com/en/datastax_enterprise/latest/datastax_enterprise/spark/simbaOdbcDriverLinux.html) |
| Simba JDBC Driver for Apache Spark™ (DSE 4.8 and later) | [JDBC 4.1](https://portal.datastax.com/downloads.php?dsedownload=tar/jdbc/1.1.2.1002/SimbaSparkJDBC41-1.1.2.1002.zip) |  | [Documentation](http://docs.datastax.com/en/datastax_enterprise/latest/datastax_enterprise/spark/simbaJdbcDriver.html) |
| Databricks ODBC Driver for Apache Spark™ (DSE 4.7 and earlier) | Linux [32-bit](https://portal.datastax.com/downloads.php?dsedownload=tar/enterprise/databricks/0.1.0.0001/Linux/SimbaSharkODBC-32bit-0.1.0.0001-1.i686.tar.gz) | [64-bit](https://portal.datastax.com/downloads.php?dsedownload=tar/enterprise/databricks/0.1.0.0001/Linux/SimbaSharkODBC-0.1.0.0001-1.x86_64.tar.gz) | Windows [32-bit](https://portal.datastax.com/downloads.php?dsedownload=tar/enterprise/databricks/0.1.0.0002/Windows/SimbaSharkODBC32.msi) | [64-bit](https://portal.datastax.com/downloads.php?dsedownload=tar/enterprise/databricks/0.1.0.0002/Windows/SimbaSharkODBC64.msi) | [Documentation](http://docs.datastax.com/en/datastax_enterprise/4.7/datastax_enterprise/spark/sharkOdbcDriver.html) |
| DataStax ODBC Driver for Apache Cassandra and DataStax Enterprise with CQL connector (DSE 4.8 and later) | Linux [32-bit](https://downloads.datastax.com/odbc-cql/2.4.1.1001/DatastaxCassandraODBC-32bit-2.4.1.1001.rpm) | [64-bit](https://downloads.datastax.com/odbc-cql/2.4.1.1001/DatastaxCassandraODBC-64bit-2.4.1.1001.rpm) | Windows [32-bit](https://downloads.datastax.com/odbc-cql/2.4.1.1001/DatastaxCassandraODBC32.msi) | [64-bit](https://downloads.datastax.com/odbc-cql/2.4.1.1001/DatastaxCassandraODBC64.msi) | [Documentation](https://downloads.datastax.com/odbc-cql/2.4.1.1001/Simba%20ODBC%20Driver%20for%20Cassandra%20Install%20Guide.pdf) |
| DataStax ODBC Driver for Hive |  | Windows [32-bit](https://downloads.datastax.com/odbc/DataStaxHiveODBC_x86.exe) | [64-bit](https://downloads.datastax.com/odbc/DataStaxHiveODBC_x64.exe) | [Documentation](http://www.datastax.com/documentation/latest-dse-hive-odbc?permalinkv1) |

Here I have chosen python driver to load the data or manipulate the data.

pip install cassandra-driver

The above command will install the Cassandra-driver version 3.8.0 which was not compatible with my python version. So I have installed the Cassandra-driver 2.7.2 using the below link

Pip install https://pypi.python.org/packages/source/c/cassandra-driver/cassandra-driver-2.7.2.tar.gz

Features provided with this driver:

* [Synchronous](http://datastax.github.io/python-driver/api/cassandra/cluster.html#cassandra.cluster.Session.execute) and [Asynchronous](http://datastax.github.io/python-driver/api/cassandra/cluster.html#cassandra.cluster.Session.execute_async) APIs
* [Simple, Prepared, and Batch statements](http://datastax.github.io/python-driver/api/cassandra/query.html#cassandra.query.Statement)
* Asynchronous IO, parallel execution, request pipelining
* [Connection pooling](http://datastax.github.io/python-driver/api/cassandra/cluster.html#cassandra.cluster.Cluster.get_core_connections_per_host)
* Automatic node discovery
* [Automatic reconnection](http://datastax.github.io/python-driver/api/cassandra/policies.html#reconnecting-to-dead-hosts)
* Configurable [load balancing](http://datastax.github.io/python-driver/api/cassandra/policies.html#load-balancing) and [retry policies](http://datastax.github.io/python-driver/api/cassandra/policies.html#retrying-failed-operations)
* [Concurrent execution utilities](http://datastax.github.io/python-driver/api/cassandra/concurrent.html)
* [Object mapper](http://datastax.github.io/python-driver/object_mapper.html)

Per the compatibility, sometimes we have to downgrade the python version using the below commands.

Conda install python=2.2.8

**CCM, A development tool for creating local Cassandra clusters**

CCM (Cassandra Cluster Manager) is a tool written by Sylvain Lebresne that creates multi-node cassandra clusters on the local machine. It is great for quickly setting up clusters for development and testing, and is the foundation that the cassandra distributed tests (dtests) are built on.

