Troubleshooting

Contents

Chapter 1. Troubleshooting DataStax Enterprise

Starting and installing DSE

Error messages when libaio not installed (DSE 6.0 and 6.7)........................................................................... 6

MX4J warning message during DataStax Enterprise installation........................................................................... 6

DataStax Enterprise fails to start after configuring authentication........................................................................... 6

JNA fails to initialize.................................................................................................................................................. 6

Cannot write to tmp directory.................................................................................................................................. 7

DSE does not start after the DSEFS keyspace is dropped....................................................................................... 7

DataStax Enterprise times out when starting........................................................................................................... 7

Linux related problems............................................................................................................................................... 7

Peculiar Linux kernel performance problem on NUMA systems................................................................................. 7

Reads are getting slower while writes are still fast...................................................................................................... 7

Nodes seem to freeze after some period of time........................................................................................................ 9

Nodes are dying with OOM errors............................................................................................................................ 9

Netd tool or JMX connections failing on remote nodes............................................................................................ 9

Handling schema disagreements.............................................................................................................................. 9

Garbage collection pauses........................................................................................................................................... 10

Java reports an error saying there are too many open files...................................................................................... 10

Insufficient user resource limits errors.................................................................................................................... 10

No DSE processing but high CPU usage.................................................................................................................. 10

Cannot initialize class org.xerial.snappy.Snappy.................................................................................................. 11

Firewall idle connection timeout causes nodes to lose communication................................................................. 11

Using nodetool sjk.................................................................................................................................................... 12

DSE Graph................................................................................................................................................................. 12

Inconsistent handling of string vertex/edge ids collections in DSE Graph steps................................................... 12

Directly querying Graph data with user-defined vertex ids using solr_query results in null returns....................... 12

Dropping a graph or a portion of a graph................................................................................................................ 13

Gremlin console hangs or behaves erratically........................................................................................................ 13

Queries sporadically fail with LOCAL_ONE/LOCAL_QUORUM............................................................................. 13

Consistency level and graph.addVertex()................................................................................................................ 13

Issues creating a graph cluster using Lifecycle Manager (LCM)........................................................................ 14

Shutting down Studio Gremlin process.................................................................................................................. 15
### Chapter 15

Dropping edge property drops edges................................................................. 15
DSE Analytics................................................................................................. 15
DSE Search....................................................................................................... 15
  DSE Search node start.................................................................................. 16
Handling inconsistencies in query results....................................................... 18
Tracing Solr HTTP requests........................................................................... 19
Security............................................................................................................... 20
  DSE fails to start............................................................................................ 20
  SSL certificate doesn't match....................................................................... 21
  SSL exceptions occur on start up or no connections.................................... 21
  Connection errors with cqlsh and other DSE tools..................................... 22
Studio............................................................................................................... 23
Other troubleshooting...................................................................................... 25
  Ring view differences................................................................................... 25
  sstableloader missing rows......................................................................... 25
  Prepared statements discarded.................................................................... 25
  Purging gossip state on a node..................................................................... 25
  Recovering expired data caused by TTL year 2038 problem......................... 26

### Chapter 2. Troubleshooting DSE OpsCenter................................................. 29

SSL issues........................................................................................................ 29
  Configuring client-to-node encryption......................................................... 29
  Troubleshooting connections with OpsCenter to DSE with SSL.................. 30
Browser issues................................................................................................. 30
  Zero nodes or OpsCenter UI not loading..................................................... 30
  Microsoft web browsers not supported....................................................... 31
SSTable issues................................................................................................. 31
  SSTables not cleaned up after failed restore................................................ 31
  The SSTables in this snapshot ‘<tag>’ are not compatible............................. 31
Other troubleshooting....................................................................................... 32
  DSE version not supported by LCM............................................................. 32
  High CPU usage by opscenterd.................................................................... 32
  Collecting or displaying metrics.................................................................. 33
  Restoring materialized views....................................................................... 33
  DataStax Agent port setting conflict........................................................... 34
  Java not installed or JAVA_HOME environment variable not set.................... 34
Troubleshooting

Insufficient user resource limits errors

Error getting version update information

OpsCenter cannot start
Chapter 1. Troubleshooting DataStax Enterprise

Use the following pages to locate solutions or workarounds for issues experienced with DataStax Enterprise (DSE).

Troubleshooting starting and installing DataStax Enterprise

The DataStax Enterprise Help Center also provides troubleshooting information.

Error messages when libaio not installed (DSE 6.0 and 6.7)

This topic has been moved to the DataStax Support Knowledge Base. See Error messages when the AIO library (libaio) is not installed on DSE 6.0 and DSE 6.7.

The DataStax Support Knowledge Base provides information to help resolve issues. Search for product-specific troubleshooting, how to articles, FAQs, and more.

MX4J warning message during DataStax Enterprise installation

This topic has been moved to the DataStax Support Knowledge Base.

The DataStax Support Knowledge Base provides information to help resolve issues. Search for product-specific troubleshooting, how to articles, FAQs, and more.

DataStax Enterprise fails to start after configuring authentication

This topic has been moved to the DataStax Support Knowledge Base.

The DataStax Support Knowledge Base provides information to help resolve issues. Search for product-specific troubleshooting, how to articles, FAQs, and more.

JNA fails to initialize

The DataStax Enterprise Help Center also provides troubleshooting information.

DataStax Enterprise fails to start because the JNA temporary directory is not available. If the tmp directory is inaccessible the following error appears in the system log:

```
tail -3 /var/log/cassandra/system.log
```

The error is similar to:

```
ERROR main 2015-12-18 09:57:00,879 CassandraDaemon.java:213 - JNA failing to initialize properly. Use -Dcassandra.boot_without_jna=true to bootstrap even so.
INFO Thread-2 2015-12-18 09:57:00,880 DseDaemon.java:418 - DSE shutting down...
INFO Thread-2 2015-12-18 09:57:00,881 PluginManager.java:103 - All plugins are stopped.
```

Configure the database JNA temporary path to an executable directory.

- In cassandra-env.sh, add the line:

  ```
  JVM_OPTS="$JVM_OPTS -Djna.tmpdir=/path_to_directory"
  ```

  Where `path_to_directory` is the absolute path to a directory to which the database user has read, write, and execute permissions.
Troubleshooting DataStax Enterprise

Unable to write to the standard tmp directory for JNA in DataStax Enterprise

The DataStax Enterprise Help Center also provides troubleshooting information.

Set JNA temporary path to be executable. In cassandra-env.sh, add the line:

```
JVM_OPTS="$JVM_OPTS -Djna.tmpdir=/var/tmp"
```

DSE does not start after the DSEFS keyspace is dropped

This topic has been moved to the DataStax Support Knowledge Base.

The DataStax Support Knowledge Base provides information to help resolve issues. Search for product-specific troubleshooting, how to articles, FAQs, and more.

DataStax Enterprise times out when starting

This topic has been moved to the DataStax Support Knowledge Base.

The DataStax Support Knowledge Base provides information to help resolve issues. Search for product-specific troubleshooting, how to articles, FAQs, and more.

Troubleshooting Linux related problems

The DataStax Enterprise Help Center also provides troubleshooting information.

Peculiar Linux kernel performance problem on NUMA systems

The DataStax Help Center also provides troubleshooting information.

Problems due to zone_reclaim_mode.

The Linux kernel can be inconsistent in enabling/disabling zone_reclaim_mode. This can result in odd performance problems:

- Random huge CPU spikes resulting in large increases in latency and throughput.
- Programs hanging indefinitely apparently doing nothing.
- Symptoms appearing and disappearing suddenly.
- After a reboot, the symptoms generally do not show again for some time.

To ensure that zone_reclaim_mode is disabled:

```
$ echo 0 > /proc/sys/vm/zone_reclaim_mode
```

Reads are getting slower while writes are still fast

The DataStax Enterprise Help Center also provides troubleshooting information.

Too many SSTables can cause slow reads. Take the following steps to determine and correct slow reads:

**Determine the total number of SSTables for each table.**

Check this number with `nodetool tablestats`.

**Get the number of SSTables consulted for each read.**

Check this number with `nodetool tablehistograms`. A median value over 2 or 3 is likely causing problems.

