Cassandra Installation and Configuration Guide

Installation

12/13/2025

Contents

Installation

Note: These instructions may vary for 64-bit versions. The following examples use 32-bit versions of Java and `prunsrv`.

Step 1: Downloading and Setting Environmental Variables

Extract the contents of the Cassandra (and Commons Daemon) archive(s) on each node. On Windows, use WinZip "Extract here..." or equivalent. On Linux, use `gunzip` or `tar -xvf`. If following the example below, the directories should be placed as such:

```
Windows: C:\Cassandra\apache-cassandra-2.2.x
Linux: /cassandra/apache-cassandra-2.2.x
```

The contents of the Commons Daemon archive are placed in the above installation directories in the `bin` folder, in a subdirectory named `daemon`, as below:

```
Windows: C:\Cassandra\apache-cassandra-2.2.x\bin\daemon
Linux: /cassandra/apache-cassandra-2.2.x/bin/daemon
```

Set the `JAVA_HOME` environment variable to the Java JRE/JDK root, for example:

```
set JAVA_HOME= C:\Program Files\Java\jdk1.8.0_73
or
export JAVA_HOME=/usr/java/jdk1.x.x_x
```

For Linux-like installs, edit `JAVA_HOME` in `%CASSANDRA_HOME%\bin\cassandra-in.sh`.

Step 2: Edit configuration files

The Cassandra distribution comes with a number of configuration files that should be edited (located in `%CASSANDRA_HOME%\conf` directory).

Step 2.0

The Cassandra distribution comes with a number of configuration files that should be edited (located in `%CASSANDRA_HOME%\conf` directory).

Step 2.1: Edit cassandra.yaml

The included `cassandra.yaml` contains default configurations for the Cassandra cluster.

When Cassandra versions 2.2.5 virtual nodes have been implemented, the initial_token should be left as is, with the exception of nodes that are being migrated from older 1.x.x versions. If this is the case, refer to the documentation specified in the yaml.

Ensure that the following options are pointing to the desired paths. Cassandra will create the directories on startup. Paths specified below are examples:

```
data_file_directories:
     - C:\Cassandra\apache-cassandra-2.2.x\data
 commitlog_directory: C:\Cassandra\apache-cassandra-2.2.x\commitlog
 saved_caches_directory: C:\Cassandra\apache-cassandra-2.2.x\saved_caches
```

Also in `cassandra.yaml`, configure the `cluster_name`, `key_cache_size_in_mb`, `counter_cache_size_in_mb`, `seeds`, `listen_address`, `start_rpc`, and `rpc_address`.

Follow the instructions in the yam regarding the settings for the following items, which all relate to memory allocation and number of processors that the installation platform has available.

```
concurrent_reads
concurrent_writes
concurrent_counter_writes
file_cache_size_in_mb
memtable_heap_space_in_mb
memtable_offheap_space_in_mb
commitlog_total_space_in_mb
```

- The `cluster_name` must be identical for all nodes within a Cassandra cluster.
- `key_cache_size_in_mb` should be set to `0` to disable key caching.
- `counter_cache_size_in_mb` should be set to `0` to disable counters.
- The seeds must be provided as a comma-delimited list of IP addresses to which new nodes will be able to contact for information about the Cassandra cluster. It is recommended that all nodes have the same list of seeds specified, and that all nodes be specified as seed nodes.
- `listen_address` is the IP address that other Cassandra nodes use to connect to this node.
- The `rpc_address` is the listen address for remote procedure calls (and clients, such as the `cassandra-cli`).

Make sure that `start_rpc` is set to `true`. If authentication is required, leave the `start_native_transport` at `true`. This port will be required to set the user name and passwords in Cassandra. If authentication is not required set `start_native_transport` to false.

> **Tip**
>
> If authentication is required – set the authenticator to `PasswordAuthenticator` and set the authorizer to `CassandraAuthorizer`. Refer to Step 5 for further information on setting the username and password.

The addresses are defaulted to localhost; it is recommended to set these to the IP address.

Verify that `storage_port` and `rpc_port` do not conflict with other configured services. The

`storage_port`, which defaults to `7000`, is the port used by Cassandra nodes for inter-node communication. The `rpc_port`, which defaults to `9160`, is used for remote procedure calls (such as `cassandra-cli`) and the Thrift service. This is the port to use when building clients for the Cassandra API.

If Cassandra is being deployed in a multiple data center configuration, the `endpoint_snitch` should be modified from the default of `SimpleSnitch` to `PropertyFileSnitch`, where the snitch then employs the `cassandra-topology.properties` to determine the nodes in the cluster. There are other snitch types available; please refer to the `cassandra.yaml` for the descriptions of these types. If a multiple data center deployment is chosen, the schema will require the correct replication information to be provided in the strategy option pairs that represent the cluster. An example for manually loading with the `PropertyFileSnitch` is described below. For Orchestration loading, the pairs in the strategy option should be the same in the persistence configuration.