Installing CCM

CCM depends on the cql and PyYAML PyPI packages. For example, on ubuntu you can install them like this:

sudo apt-get install -y python-pip; sudo pip install cql PyYAML

**Step 2: Install CCM “Cassandra Cluster Manager”**

In a new Command Prompt/Powershell window (login as yourself)

type “pip install ccm” – which will automatically download and install [ccm](https://pypi.python.org/pypi/ccm)

> pip install ccm

Collecting ccm

Downloading ccm-2.0.6.tar.gz (56kB)

100% |################################| 57kB 1.8MB/s

Collecting pyYaml (from ccm)

Downloading PyYAML-3.11.tar.gz (248kB)

100% |################################| 249kB 1.7MB/s

Collecting six>=1.4.1 (from ccm)

Downloading six-1.10.0-py2.py3-none-any.whl

Installing collected packages: pyYaml, six, ccm

Running setup.py install for pyYaml

Running setup.py install for ccm

Successfully installed ccm-2.0.6 pyYaml-3.11 six-1.10.0

If the above command is not working, then try to install the recommended source of CCM is Sylvain’s git repo:

git clone https://github.com/pcmanus/ccm.git

Then install it like this:

cd ccm; sudo ./setup.py install; cd ..

**Step 3: Install “psutil (python system and process utilities)”**

In the same window as for Step 2:

type “pip install psutil” – – which will automatically download and install [psutil](https://pypi.python.org/pypi/psutil)

> pip install psutil

Collecting psutil

Downloading psutil-3.3.0-cp27-none-win\_amd64.whl (92kB)

100% |################################| 94kB 1.4MB/s

Installing collected packages: psutil

Successfully installed psutil-3.3.0

*Note: This window can now be closed*

**Step 4: Set-ExecutionPolicy Unrestricted** In a new Powershell window (login as local admin), type “Set-ExecutionPolicy Unrestricted”

\*Note: You must set the execution policy of Windows Powershell to allow CCM to launch instances of Cassandra. An unrestricted execution policy will also allow CCM to run on the regular command prompt (cmd) as well as Windows Powershell

PS C:\Windows\system32> Set-ExecutionPolicy Unrestricted

Execution Policy Change

The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose

you to the security risks described in the about\_Execution\_Policies help topic. Do you want to change the execution

policy?

[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y

**Step 5: Register PY extension**

*Note: Add .PY extension to environment variable $PATHEXT, to allow ccm to be executed from any location (run on PowerShell as administrator):*

In the same window as for Step 4 type

[Environment]::SetEnvironmentVariable("PATHEXT", "$env:PATHEXT;.PY", "MACHINE")

*Note: This window can now be closed*

**Step 6: Check if CCM is up and running**

In a new Command Prompt window (login as yourself) type:

>ccm status

No currently active cluster (use ccm cluster switch)

**Step 7: Update hosts file**

Open Notepad as Administrator and the following lines to the C:\Windows\System32\drivers\etc\hosts file:

#cassandra nodes

127.0.0.1 127.0.0.2

127.0.0.1 127.0.0.3

127.0.0.1 127.0.0.4

127.0.0.1 127.0.0.5

127.0.0.1 127.0.0.6

**Step 8: Create and populate a 3 node cluster using Cassandra v2.1.2**

*Note: This will download version 2.1.2 of Cassandra, build it and then use it to create a new CCM cluster called “mytestcluster”.*

* Cassandra installation path %USERPROFILE%.ccm\repository\2.1.2
* “test” cluster path %USERPROFILE%.ccm\test

C:\Users\myusername>ccm create mytestcluster -v 2.1.2

Downloading http://archive.apache.org/dist/cassandra/2.1.2/apache-cassandra-2.1.2-bin.tar.gz to c:\users\myusername\appdata\local\temp\ccm-qwauvs.tar.gz (21.735MB)

22790390 [100.00%]

Extracting c:\users\myusername\appdata\local\temp\ccm-qwauvs.tar.gz as version 2.1.2 ...