**Make sure SSTables are not flushing too frequently**

Check `debug.log` for `enqueuing flush` messages and note the size and frequency of flushes:

- If the SlabPoolCleaner thread is frequently enqueuing many small flushes, increase `memtable_cleanup_threshold`. 
The value of memtable_cleanup_threshold should be inversely proportional to the number of tables that receive heavy writes.

- If spare memory is available, consider increasing `memtable_heap_space_in_mb` (deprecated) or `memtable_offheap_space_in_mb` (deprecated).

- If the COMMIT-LOG-ALLOCATOR thread is frequently enqueuing flushes, increase `commitlog_total_space_in_mb`.

The number of flushes enqueued by COMMIT-LOG-ALLOCATOR should be a small minority of the total flushes enqueued.

- Check the pending compactions using `nodetool compactionstats`.

  Even if flushes are not excessively frequent, compactions might not be able to keep up. If a high number of pending compactions exist, compactions are not keeping up. Note that this number is only an estimate for LeveledCompactionStrategy.

**Make sure the compaction_throughput_mb_per_sec is set appropriately for your storage.**

The default value of 16 MB/sec for `compaction_throughput_mb_per_sec` is chosen for spinning disks; SSDs can use a much higher setting such as 128 MB/sec or more.

1. Temporarily adjust the value using `nodetool setcompactionthroughput`.

2. Watch the I/O utilization using `iostat -x -t 10`, which shows the averages for 10 second intervals and prints timestamps:
   - `%iowait` over 1 indicates that the node is starting to get I/O bound.
   - The acceptable bounds for `await` (Average Wait in Milliseconds) are:
     # Most SSDs: below 10 ms.
     # Most 7200 RPM spinning disks: below 200 ms.

3. Once you have found a good throughput for your system, set it permanently in `cassandra.yaml`.

4. If your I/O is not able to keep up with the necessary compaction throughput, you probably need to get faster disks or add more nodes.

**If you have set a high compaction throughput but I/O utilization is low and compactions are still not keeping up, the compactions may be CPU-bound.**

Check the per-core CPU utilization of CompactionExecutor threads.

If the threads are utilizing 100% of a single core, the compaction may be CPU bound. Increasing `concurrent_compactors` will allow multiple concurrent compactions of different sets of SSTables, but compaction of each set of SSTables is inherently single-threaded. If you are using LeveledCompactionStrategy (LCS), you need to either switch to SizeTieredCompactionStrategy (STCS) or add more nodes to spread compaction load.

Increasing concurrent compactors beyond the number of physical CPU cores (not Hyperthreaded cores) can be counter productive. Using all available CPU for compaction means no CPUs remain to handle reads and writes. If you need to continue to service requests while catching up on compactions, be sure to leave 1 or 2 physical CPUs free for reads and/or writes.

**Make sure that there is enough free memory for file cache (page cache).**

The `free -m` command shows the amount of memory available for caches. You should have enough memory available for file cache to hold your hot working set in memory.

**Switch to SizeTieredCompactionStrategy.**

If using LeveledCompactionStrategy (LCS) and the above steps haven’t worked, consider switching to SizeTieredCompactionStrategy (STCS). LCS uses more resources to compact than STCS. Often nodes that are falling behind while compacting with LCS can easily keep up using STCS.
Troubleshooting DataStax Enterprise

**Nodes seem to freeze after some period of time**

The DataStax Help Center also provides troubleshooting information.

Some portion of the JVM is being swapped out by the Linux operating system (OS).

**Linux**

Check your system.log for messages from the GCInspector. If the GCInspector is indicating that either the ParNew or ConcurrentMarkSweep collectors took longer than 15 seconds, there is a high probability that some portion of the JVM is being swapped out by the OS.

DataStax strongly recommends disabling swap entirely (sudo swapoff --all). Because the database has multiple replicas and transparent failover, it is preferable for a replica to be killed immediately when memory is low rather than go into swap. This allows traffic to be immediately redirected to a functioning replica instead of continuing to hit the replica that has high latency due to swapping. If your system has a lot of DRAM, swapping still lowers performance significantly because the OS swaps out executable code so that more DRAM is available for caching disks.

```bash
$ sudo swapoff --all
```

To make this change permanent, remove all swap file entries from /etc/fstab.

    If you insist on using swap, you can set vm.swappiness=1. This allows the kernel swap out the absolute least used parts.

If the GCInspector isn't reporting very long GC times, but is reporting moderate times frequently (ConcurrentMarkSweep taking a few seconds very often) then it is likely that the JVM is experiencing extreme GC pressure and will eventually OOM. See Nodes are dying with OOM errors.

**Nodes are dying with OOM errors**

The DataStax Enterprise Help Center also provides troubleshooting information.

Nodes are dying with OutOfMemory exceptions.

Check for these typical causes:

**Row cache is too large, or is caching large rows**

Row cache is generally a high-end optimization. Try disabling it and see if the OOM problems continue.

**There is a large user query running on the node which takes up all the heap**

In production, understand and test all queries upfront to avoid arbitrary query patterns. Test to discover each query's max response size. Paging in CQL can often prevent a query from pulling too much data at once.

If none of these apply to your situation, try loading the heap dump in MAT and see which class is consuming the bulk of the heap for clues.

**Nodetool or JMX connections failing on remote nodes**

The DataStax Enterprise Help Center also provides troubleshooting information.

Nodetool commands can be run locally but not on other nodes in the cluster.

If you can run nodetool commands locally but not on other nodes in the ring, it might be common JMX connection problem. Add an entry like the following in cassandra-env.sh on each node:

```bash
JVM_OPTS = "$JVM_OPTS -Djava.rmi.server.hostname=public name"
```

The default settings start up JMX only on the local node. In cassandra-env.sh or cassandra-env.psl, add JMX authentication to contact remote nodes. See Jmx Security for details.

If you still cannot run nodetool commands remotely after making this configuration change, do a full evaluation of your firewall and network security. The nodetool utility communicates through JMX on port 7199.

**Handling schema disagreements**

This topic has been moved to the DataStax Support Knowledge Base.
Garbage collection pauses

The DataStax Support Knowledge Base provides information to help resolve issues. Search for product-specific troubleshooting, how to articles, FAQs, and more.

Garbage collection is the process by which Java removes data that is no longer needed from memory. A garbage collection pause, also known as a stop-the-world event, happens when a region of memory is full and the JVM requires space to continue. During a pause all operations are suspended. Because a pause affects networking, the node can appear as down to other nodes in the cluster. Additionally, any Select and Insert statements will wait, which increases read and write latencies. Any pause of more than a second, or multiple pauses within a second that add up to a large fraction of that second, should be avoided. The basic cause of the problem is that the rate of data stored in memory outpaces the rate at which data can be removed.

The two most common log messages that indicate excessive pausing is occurring are:

```
INFO [ScheduledTasks:1] 2013-03-07 18:44:46,795 GCInspector.java (line 122) GC for ConcurrentMarkSweep: 1835 ms for 3 collections, 2606015656 used; max is 10611589120
INFO [ScheduledTasks:1] 2013-03-07 19:45:08,029 GCInspector.java (line 122) GC for ParNew: 9866 ms for 8 collections, 2910124308 used; max is 6358564864
```

Causes of garbage collection pause include:

- If the problem is recent, check for any recent applications changes.
- Excessive tombstone activity: often caused by heavy delete workloads.
- Large row updates or large batch updates: reduce the size of the individual write below 1 Mb (at the most).
- Extremely wide rows: manifests as problems in repairs, selects, caching, and elsewhere.

Server side factors include:

- Missing or strange JVM parameters. Compare those set to the default settings shipped with latest product version.
- JNA not found.
- Swap enabled.

For more information, see Tuning Java Virtual Machine.

Java reports an error saying there are too many open files

This topic has been moved to the DataStax Support Knowledge Base.

Insufficient user resource limits errors

This topic has been moved to the DataStax Support Knowledge Base.

No DataStax Enterprise processing but high CPU usage

Extremely high CPU usage but no DataStax Enterprise processing on Linux platforms.