Step 2.2: Edit cassandra-env and Cassandra Startup Script

If deployed on a Linux platform proceed to Step 2.2.3.

If deployed on a Windows platform, determine if Windows PowerShell is enabled and the current execution policies with the following command.

From a command prompt execute:

```
C:\>powershell Get-ExecutionPolicy -List
```

If PowerShell is available, the result should be similar to the following:

```
Scope          ExecutionPolicy
-----          ---------------
MachinePolicy  Undefined
UserPolicy     Undefined
Process        Undefined
CurrentUser    Unrestricted
LocalMachine   Restricted
```

If PowerShell is not available, the above command will fail, proceed to Step 2.2.1.

> **Important**
>
> The default cassandra.bat attempts to invoke the PowerShell startup scripts, and if not restricted will do so.

At this point, since PowerShell is available, decide if Cassandra startup should be performed with the PowerShell or legacy methods. If the choice is legacy, then restrict the PowerShell execution policy at the level of your choice. The following command will restrict the policy at the CurrentUser scope:

```
C:\> powershell Set-ExecutionPolicy -ExecutionPolicy Restricted -Scope CurrentUser
```

After successful execution of the above, re-check the policy with the powershell `Get-ExecutionPolicy -List` command. If set to your choice proceed to Step 2.2.1.

If your choice is to allow PowerShell execution of Cassandra startup, proceed to Step 2.2.2.

Step 2.2.1: Windows without Windows PowerShell or PowerShell Restricted

To configure Cassandra's JVM (Java Virtual Machine) and the JMX (Java Management Extensions) interface, edit `bin/cassandra.bat`.

The JMX port is used for management connections (such as nodetool). If necessary, edit the following line and ensure that there are no port conflicts with existing services.

To enable remote JMX access see this topic on the Apache website.

Note that remote access via the JMX port is not recommended due to the possibility of unintended access to that port, which could disrupt Cassandra operation.

If the default JMX port, 7199, needs to be modified, change the following line to the desired port.

```
cassandra.jmx.local.port=7199>
```

If Cassandra will be installed as a service with a service name other than ‘cassandra’, modify the `SERVICE_JVM` as shown below to the desired name.

```
:doInstallOperation
set SERVICE_JVM="cassandra_gre_01"
rem location of Prunsrv
set PATH_PRUNSRV=%CASSANDRA_HOME%\bin\daemon\

# For x64 installations, (OS and JAVA) set
PATH_PRUNSRV=%CASSANDRA_HOME%\bin\daemon\amd64
set PR_LOGPATH=%CASSANDRA_HOME%\logs
```

Proceed to Step 2.3.

Step 2.2.2: Windows with Windows PowerShell Execution Policy Unrestricted

To configure Cassandra's JVM (Java Virtual Machine) options and the JMX (Java Management Extensions) interface, edit `conf/ cassandra-env.ps1`.

The JMX port is used for management connections (such as nodetool). If necessary, edit the following line and ensure that there are no port conflicts with existing services.

To enable remote JMX access see: https://wiki.apache.org/cassandra/JmxSecurity.

Note that remote access via the JMX port is not recommended due to the possibility of unintended access to that port, which could disrupt Cassandra operation.

If the default JMX port, 7199, needs to be modified, change the `JMX_PORT` to the desired port.

```
#   Specifies the default port over which Cassandra will be available for
#   JMX connections.
$JMX_PORT="7199"
```

The PowerShell function, `CalculateHeapSizes`, set these as follows:

```
# set max heap size based on the following
```

```
# max(min(1/2 ram, 1024MB), min(1/4 ram, 8GB))
# calculate 1/2 ram and cap to 1024MB
# calculate 1/4 ram and cap to 8192MB
# pick the max
```

If the heap sizes need to be defined rather than allowing the script to determine the values, modify the following to set the values:

```
    #$env:MAX_HEAP_SIZE="4096M"
    #$env:HEAP_NEWSIZE="800M"
    CalculateHeapSizes
```

If Cassandra will be installed as a service with a service name other than ‘cassandra’, edit `bin/cassandra.ps1`, and set the `SERVICE_JVM` to the desired name. Function HandleInstallation

```
{
                    $SERVICE_JVM = """CassandraForCluster1"""
                    $PATH_PRUNSRV = "$env:CASSANDRA_HOME\bin\daemon"
        …
```

Proceed to Step 2.3.

Step 2.2.3: Linux

To configure Cassandra's JVM (Java Virtual Machine) and the JMX (Java Management Extensions) interface, edit `conf/cassandra-env.sh`.

The JMX port is used for management connections (such as nodetool). If necessary, edit the following line and ensure that there are no port conflicts with existing services.