Current cluster is now: mytestcluster

C:\Users\myusername>ccm status

Cluster: 'mytestcluster'

------------------------

No node in this cluster yet

C:\Users\myusername>ccm populate -n 3

C:\Users\myusername>ccm status

Cluster: 'mytestcluster'

------------------------

node1: DOWN (Not initialized)

node3: DOWN (Not initialized)

node2: DOWN (Not initialized)

C:\Users\myusername>ccm start

Started: node1 with pid: 17432

Started: node3 with pid: 6308

Started: node2 with pid: 22484

C:\Users\myusername>ccm status

Cluster: 'mytestcluster'

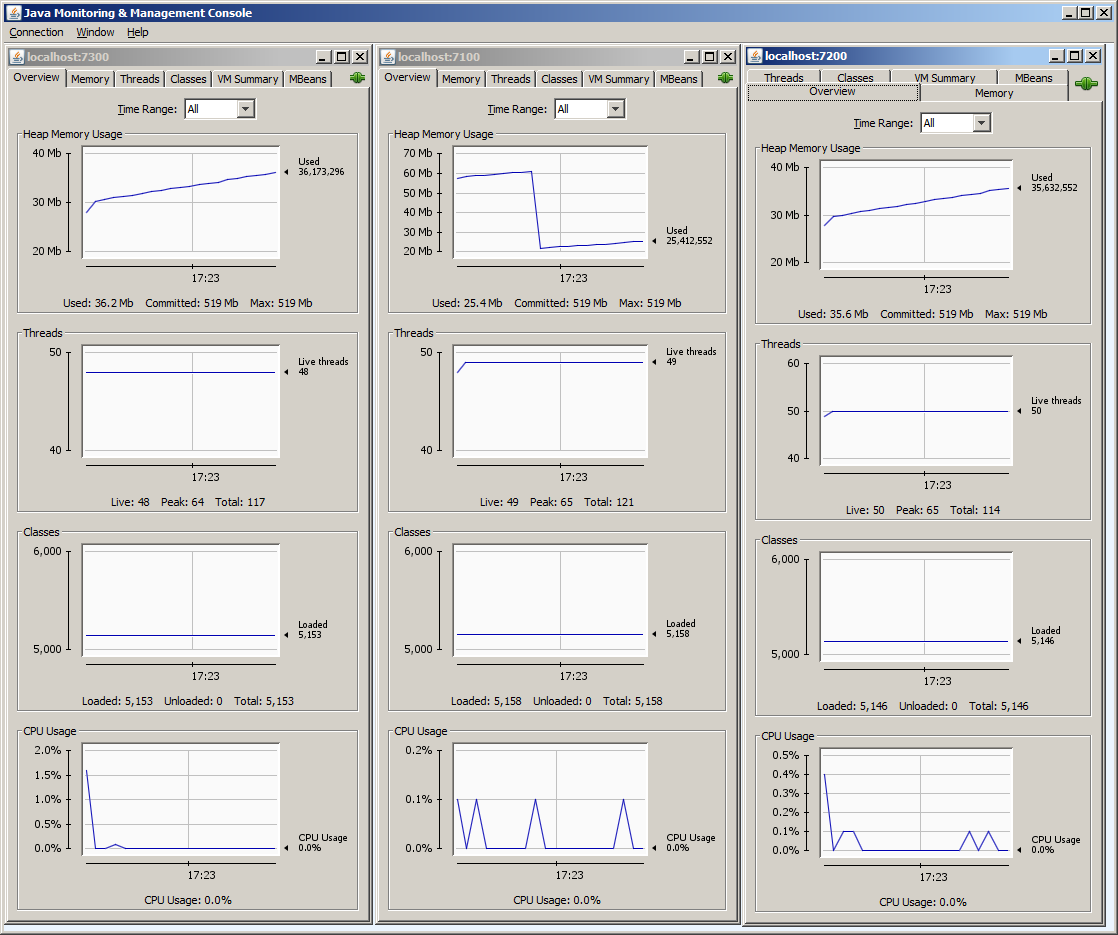
------------------------

node1: UP

node3: UP

node2: UP

C:\Users\myusername>ccm jconsole

[](https://i.stack.imgur.com/ZljKx.png)

C:\Users\myusername>ccm node1 show

node1: UP

cluster=mytestcluster

auto\_bootstrap=False

thrift=('127.0.0.1', 9160)

binary=('127.0.0.1', 9042)

storage=('127.0.0.1', 7000)

jmx\_port=7100

remote\_debug\_port=0

initial\_token=-9223372036854775808

pid=17432

C:\Users\myusername>ccm node2 show

node2: UP

cluster=mytestcluster

auto\_bootstrap=False

thrift=('127.0.0.2', 9160)

binary=('127.0.0.2', 9042)

storage=('127.0.0.2', 7000)

jmx\_port=7200

remote\_debug\_port=0

initial\_token=-3074457345618258603

pid=22484

C:\Users\myusername>ccm node3 show

node3: UP

cluster=mytestcluster

auto\_bootstrap=False

thrift=('127.0.0.3', 9160)

binary=('127.0.0.3', 9042)

storage=('127.0.0.3', 7000)

jmx\_port=7300

remote\_debug\_port=0

initial\_token=3074457345618258602

pid=6308

|  |
| --- |
|  |