Check the CPU usage for the process `khugepaged`. It may run as high as 100%, blocking other processes.
Cause:
Many modern Linux distributions ship with Transparent Hugepages enabled by default. When Linux uses Transparent Hugepages, the kernel tries to allocate memory in large chunks (usually 2 MB), rather than 4K. This can improve performance by reducing the number of pages the CPU must track. However, some applications still allocate memory based on 4K pages. This can cause noticeable performance problems when Linux tries to defrag 2 MB pages. For more information, see Cassandra Java Huge Pages and this RedHat bug report.

Possible solutions:

- A temporary fix: drop caches by entering:

  ```bash
  sync && echo 3 > /proc/sys/vm/drop_caches
  ```

- A better solution: disable defrag for hugepages by entering:

  ```bash
  echo never | sudo tee /sys/kernel/mm/transparent_hugepage/defrag
  ```

- Another alternative: add `-XX:+AlwaysPreTouch` to the `jvm.options` file. This change should be tested carefully before being put into production. For details, see Tuning Java Virtual Machine and blog post.

**Cannot initialize class org.xerial.snappy.Snappy**

The DataStax Enterprise Help Center also provides troubleshooting information.

On Linux platforms, an error may occur when Snappy compression/decompression is enabled although its library is available from the classpath.

```java
java.util.concurrent.ExecutionException: java.lang.NoClassDefFoundError:
  Could not initialize class org.xerial.snappy.Snappy

Caused by: java.lang.NoClassDefFoundError: Could not initialize class
  org.xerial.snappy.Snappy
  at org.apache.cassandra.io.compress.SnappyCompressor.initialCompressedBufferLength
    (SnappyCompressor.java:39)
```

The native library `snappy-1.0.4.1-libsnappyjava.so` for Snappy compression is included in the `snappy-java-1.0.4.1.jar` file. When the JVM initializes the JAR, the library is added to the default temp directory. If the default temp directory is mounted with a noexec option, it results in the above exception.

One solution is to specify a different temp directory that has already been mounted without the noexec option, as follows:

- If you use the dse start command `$(_BIN/dse cassandra or $(_BIN/cassandra, simply append the command line:

  ```bash
  $ bin/dse cassandra -t -Dorg.xerial.snappy.tempdir=/path/to/newtmp
  ```

- If starting from a package using service dse start or service cassandra start, add a system environment variable `JVM_OPTS` with the value:

  ```bash
  JVM_OPTS=-Dorg.xerial.snappy.tempdir=/path/to/newtmp
  ```

  The default `cassandra-env.sh` looks for the variable and appends to it when starting the JVM.

**Firewall idle connection timeout causes nodes to lose communication during low traffic times on Linux**

Be sure to follow DataStax recommended settings for using DataStax Enterprise (DSE) in production environments. See Recommended production settings.

See the DataStax Support Knowledge Base for steps on configuring keepalive with Linux TCP settings.
Troubleshooting DataStax Enterprise

The DataStax Support Knowledge Base provides information to help resolve issues. Search for product-specific troubleshooting, how to articles, FAQs, and more.

Using nodetool sjk

Use nodetool sjk mx to gather database information from MBeans. See Using nodetool sjk 6.7 | 6.0 | 5.1.

DSE Graph troubleshooting

The DataStax Enterprise Help Center also provides troubleshooting information.

Inconsistent handling of string vertex/edge ids collections in DSE Graph steps

The DataStax Enterprise Help Center also provides troubleshooting information.

The string formatting for vertices with user-defined vertex ids has changed. Invoking toString on a user-defined vertex id containing a text property, or on an edge id that is incident upon a vertex with a user-defined vertex id now returns a value that double-quotes that text property value and escapes the value’s internal double-quotes. This change avoids cases where the old format could lead to irresolvable parsing ambiguity. For example, examine the input value and returned value for the following graph statements:

```
gremlin> graph.addVertex(T.label, "v", "p", "a""b")
==>v{"label=v, p="a""b"}
gremlin> g.V().id().toList()
==>"v{"label=v, p="a""b"}"
gremlin> g.V().id().next().toString()
==>"v{"label=v, p="a""b"}"
gremlin> g.V().toList()
==>v{"label=v, p="a""b"}"
```

Directly querying Graph data with user-defined vertex ids using solr_query results in null returns

The DataStax Enterprise Help Center also provides troubleshooting information.

The valid way to issue a direct CQL query that includes a solr_query is to retrieve only the user-defined vertex id fields in the select statement. Moving from DSE 5.0 to 6.0 may cause issues with previously created queries.

Because of these complexities, DataStax currently does not recommend querying the underlying Graph generated CQL tables or SOLR Indexes. Such queries will be required to rewrite their queries in a future DSE Graph release.

In DSE 50. and DSE 5.1, when someone issues a CQL statement using the SOLR API in the WHERE clause against a graph table that contains a DSE Search index (assuming search index was created by DSE Graph)

- The end user will see unexpected results like the following:

```
cassandra_admin@cqlsh:ecdc_graph> select first_name, last_name, record_type from contact_p where solr_query = 'last_name:Abbott';
```

<table>
<thead>
<tr>
<th>first_name</th>
<th>last_name</th>
<th>record_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>null</td>
<td>null</td>
<td>G</td>
</tr>
<tr>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
</tbody>
</table>
In DSE 6.0, the user will get an error message telling them that they cannot do that query.

**Dropping a graph or a portion of a graph**

The DataStax Enterprise Help Center also provides troubleshooting information.

Dropping a graph or a portion of a graph such as some vertices can hang based on errors in logged/unlogged or causes error depending on logged/unlogged DSE database batches. A method to resolve the problem is to **DROP TABLE** or **TRUNCATE** the underlying DSE database tables storing graph data.

The data for a graph is stored in `<graph_name>.<vertex_label>_p` and `<graph_name>.<vertex_label>_e`. For example, recipe data stored in a graph `food` will be `food.recipe_p` and `food.recipe_e`.

In some cases, additional steps must be taken to delete a graph. A graph consists of three DSE database keyspaces: `<graph_name>`, `<graph_name>_system`, and `<graph_name>_pvt`. All three keyspaces must be deleted, with `cqlsh` if necessary, to completely delete the graph.

If a graph hangs during provisioning, use the following `cqlsh` commands:

```
cqlsh> delete from dse_system.shared_data where dataspace = 'Cluster'
and valid_until = 13814000-1dd2-11b2-0000-000000000000
and namespace = 'system' and name = '<graph_name>';
cqlsh> update dse_system.shared_data set last_updated = now() where dataspace = 'Cluster';
```

Shared data is not normally manually updated. However, this procedure can be used in the case of a node failure during a graph provisioning operation.

**Gremlin console hangs or behaves erratically**

The DataStax Enterprise Help Center also provides troubleshooting information.

The Gremlin console can quit responding if a bad query is entered. Here are some hints for how to regain control:

`:clear`

This command will clear the current buffer and reset the prompt counter.

`:remote config alias reset`

This command will reset the graph traversal `g` to no graph. Use this command before running `system` commands.

`:remote config alias g some_graph.g`

This command will reset the graph traversal `g` to another graph. If queries are hanging in the current graph, switching to another graph allows the console to respond.

**Queries sporadically fail with LOCAL_ONE/LOCAL QUORUM**

The DataStax Enterprise Help Center also provides troubleshooting information.

The DataStax Enterprise database will sporadically fail in a multi-datacenter cluster using SimpleStrategy if a query uses LOCAL_ONE or LOCAL QUORUM. This affects DSE Graph queries in a multi-datacenter cluster as well. If you use DataStax Studio to auto-create graphs in Development mode, this issue is likely to appear.

To avoid this issue, in or multi-datacenter clusters, create graphs with appropriate DSE database settings.

**Consistency level and graph.addVertex()**

The DataStax Enterprise Help Center also provides troubleshooting information.

Vertex id allocation requires QUORUM consistency to ensure that allocated ids are unique. Vertex ids are allocated in blocks. If a block runs out, write failures can occur and a consistency level of QUORUM cannot be met. Graph clusters can operate while falling below QUORUM, but it is possible for queries involving adding a vertex can fail if a consistency level of ONE is used.

Do not use a consistency level of ONE for operations that add vertex data.
Issues creating a graph cluster using Lifecycle Manager (LCM)

The DataStax Enterprise Help Center also provides troubleshooting information.

Currently, OpsCenter LCM is blocking proper setup and usage of DSE Graph. Some steps can be taken to fix the problems until changes are made in DSE 5.1.3.