To enable remote JMX access see: https://wiki.apache.org/cassandra/JmxSecurity.

Note that remote access via the JMX port is not recommended due to the possibility of unintended access to that port, which could disrupt Cassandra operation.

If the default JMX port, 7199, needs to be modified, change the following line to the desired port.

```
#Specifies the default port over which Cassandra will be available for
# local JMX connections.
# JMX_PORT="7199"
```

Step 2.3: Edit logback.xml

Logging options can be found in `conf/logback.xml`. The default directory for logging is `%CASSANDRA_HOME%\logs`, with log file name `system.log.n`, where `n` is the wrap number. Cassandra versions 2.2.x and later, by default, enables debug level logging to separate file names. In order to disable `debug.log`, comment-out the `ASYNCDEBUGLOG` appender reference in the root level section.

Step 2B: Set up the Cassandra service (Windows)

Install the Cassandra service if desired.

To install: 'bin\cassandra.bat -INSTALL'

To uninstall: 'bin\cassandra.bat -UNINSTALL'

Once installed, you will be able to find and start up the Cassandra service from the Windows Services GUI. The name of the service depends on the value of `SERVICE_JVM`.

Step 3: Start up Cassandra

Linux:

Start up Cassandra by invoking `bin/cassandra -f`. It will start up in the foreground and will log to std-out. If you don't see any *error* or *fatal* log messages or Java stack traces, then chances are you've succeeded.

Press "Control-C" to stop Cassandra.

If you start up Cassandra without "-f" option, it will run in background, so you need to kill the process to stop.

Windows:

To start the service from Windows, there are two options:

1. Use the Windows Services GUI
2. From commandline, in the `daemon` dir (as set above):

- start: prunsrv.exe start <Cassandra Service Name>
- stop: prunsrv.exe stop <Cassandra Service Name>;

NOTE: There is currently a bug in prunsrv version 1.0.15.0 which prevents the service from being stopped. Use version 1.0.14.0 to prevent this, available from http://archive.apache.org/dist/commons/daemon/binaries/windows/

Step 4: Using nodetool

Once all Cassandra nodes have been started, we can check the status of the Cassandra cluster using nodetool.

On one of the Cassandra nodes, from `%CASSANDRA_HOME%`, run

```
/bin/nodetool -h <listen_address> -p <jmx_port> status
```

The output of this command should be similar to the example found in the Useful Tools section. There should be as many addresses in the list as the number of Cassandra nodes configured. If there are fewer nodes than expected, make sure that all nodes have unique initial tokens.

Step 5: Setting Username and Password for Authentication (Optional)

Once all Cassandra nodes have been started, set the user name and password that will be used for client connection login authorization. This is performed with the `cqlsh.bat` procedure for Windows-based deployments, and with `cqlsh` procedure for Linux-based deployments. These are located in the Cassandra installation bin directory.

Step 5.1: Login using CQLSH with Default Superuser

Start `cqlsh` with the Cassandra default superuser, user name `cassandra` and password `cassandra`. Note that the host and port that CQL connects to by default are 'localhost' and 9042. The host should be that which was specified in the `listen_address`, and the port should be what was defined in `native_transport_port`. Set the environment variable `CQLSH_HOST` to that specified in the listen_address. If the `native_transport_port` was changed from the default, the port can be set in the `CQLSH_PORT` environment variable. The following example for a Windows deployment had `CQLSH_HOST=dwswin7`. The port was not changed.

```
C:\Cassandra\apache-cassandra-2.2.5>.\bin\cqlsh.bat -u cassandra -p cassandra
Connected to Orchestration Windows PerfTest Cluster 2.1.12 at dwswin7:9042.
[cqlsh 5.0.1 | Cassandra 2.2.5 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing.  Install to enable tab completion.
cassandra@cqlsh>
```

Step 5.2: Update the system_auth Replication Factor

Once connected, display the properties of the `system_auth` keyspace. This keyspace holds the authentication information that will be defined as described below. The replication factor of this keyspace should be increased, if the cluster has a small number of nodes, i.e., less than 10. In this case, set the replication factor to the number of nodes in the Cassandra cluster, or in each datacenter for multiple datacenter deployments. The replication factor will determine the number of instances that can fail and a client will still be able to login. The following describes the method to accomplish this for each datacenter with two Cassandra instances.

```
cassandra@cqlsh> ALTER KEYSPACE system_auth WITH
 replication = {'class': 'SimpleStrategy', 'replication_factor': '2'};
cassandra@cqlsh> DESCRIBE KEYSPACE system_auth;
CREATE KEYSPACE system_auth WITH replication = {'class': 'SimpleStrategy',
'replication_factor': '2'}
 AND durable_writes = true;
..
```

The DESCRIBE KEYSPACE shows the CQL that would be used to create the keyspace; it also displays the tables (column families) within the keyspace. For multiple datacenters, the `ALTER KEYSPACE` must be performed in each datacenter.