The default serializers in dse.yaml file must be updated to match the following for the gremlin_server settings:

```
gremlin_server:
  maxContentLength: 65536000
  maxChunkSize: 4096000
  port: 8182
  serializers:
    - { className: org.apache.tinkerpop.gremlin.driver.ser.GryoMessageSerializerV1d0, config: { ioRegistries: [org.apache.tinkerpop.gremlin.tinkergraph.structure.TinkerIoRegistry], classResolverSupplier: com.datastax.bdp.graph.impl.tinkerpop.io.DseClassResolverProvider } }
    - { className: org.apache.tinkerpop.gremlin.driver.ser.GryoLiteMessageSerializerV1d0, config: { ioRegistries: [org.apache.tinkerpop.gremlin.tinkergraph.structure.TinkerIoRegistry], classResolverSupplier: com.datastax.bdp.graph.impl.tinkerpop.io.DseClassResolverProvider } }
    - { className: org.apache.tinkerpop.gremlin.driver.ser.GryoMessageSerializerV1d0, config: { serializeResultToString: true } }
    - { className: org.apache.tinkerpop.gremlin.driver.ser.GraphSONMessageSerializerGremlinV1d0, config: { ioRegistries: [org.apache.tinkerpop.gremlin.tinkergraph.structure.TinkerIoRegistry] } }
    - { className: org.apache.tinkerpop.gremlin.driver.ser.GraphSONMessageSerializerV1d0, config: { ioRegistries: [org.apache.tinkerpop.gremlin.tinkergraph.structure.TinkerIoRegistry] } }

scriptEngines:
  gremlin-groovy:
    config:
      compilerCustomizerProviders:

"org.apache.tinkerpop.gremlin.groovy.jsr223.customizer.ThreadInterruptCustomizerProvider": []

"org.apache.tinkerpop.gremlin.groovy.jsr223.customizer.InterpreterModeCustomizerProvider": []
```

The rpc_address setting must be set to 0.0.0.0 in the dse.yaml file.

LCM does not set the default log level to INFO. Change the setting in the logback-gremlin-server.xml file found in install_location/resources/graph/conf:

```
<appender name="SYSTEMLOG" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
    <level>INFO</level>
  </filter>
</appender>
```
To add the gremlin log reference, insert the following after the SPARK_SERVER_LOGBACK_CONF_FILE reference:

```xml
<include file="${GREMLIN_SERVER_LOGBACK_CONF_FILE}"/>
```

and the following in the root tag definition after the SparkWorkerFileAppender:

```xml
<appender-ref ref="GremlinServerFileAppender"/>
```

**Shutting down Studio Gremlin process**

The DataStax Enterprise Help Center also provides troubleshooting information.

It is possible for a server-side execution in DataStax Studio to become unresponsive. The symptom of this issue is a never-ending progress spinner in the cell where the command was executed. To fix this issue two steps should be taken.

1. Copy the cell's command to a new cell, delete the hung cell, and restart DataStax Studio.

2. Restart DataStax Enterprise, since DataStax Studio cell's status will most likely still be out of sync with DSE and the issue will not resolve.

This issue will be fixed in DataStax Studio 1.1.0.

**Dropping edge property drops edges**

The DataStax Enterprise Help Center also provides troubleshooting information.

For existing graph data created prior to DSE 5.0.5, dropping an edge property might drop the edge, depending on the method used to create the edge property.

- If the edge property was created at the same time as the edge label in previous DSE versions, dropping the edge property will drop the edge entirely.

- If the edge property is created separately from the edge label in previous DSE versions, dropping the edge property will not drop the edge.

This issue is fixed in DSE 5.0.5, but data created prior to that version will continue to drop edges in an unexpected manner.

**Adding or removing a Spark application, driver, or worker fails**

The DataStax Enterprise Help Center also provides troubleshooting information.

Adding or removing a Spark application, driver, or worker fails if it cannot be written to recovery storage.

When the Spark Master first starts it updates its transient state and begins process management tasks, like communicating with workers. Then the Master tries to store changes in the persistence engine. If that task fails, the Master is restarted and its transient state is rebuilt from recovery storage, from the last state that was successfully persisted. To avoid an inconsistent state between these stages, any registration or removal of Spark applications, drivers, or workers will fail until the Master can write the data to recovery storage.

If you encounter a failure, wait until the Master is running and has successfully written to recovery storage before adding or removing applications, drivers, or workers.

**DSE Search troubleshooting**

The DataStax Enterprise Help Center also provides troubleshooting information.
Troubleshooting DataStax Enterprise

DSE Search node start

Slow startup on nodes with large encrypted indexes

Slow startup on nodes with large encrypted indexes is resolved in 6.0.0 and later, 5.1.6 and later, and 5.0.12 and later. However, action is required to realize the performance gains.

First upgrade the nodes and then do a full reindex of all encrypted search indexes on each node in your cluster. Plan sufficient time after the upgrade is complete to reindex with deleteAll=true in a rolling fashion. For example:

```
$ dsetool reload_core keyspace_name.table_name distributed=false reindex=true deleteAll=true
```

DSE Search nodes fail to start

DSE 5.1.11 and later and DSE 6.0.3 and later refuse to start when required Tomcat files are not present.

When the `tomcat/conf` directory is missing, a WARN message like this appears in the system.log:

```
java.io.FileNotFoundException: /usr/local/dse/dse-5.1.9/resources/cassandra/lib/apache-cassandra-thrift-3.11.1.2261.jar (No such file or directory)
at java.util.zip.ZipFile.open(Native Method) ~[na:1.8.0_181]
at java.util.zip.ZipFile.<init>(ZipFile.java:225) ~[na:1.8.0_181]
at java.util.jar.JarFile.<init>(JarFile.java:155) ~[na:1.8.0_181]
at java.util.jar.JarFile.<init>(JarFile.java:166) ~[na:1.8.0_181]
at org.apache.tomcat.util.scan.JarFileUrlJar.<init>(JarFileUrlJar.java:60)
    ~[tomcat-util-scan-8.0.47.jar:8.0.47]
at org.apache.tomcat.util.scan.JarFactory.newInstance(JarFactory.java:49)
    ~[tomcat-util-scan-8.0.47.jar:8.0.47]
    ~[tomcat-util-scan-8.0.47.jar:8.0.47]
at org.apache.tomcat.util.scan.StandardJarScanner.scan(StandardJarScanner.java:284)
    ~[tomcat-util-scan-8.0.47.jar:8.0.47]
at org.apache.catalina.startup.ContextConfig.processJarsForWebFragments(ContextConfig.java:1898)
    ~[tomcat-embed-core-8.0.47.jar:8.0.47]
at org.apache.catalina.startup.ContextConfig.webConfig(ContextConfig.java:1131)
    ~[tomcat-embed-core-8.0.47.jar:8.0.47]
at org.apache.catalina.startup.ContextConfig.configureStart(ContextConfig.java:783)
    ~[tomcat-embed-core-8.0.47.jar:8.0.47]
at org.apache.catalina.startup.ContextConfig.lifecycleEvent(ContextConfig.java:307)
    ~[tomcat-embed-core-8.0.47.jar:8.0.47]
at org.apache.catalina.util.LifecycleBase.fireLifecycleEvent(LifecycleBase.java:95)
    ~[tomcat-embed-core-8.0.47.jar:8.0.47]
at org.apache.catalina.util.LifecycleBase.fireLifecycleEvent(LifecycleBase.java:90)
    ~[tomcat-embed-core-8.0.47.jar:8.0.47]
at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5213)
    ~[tomcat-embed-core-8.0.47.jar:8.0.47]
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:145)
    ~[tomcat-embed-core-8.0.47.jar:8.0.47]
at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:753)
    ~[tomcat-embed-core-8.0.47.jar:8.0.47]
```
Followed by reference to the missing locale Tomcat JAR files:

```
- Failed to scan [file:/usr/local/dse/dse-5.1.9/resources/spark/lib/derbyLocale_cs.jar] from classloader hierarchy
java.io.FileNotFoundException: /usr/local/dse/dse-5.1.9/resources/spark/lib/derbyLocale_cs.jar (No such file or directory)
  at java.util.zip.ZipFile.open(Native Method) ~[na:1.8.0_181]
  at java.util.zip.ZipFile.<init>(ZipFile.java:225) ~[na:1.8.0_181]
  at java.util.zip.ZipFile.<init>(ZipFile.java:155) ~[na:1.8.0_181]
  at java.util.jar.JarFile.<init>(JarFile.java:166) ~[na:1.8.0_181]
  at java.util.jar.JarFile.<init>(JarFile.java:130) ~[na:1.8.0_181]
  at org.apache.tomcat.util.scan.JarFileUrlJar.<init>(JarFileUrlJar.java:60) ~[tomcat-util-scan-8.0.47.jar:8.0.47]
  at org.apache.tomcat.util.scan.JarFactory.newInstance(JarFactory.java:49) ~[tomcat-util-scan-8.0.47.jar:8.0.47]
  at org.apache.tomcat.util.scan.StandardJarScanner.process(StandardJarScanner.java:334) ~[tomcat-util-scan-8.0.47.jar:8.0.47]
  at org.apache.tomcat.util.scan.StandardJarScanner.scan(StandardJarScanner.java:284) ~[tomcat-util-scan-8.0.47.jar:8.0.47]
  at org.apache.catalina.startup.ContextConfig.processJarsForWebFragments(ContextConfig.java:1898) ~[tomcat-embed-core-8.0.47.jar:8.0.47]
  at org.apache.catalina.startup.ContextConfig.webConfig(ContextConfig.java:1131) ~[tomcat-embed-core-8.0.47.jar:8.0.47]
  at org.apache.catalina.startup.ContextConfig.configureStart(ContextConfig.java:783) ~[tomcat-embed-core-8.0.47.jar:8.0.47]
  at org.apache.catalina.startup.ContextConfig.lifecycleEvent(ContextConfig.java:307) ~[tomcat-embed-core-8.0.47.jar:8.0.47]
```
To resolve:

- Ensure that the `catalina.properties` and `context.xml` files are present in the Tomcat `conf` dir. DSE will not start after upgrade if these files are missing.

The default location of the Tomcat `conf` directory depends on the type of installation:

- **Package installations:** `/etc/dse/tomcat/conf`
- **Tarball installations:** `installation_location/resources/tomcat/conf`

### Handling inconsistencies in query results

The [DataStax Enterprise Help Center](https://datastax-enterprise.com) also provides troubleshooting information.

To troubleshoot inconsistencies in query results, consider session stickiness, subrange node repair, and follow best practices for soft commit points on different replica nodes.

DSE Search implements an efficient, highly available distributed search algorithm on top of the database, which tries to select the minimum number of replica nodes required to cover all token ranges, and also avoid hot spots. Consequently, due to the eventually consistent nature of the database, some replica nodes might not have received or might not have indexed the latest updates yet. This situation might cause DSE Search to return inconsistent results (different numFound counts) between queries due to different replica node selections.
This behavior is intrinsic to how highly available distributed systems work, as described in the ACM article, “Eventually Consistent” by Werner Vogels. Most of the time, eventual consistency is not an issue, yet DSE Search implements session stickiness to guarantee that consecutive queries will hit the same set of nodes on a healthy, stable cluster, to provide monotonic results. Session stickiness works by adding a session seed to request parameters as follows:

```
shard.shuffling.strategy=SEED
shard.shuffling.seed=session_id
```

In the event of unstable clusters with missed updates due to failures or network partitions, consistent results can be achieved by repairing nodes using the DSE OpsCenter Repair Service.

Finally, another minor source of inconsistencies is caused by different soft commit points on different replica nodes: A given item might be indexed and committed on a given node, but not yet on its replica. This situation is primarily a function of the load on each node. Implement the following best practices:

- Evenly balance read/write load between nodes
- Properly tune soft commit time and async indexing concurrency
- Configure back pressure in the dse.yaml file

For information about multi-threaded asynchronous indexing that uses a back pressure mechanism, see Configuring and tuning indexing performance.

To maximize insert throughput, DSE Search buffers insert requests from the database so that application insert requests can be acknowledged as quickly as possible. However, if too many requests accumulate in the buffer (a configurable setting), DSE Search pauses or blocks incoming requests until DSE Search catches up with the buffered requests. In extreme cases, that pause causes a timeout to the application.

### Troubleshooting Solr HTTP requests

The DataStax Enterprise Help Center also provides troubleshooting information.

For debugging and troubleshooting queries, you can trace Solr HTTP requests in one of the following ways:

- Enable probabilistic tracing.
- Pass an explicit cassandra.trace=true request parameter in the HTTP query.

After running the example of using a join query, you can trace the join query by adding the cassandra.trace parameter to the HTTP request:

```
http://localhost:8983/solr/internet.songs/select/?q=(!join+from=song+to=id
+fromIndex=internet.lyrics+force=true)words:love&indent=true&wt=json&cassandra.trace=true
```

The Solr response includes a cassandra.trace.session value, the unique session id of the tracing session for the request:

```
{
  "cassandra.trace.session":"3e503490-bdb9-11e3-860f-73ded3cb6170",
  "responseHeader":{
    "status":0,
    "QTime":1,
    "params":{
      "indent":true,
      "q":"(!join from=song to=id fromIndex=internet.lyrics force=true)words:love",
      "wt":json",
      "cassandra.trace":true
    }
  },
  "response":{"numFound":2,"start":0,"docs":[
    {
      "id":"8a172618-b121-4136-bb10-f665cfc469eb",
      "title":"Internet Love Song",
      "artist":"John Cedrick"
    }
  ]
```

---

Troubleshooting Page 19
To see the information from the trace, query the system_traces.events, using the session id to filter the output.

cqlsh> select * from system_traces.events where session_id = 3e503490-bdb9-11e3-860f-73ded3cb6170;

<table>
<thead>
<tr>
<th>session_id</th>
<th>activity</th>
<th>source elapsed</th>
</tr>
</thead>
<tbody>
<tr>
<td>3e503490...</td>
<td>Parsing SELECT * from &quot;internet&quot;.&quot;songs&quot; WHERE &quot;id&quot; = 8a172618...</td>
<td>2607</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Preparing statement</td>
<td>3943</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Executing single-partition query on songs</td>
<td>4246</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Acquiring sstable references</td>
<td>4261</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Merging memtable tombstones</td>
<td>4305</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Key cache hit for sstable 1</td>
<td>4388</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Seeking to partition indexed section in data file</td>
<td>4399</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Skipped 0/1 non-slice-intersecting sstables, included 0 due to tombstones</td>
<td>4873</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Merging data from memtables and 1 sstables</td>
<td>4954</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Read 1 live and 0 tombstoned cells</td>
<td>5162</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Parsing SELECT * from &quot;internet&quot;.&quot;songs&quot; WHERE &quot;id&quot; = a3e64f8f...</td>
<td>6160</td>
</tr>
<tr>
<td>3e503490...</td>
<td>Preparing statement</td>
<td>7424</td>
</tr>
</tbody>
</table>

For example purposes, the event_id, node IP address, and thread id have been deleted from this output to fit on the page.

In the case of distributed queries over several nodes, the same tracing session id is used on all nodes, which makes it possible to correlate database operations on all the nodes taking part in the distributed query.

### Troubleshooting security

The [DataStax Enterprise Help Center](https://datastax.com) also provides troubleshooting information.

#### DSE fails to start

The [DataStax Enterprise Help Center](https://datastax.com) also provides troubleshooting information.

DSE verifies authentication services during start up. When a service is not available, such as Kerberos or LDAP, that has been set as a default_scheme or other_schemes in the authentication_options, the following error appears in the log:

```
ERROR [main] 2017-03-08 17:03:33,482  DseModule.java:97 - {}. Exiting...
```

com.google.inject.CreationException: Unable to create injector, see the following errors:
1) An exception was caught and reported. Message: The dse service keytab at this location resources/dse/conf/dse.keytab either doesn't exist or cannot be read by the dse service at com.datastax.bdp.DseModule.configure(Unknown Source)

1 error

To allow DSE to properly start either ensure that the service is available or remove it from the configuration.

SSL certificate doesn’t match

The Common Name (CN) that is used to generate the SSL certificate must match the DNS resolvable host name. Mismatches between the CN and node hostname cause an exception and the connection is refused. With dsetool and other tools that issue commands to the cluster, error messages indicate that the environment is not configured correctly. For example:

```
dsetool -h 10.236.136.55 reload_core keyspace_name.table_name deleteAll=true reindex=true distributed=false ...
javax.net.ssl.SSLException: Certificate for <node35.foo.com> doesn't match any of the subject alternative names: clustercert.foo.com ...
```

This message shows that 10.236.136.55 is being resolved to node35.foo.com, then node node35.foo.com is being asked for it's certificate which is a generic certificate issued with a CN of clustercert.foo.com. For security reasons, SSL verifies that node35.foo.com and clustercert.foo.com match. If they don’t match, a certificate mismatch error occurs. Do not use a generic certificate across multiple nodes, because each node has a different name that won't match. Do not copy a certificate that is issued to node35.foo.com over to another node.

Nodes must be configured with correct names that match the certificate CN. You can use a wildcard in named certificates, like * in CN=*.foo.com, or any other matching mechanism allowed by SSL standards. All others configurations where names mismatch will result in an error.

When testing connections with other tools, enable them with secure settings. Avoid testing with insecure settings that do not require name matching. For example, `curl --insecure`. These insecure settings do not identify certificate mismatches and are not supported in DSE tools.

SSL exceptions occur on start up or no connections

Follow these steps to troubleshoot SSL connections when exceptions occur on start up, or no connections to the database can be established.

- To enable debugging, add the following option to cassandra-env.sh:

  ```
  -Djavax.net.debug=ssl
  ```

  Detail startup and connection messages are printed to STDOUT, including SSL handshake errors. See Debugging SSL/TLS Connections for message details.

- Verify SSL encryption messages in /var/log/cassandra/system.log.

  # SSL starts properly:
For client-to-server connections:


For node-to-node messaging:


SSL fails to start:

# Truststore or keystore file not found:

Caused by: org.apache.cassandra.exceptions.ConfigurationException: Failed to initialize SSL

...  
Caused by: java.io.FileNotFoundException: resources/dse/conf/.truststore (No such file or directory)

Example shows when the default setting was not changed for node-to-node.

# Truststore or keystore password is invalid:

Caused by: java.io.IOException: Error creating the initializing the SSL Context

at org.apache.cassandra.security.SSLFactory.createSSLContext(SSLFactory.java:201)

at com.datastax.bdp.node.transport.SSLOptions.getDefault(SSLOptions.java:82)
...  
Caused by: java.io.IOException: Keystore was tampered with, or password was incorrect

Example show when password of keystore for node-to-node is incorrect.

Connection errors with cqlsh and other DSE tools

If authentication_options are enabled (true) in the dse.yaml, credentials are required to launch cqlsh and other tools:

```
cqlsh -u username -p password
```

If a client connection is attempted without permissions, the following error occurs:

```
Connection error: ('Unable to connect to any servers', {'127.0.0.1': error(111, "Tried connecting to ['127.0.0.1', 9042]). Last error: Connection refused")})
```

See Providing credentials with cqlsh and SSL certificate doesn't match.
Troubleshooting DataStax Studio

The DataStax Enterprise Help Center also provides troubleshooting information.

**Studio will not start with wrong Java version**

Studio checks the JAVA_HOME environment variable at startup. To ensure that Studio starts with the correct version of Java, make sure the JAVA_HOME environment variable points to the installation directory with the supported version of Java. See Supported platforms. Note that even when the `java -version` command shows a supported Java version, it is possible that JAVA_HOME environment variable is not set to the correct Java installation directory.

**NOT ENABLED error: AlwaysOn SQL cell cannot be executed**

The AlwaysOn SQL cell cannot be executed when the connection is to a DSE cluster that is not correctly configured for the AlwaysOn SQL service.

In order to run AlwaysOn SQL, you must have:

- A running datacenter with DSE Analytics nodes enabled.
- Enabled AlwaysOn SQL on every Analytics node in the datacenter.
- Modify the replication factor for all Analytics nodes, if necessary.
- Set the `native_transport_address` in cassandra.yaml to an IP address that is accessible by the AlwaysOn SQL clients. This address depends on your network topology and deployment scenario.
- Configured AlwaysOn SQL for security, if authentication is enabled.

**STOPPED error: AlwaysOn SQL cell cannot be executed**

The AlwaysOn SQL cell cannot be executed when the AlwaysOn SQL service is stopped on the DSE node. See Checking the status of AlwaysOn SQL.

**HTTP protocol**

DataStax Studio does not run with HTTPS everywhere. Preface the URL used to run DataStax Studio with HTTP and not HTTPS.

**Studio won't accept connections from external addresses**

By default, Studio binds itself to the IP address 127.0.0.1 (or localhost). If running in the cloud, this IP address prevents Studio from accepting connections from external addresses. If you have already installed Studio in the cloud using localhost, change the `httpBindAddress` value in the `configuration.yaml` Studio configuration file to the IP address of the host it is running on.

Changing the `httpBindAddress` setting from the default (localhost) can pose a security risk as users on external machines can gain access to notebooks and the DSE clusters those notebooks are connected to. Studio is designed to be used as a desktop application. Distributed deployment introduces potential security risks.

**Unable to connect Studio to a secured DSE cluster**

If you are unable to connect Studio when client-to-node encryption enabled on a DSE cluster, the Studio log has this entry:

```plaintext
Cannot support TLS_RSA_WITH_AES_256_CBC_SHA with currently installed providers
```

Java Cryptography Extension (JCE) Unlimited Strength Policy files is required to ensure support for all encryption algorithms. See Using SSL connections in DataStax Studio.

**Content assist not working**

On MacOS 10.12, a network configuration issue when running Studio causes content assist functionality to become slow or unresponsive. For troubleshooting steps to resolve this networking issue, see this Stack Overflow post.
Notebook needs troubleshooting

When you export a notebook, you can specify actions for the notebook. As appropriate, select to include cell code and results or cell code only. For either of these actions, you can also select to include diagnostics. Exporting a notebook with diagnostics might include sensitive information. Credentials are not included in the export.

Studio UI behaves strangely after upgrade

If you leave an old Studio session open in a browser, using an older version of Studio, and then upgrade to a new version, you can experience strange behavior. This is caused by caching in the browser. You can solve this problem by force-loading the index.html page. For example, if using the Chrome browser, you can navigate to the top-level index.html page and click on the refresh button while holding down the control key.

Starting Studio on Windows 10

While running Studio on the Microsoft Windows 10 operating system, out-of-date libraries can prevent system-specific JNI libraries from being loaded correctly. The result is the following stack trace:

```
  java.util.NoSuchElementException: head of empty list
    at scala.collection.immutable.Nil$.head(List.scala:420) ~[scala-library-2.11.8.jar:?]
    at scala.collection.immutable.Nil$.head(List.scala:417) ~[scala-library-2.11.8.jar:?]
    at akka.actor.ActorCell.receiveMessage(ActorCell.scala:526) ~[akka-actor_2.11-2.4.17.jar:?
    at akka.actor.ActorCell.invoke(ActorCell.scala:495) ~[akka-actor_2.11-2.4.17.jar:?
    at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:257) ~[akka-actor_2.11-2.4.17.jar:?
    at akka.dispatch.Mailbox.run(Mailbox.scala:224) ~[akka-actor_2.11-2.4.17.jar:?
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149
```

You can fix this problem by updating the Microsoft Visual C++ 2010 Redistributable Package libraries in question:

- 32-bit version
- 64-bit version

Starting Studio on Windows 7

While running Studio on the Microsoft Windows 7 operating system, out-of-date libraries can prevent system-specific JNI libraries from being loaded correctly. The result is the following stack trace:

```
leveldbjni-64-1-603687032598847314.8: Can't find dependent libraries]
  at org.fusesource.hawtjni.runtime.Library.doLoad(Library.java:182)
  at org.fusesource.hawtjni.runtime.Library.load(Library.java:140)
  at org.fusesource.leveldbjni.JniDBFactory.<clinit>(JniDBFactory.java:48)
  at com.rbmhtechnology.eventuate.log.leveldb.LeveldbEventLog.<init>(LeveldbEventLog.scala:97)
  at com.rbmhtechnology.eventuate.log.leveldb.LeveldbEventLog$.apply(LeveldbEventLog.scala:358)
  at akka.actor.TypedCreatorFunctionConsumer.produce(IndirectActorProducer.scala:87)
```

You can fix this problem by updating the Microsoft Visual C++ 2010 Redistributable Package libraries in question:

You can fix this problem by updating the Microsoft Visual C++ 2010 Redistributable Package libraries in question:

- 32-bit version
- 64-bit version
Other troubleshooting

The DataStax Enterprise Help Center also provides troubleshooting information.