Step 5.3: Create the Desired User and Password

Now that the replication factor has been set on the `system_auth` keyspace, create a new user and password, as below. Note that this must be performed in each datacenter, also that the help requires

an underscore, while the command does not. For mixed or special characters, the single quotes are required:

```
cassandra@cqlsh> help CREATE_ROLE
cassandra@cqlsh> CREATE ROLE genesys WITH PASSWORD = 'g3n3sys!' AND LOGIN = true AND
SUPERUSER = true;
```

In this case a superuser genesys with password g3n3sys! is created. Once a new superuser is created, it is recommended that the default superuser cassandra be dropped. The following process describes how this can be accomplished.

First, perform a LIST USERS to make sure the new superuser is there:

```
cassandra@cqlsh> LIST USERS;
name      | super
-----------+-------
cassandra | True
genesys   | True
```

Now the original, default, superuser can be dropped, which must be done for each datacenter.

> **Tip**
>
> Make sure the created superuser and password are remembered; if not the Cassandra instances will have to be re-installed to allow the defaults to be used.

To drop the default user, `cqlsh` must be restarted with the new superuser:

```
cassandra@cqlsh> quit;
C:\Cassandra\apache-cassandra-2.2.5>.\bin\cqlsh.bat -u genesys -p g3n3sys!
```

Now drop the cassandra user:

```
genesys@cqlsh> DROP USER cassandra;
genesys@cqlsh> LIST USERS;
name   | super
-----------+-------
genesys| True
```

At this point, the new superuser can be specified in the Orchestration `persistence` section, options `username` and `password`, or another, non-superuser, may be created for that purpose, reserving the superuser for administrative purposes. An example follows.

```
genesys@cqlsh> CREATE ROLE orspersistence WITH PASSWORD = 'orsG3n3sys!' AND LOGIN = true;
genesys@cqlsh> LIST USERS;

cassandra@cqlsh> LIST USERS;

name           | super
----------------+-------
genesys        | True
orspersistence | False
```

Next, determine if the Orchestration KEYSPACE exists in Cassandra with the following:

```
genesys@cqlsh> DESCRIBE KEYSPACES;
```

The result will be something like:

```
system_traces system_auth system system_distributed
```

or

```
system_traces system_auth system “Orchestration” system_distributed
```

In the first case, proceed to step 5.3.1. In the second case, proceed to step 5.3.2.

Step 5.3.1 Orchestration KEYSPACE Does Not Yet Exist in Cassandra

The non superuser must be granted permission to create KEYSPACEs, the following will set the permission:

```
genesys@cqlsh>GRANT CREATE ON ALL KEYSPACES TO orspersistence;
```

Next, set the username and password in the Orchestration options to the non superuser name and password, then start Orchestration. This will create the Orchestration KEYSPACE. Check that the KEYSPACE exists with the DESCRIBE KEYSPACES command. If it exists, stop Orchestration, then proceed to step 5.3.2.

Step 5.3.2 Orchestration KEYSPACE Exists in Cassandra

The non-superuser must be granted access to the Orchestration keyspace. This can be done with the following CQL command. (note that if mixed case is present the double quotes are required)

> **Tip**
> if mixed case is present the double quotes are required.

```
genesys@cqlsh>GRANT ALL PERMISSIONS ON KEYSPACE “Orchestration” TO orspersistence;
```

If not already done, set the non-superuser name and password in the Orchestration options and start Orchestration.

Step 5.3.3 Troubleshooting Authorization

If the Orchestration connection to Cassandra cannot be authorized, the following entry will appear in the Orchestration log, and Orchestration will continue to run without persistence enabled:

```
Std 23002 ORS Cassandra schema version ORS8130000 No Cassandra hosts available
<Cassandra node [172.21.83.74] login failed - Transport exception>. Persistence is
unavailable.
```

Check the steps taken in 5.3.

If the Orchestration KEYSPACE does not exist and the non-superuser username has not been granted

CREATE permission on all KEYSPACES, the following entry will appear in the Orchestration log, Orchestration will continue to run without persistence enabled:

```
Std 23002 ORS Cassandra schema version ORS8130000 No Cassandra hosts available
<User orspersistence has no CREATE permission on <all keyspaces> or any of its parents>.
Persistence is unavailable.
```

Check the steps taken in 5.3.1.

If the Orchestration non-superuser username has not been granted permissions to the Orchestration keyspace, the following entry will appear in the Orchestration log, and Orchestration will be terminated:

```
Std 23001 ORS Cassandra schema version ORS8130000 Schema validation failed
<Insert into schema version failed>. Orchestration is terminating.
Std 23011 Orchestration Server::Stop() entered - stopping components
```

Check the steps taken in 5.3.2.