View of ring differs between some nodes

The DataStax Enterprise Help Center also provides troubleshooting information.

Indicates that the ring is in a bad state.

This situation can happen when not using virtual nodes (vnodes) and token conflicts exist. You might see this when bootstrapping two nodes simultaneously with automatic token selection. Unfortunately, the only way to resolve this is to do a full cluster restart. A rolling restart is insufficient since gossip from nodes with the bad state will repopulate it on newly booted nodes.

SSTables imported using sstableloader are missing rows

The DataStax Enterprise Help Center also provides troubleshooting information.

The source SSTables used by sstableloader (bulk loader) may not have included all the table data.

The sstableloader streams data from SSTables on disk to a cluster.

Before importing existing SSTables, run nodetool flush on each source node to assure that any data in memtables is written to the SSTables on disk.

For more information and instructions, see sstableloader.

Prepared statements discarded

The DataStax Enterprise Help Center also provides troubleshooting information.

If you repeatedly see messages about prepared statements being discarded because cache limit have been reached:

1. Investigate the root cause of these messages and check whether prepared statements are used correctly by using bind markers for variable parts.

2. Only change the default value of prepared_statements_cache_size_mb when there are more prepared statements than fit in the cache. In most cases it is not necessary to change this value.

Purging gossip state on a node

The DataStax Enterprise Help Center also provides troubleshooting information.

Gossip information is persisted locally by each node to use immediately on node restart without having to wait for gossip communications.

In the unlikely event you need to correct a problem in the gossip state:

1. Use the nodetool assassinate to shut down the problem node.
   
   This takes approximately 35 seconds to complete, so wait for confirmation that the node is deleted.

2. If this method doesn’t solve the problem, stop your client application from sending writes to the cluster.

3. Take the entire cluster offline:

   a. Drain each node.

      $ nodetool options drain

   b. Stop each node.
4. Clear the data from the peers directory, remove all directories in the peers-UUID directory, where UUID is the particular directory that corresponds to the appropriate node:

```
$ sudo rm -r /var/lib/cassandra/data/system/peers-UUID/*
```

Use caution when performing this step. The action clears internal system data from the database and may cause application outage without careful execution and validation of the results. To validate the results, run the following query individually on each node to confirm that all of the nodes are able to see all other nodes.

```
select * from system.peers;
```

5. Clear the gossip state when the node starts:

- For tarball or Installer No-Services installations, use a command line option or edit the cassandra-env.sh. To use the command line:

```
$ installation_location/bin/cassandra -Dcassandra.load_ring_state=false
```

- For package or Installer Services installations or if you are not using the command line option above, add the following line to the cassandra-env.sh file:

```
JVM_OPTS="$JVM_OPTS -Dcassandra.load_ring_state=false"
```

6. Bring the cluster online one node at a time, starting with the seed nodes.

   See Starting and stopping DataStax Enterprise.

What’s next:

Remove the line you added in the cassandra-env.sh file.

Recovering expired data caused by TTL year 2038 problem

Prior to DataStax Enterprise version 5.1.7 in the 5.1.X series and 5.0.12 in the 5.0.X series, there was no protection against INSERTS if the TTL expiration timestamp was after the maximum date that the storage engine could represent \(2038-01-19T03:14:06+00:00\). Before 5.1.7 or 5.0.12, if an expiration timestamps with a later date was inserted, the date calculation overflowed causing the data to expire immediately. Records expired by overflow are not queryable and are permanently removed after a compaction. This issue occurs only for INSERTs that have a long TTL value that is close to the maximum 630720000 seconds (20 years). The earliest possible date overflow for expiration timestamps is \(2018-01-19T03:14:06+00:00\). As time progresses, the maximum supported TTL gradually reduces as the date \(2038-01-19T03:14:06+00:00\) approaches.

To recover data with overflowed timestamps from SSTables that were backed up or that did not go through compaction, use the `--reinsert-overflowed-ttl` option, because tombstones might have been generated with the original timestamp.

To find out if an SSTable has an entry with overflowed expiration, inspect it with the `sstablemetadata` tool and look for a negative `min local deletion time` field. Back up SSTables in this condition immediately, as they are subject to data loss during compaction.

Use only one of the following options.

**Offline scrub**
For **offline scrub**, use the `sstablescrub` command with the `--reinsert-overflowed-ttl` parameter to recover data from a backed up SSTable.

<table>
<thead>
<tr>
<th>DSE version</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSE 6.7</td>
<td><code>sstablescrub --reinsert-overflowed-ttl</code></td>
</tr>
<tr>
<td>DSE 6.0</td>
<td><code>sstablescrub --reinsert-overflowed-ttl</code></td>
</tr>
<tr>
<td>DSE 5.1.7 and later</td>
<td><code>sstablescrub --reinsert-overflowed-ttl</code></td>
</tr>
<tr>
<td>DSE 5.0.12 and later</td>
<td><code>sstablescrub --reinsert-overflowed-ttl</code></td>
</tr>
<tr>
<td>DSE 4.8.16</td>
<td><code>sstablescrub --reinsert-overflowed-ttl</code></td>
</tr>
</tbody>
</table>

**Online scrub**

For **online scrub**, use the `nodetool scrub` command with the `--reinsert-overflowed-ttl` parameter to recover data from a table that has not gone through compaction.

<table>
<thead>
<tr>
<th>DSE version</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSE 6.7</td>
<td><code>nodetool --reinsert-overflowed-ttl</code></td>
</tr>
<tr>
<td>DSE 6.0</td>
<td><code>nodetool --reinsert-overflowed-ttl</code></td>
</tr>
<tr>
<td>DSE 5.1.7 and later</td>
<td><code>nodetool --reinsert-overflowed-ttl</code></td>
</tr>
<tr>
<td>DSE 5.0.12 and later</td>
<td><code>nodetool --reinsert-overflowed-ttl</code></td>
</tr>
<tr>
<td>DSE 4.8.16</td>
<td><code>nodetool --reinsert-overflowed-ttl</code></td>
</tr>
</tbody>
</table>

- Use the offline scrub option to recover data from a backed up SSTable:
  1. Clone the data directory tree to another location. Keep only the folders and the contents of the system tables.
  2. Clone the configuration directory to another location. Set the `data_file_directories` property to the cloned data directory in the cloned `cassandra.yaml`.
  3. Copy the affected SSTables to the cloned data location of the affected table.
  4. Set the environment variable `CASSANDRA_CONF=cloned_configuration_directory`.
  5. Run the following command to update the table.

```bash
$ sstablescrub --reinsert-overflowed-ttl keyspace_name table_name
```

  6. After the scrub is completed, copy the resulting SSTables to the original data directory.
  7. Use `nodetool refresh` on a live node to load the recovered SSTables.

- Use the online scrub option to recover data from a table that has not gone through compaction:
  1. Disable compaction on the node.

```bash
$ nodetool disableautocompaction
```

This step is crucial. The data might be removed permanently during compaction.
2. Copy the SSTables containing entries with overflowed expiration time to the data directory.

3. Load the SSTables.

   ```
   $ nodetool refresh
   ```

4. Run the scrub command with reinsert overflow option on the tables.

   ```
   $ nodetool scrub --reinsert-overflowed-ttl keyspace_name table_name
   ```

5. For indexed tables, use the `dsetool reload_core` command on a search node to load and reindex the reinserted values:

   ```
   $ dsetool reload_core reindex=true keyspace_name.table_name
   ```

6. Re-enable compactions after verifying that scrub recovered the missing entries.
Chapter 2. Troubleshooting DSE OpsCenter

Use the following pages to locate solutions or workarounds for issues experienced with OpsCenter.

When reporting issues to DataStax Support, refer to the following files in the /usr/share/doc/opscenter directory:

- ds_branch.txt
- ds_timestamp.txt
- ds_version.txt

These files contain build and version information that is useful when troubleshooting issues. Run the following commands to display all information from these files:

```
$ cd /usr/share/doc/opscenter
$ cat ds_*.txt
```

```
DSBranch=6.7.1
DSCommit=c6512b39fa9c8ca8adcf1a6edc79d2249c9e9947
DSTIMESTAMP=20190215112713+0000
DSVersion=6.7.1
```

Troubleshooting SSL

Configuring client-to-node encryption

In all supported versions, OpsCenter SSL connections to DataStax Enterprise (DSE) clusters are validated automatically. OpsCenter uses a keystore/truststore model to connect to DSE, which requires stricter enforcement of SSL certificates. Therefore, there is no longer an option to disable SSL certificate validation using the `ssl_validate` option in cluster settings.

To ensure that OpsCenter is configured to work with your client-to-node encryption-enabled cluster, follow these steps:

1. Create a truststore file on the OpsCenter machine using each DSE node’s certificates.
   
   a. Obtain the public key certificates from each DSE node you want to connect to OpsCenter.
b. Import these public key certificates into the truststore file. For example:

```
$ keytool -import -v -trustcacerts -alias node0 -file node0.cer -keystore .truststore
```

2. Create a client certificate and keystore on the OpsCenter machine:

```
$ keytool -genkey -alias opscenter -keystore keystore.jks
$ keytool -export -alias opscenter -file opscenter.cer -keystore keystore.jks
```

3. If your DSE cluster is set up to require client authentication (require_client_auth under client-to-node-encryption in cassandra.yaml) when using client-to-node encryption, you will need to import the public certificate generated in step 2 into every node’s truststore so that DSE will trust connections coming from OpsCenter.

Troubleshooting connections with OpsCenter to DSE with SSL

The following errors can indicate issues where OpsCenter is connecting to DataStax Enterprise (DSE) with SSL.

**OpsCenter shuts down due to invalid property in cluster config: ssl_validate**

This error occurs when your cluster configuration contains the deprecated ssl_validate configuration value in your cluster_name.conf configuration file. Remove the ssl_validate option and **Restart OpsCenter**.

**OpsCenter cannot connect to the cluster with No DSE connection available error in logs**

While this error can be due to a number of issues with the cluster connection, when working with SSL, it can be an indicator that the keystore/truststore setup needs tweaking. You might see errors similar to these in the logs:

```
2016-02-04 16:06:53,255 [] DEBUG: Node 127.0.0.1 seems to be down, trying next contact point (MainThread)
2016-02-04 16:06:53,255 [] DEBUG: Unable to connect to any seed nodes, tried ['127.0.0.1'] (MainThread)
2016-02-04 16:06:53,256 [] WARN: No cassandra connection available for hostlist ['127.0.0.1'] . Retrying. (MainThread)
```

These errors indicate that OpsCenter cannot make a connection with the DataStax Enterprise (DSE) cluster. To ensure that your SSL configuration is correct, check the following:

- Certificates in keystore/truststore are valid (hostnames match and certificates are not expired).
- All public certificates for each node are in the truststore for OpsCenter.
- The certificate for OpsCenter is in each node’s truststore if require_client_auth is enabled.

**Browser issues**

**Zero nodes detected in cluster or Loading OpsCenter screen hanging**

Some environments might experience connectivity issues with the persistent connection between the browser and opscneterd. Symptoms of this issue include a blinking icon near the top right of the OpsCenter UI, and “0 nodes” displays as well. Another symptom is the **Loading OpsCenter** screen seems frozen, and does not load OpsCenter.

Follow the workaround steps to resolve the issue:

1. Open opscenterd.conf and add a [labs] section.
2. Add the `orbited_longpoll` option and set the value to `true`:

```
[ labs ]
orbited_longpoll = true
```

3. Restart `opscenterd`.

4. Refresh the browser.

**Microsoft web browsers not supported**

If you try to load the OpsCenter client in Microsoft Internet Explorer (IE) or Microsoft Edge, a dialog indicates that the browser is not supported.

OpsCenter is only supported on the latest versions of:

- Apple Safari
- Google Chrome
- Mozilla Firefox

**Troubleshooting SSTables**

**SSTables not cleaned up after failed restore**

If a restore that bypasses SSTableloader fails, some SSTables might be copied over, whereas other SSTables are not. In this scenario, OpsCenter does not clean up SSTables that were copied or revert any changes in the Cassandra data directory. This behavior could cause the keyspace or table to be unusable, and possibly result in duplicate, inconsistent, and even lost data.

This scenario only applies to restores that do not use SSTableloader. If SSTableloader is used with the restore, the copied files are staged in a temporary directory that OpsCenter cleans up.

Retry the restore and view the Restore Report to track the progress and status of the restore. The Restore Report dialog can be closed at any time without impacting the restore process. Reopen the report by clicking on the In Progress restore in the Activity tab. If the restore completes successfully, all SSTables are copied over.

**The SSTables in this snapshot `<tag>` are not compatible**

If you receive an error message that “The SSTables in this snapshot `<tag>` are not compatible with the current version of Cassandra”, you must upgrade your snapshot to the current major version of DataStax Enterprise (DSE).

1. Log in to each node.

2. Run the `sstableupgrade` script for every keyspace and table you need to restore by passing it the keyspace, table, and OpsCenter snapshot tag received from the error message.

   - **Location of the script:**
     - Package installation: `/usr/bin/`
     - Tarball installation: `installation_location/resources/cassandra/bin`

   See `sstableupgrade` for more information.

3. Retry the restore operation using OpsCenter.
Other troubleshooting

The DSE OpsCenter Help Center also provides troubleshooting information.

DSE version not supported by LCM

When attempting to edit a Config Profile in Lifecycle Manager (LCM), the following error message might display:

<table>
<thead>
<tr>
<th>Unsupported DSE Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSE 6.1.2 is not supported by this version of LCM</td>
</tr>
</tbody>
</table>

LCM stores metadata about DataStax Enterprise (DSE) configuration parameters in definition files. These errors indicate that the specified version of DSE is not listed in the definition files available to OpsCenter and LCM. This situation can occur for the following reasons:

Upgrading OpsCenter

If the version of DSE is End Of Life (EOL) or End Of Service Life (EOSL), and you completed a major release upgrade of OpsCenter, then OpsCenter cannot access definition files for that DSE version. Check the supported software page for supported DSE versions. If your DSE version is unsupported, upgrade DSE to a supported version or downgrade OpsCenter to a version that supports the currently installed DSE version.

Updating definition files offline

By default, OpsCenter attempts to download definition files automatically from DataStax. If network connectivity or security policies prevent this outbound HTTPS connection from completing, definitions cannot be updated automatically. The result is that newly released DSE versions will not be supported by OpsCenter and LCM. Because DataStax cannot control the order and timing in which these updates are applied, it can lead to the following scenarios:

Manually updating definition files improperly can cause this error to display. Consider the following example:

1. OpsCenter administrator manually updates OpsCenter definition files. These definition files include support for DSE 6.1.2, which was released more recently than the installed OpsCenter version.
2. OpsCenter user creates a configuration profile in LCM for DSE 6.1.2.
3. OpsCenter administrator upgrades OpsCenter to a newer version, but one which still predates DSE 6.1.2. The definition files bundled with this version of OpsCenter do not include DSE 6.1.2 support.
4. OpsCenter user attempts to edit the previously created configuration profile, or run a job on a cluster with that profile applied. LCM cannot access metadata associated with DSE 6.1.2 because the definition files are not included, which leads to errors.

In this scenario, manually updating definition files again should restore the ability for LCM to view, edit, and run jobs associated with the DSE 6.1.2 configuration profile.

High CPU usage by opscenterd

Increasing the nodelist polling period or setting a sleep delay can reduce excessive CPU usage when running opscenterd. In some environments, you might notice CPU usage for the opscenterd spiking dramatically (almost to 100%) upon startup or while already running. Typically, this spike is caused by the retrieval and parsing of cluster topology performed during startup, and every 60 seconds by default while opscenterd is running. When OpsCenter is managing multiple clusters with vnodes enabled, the impact of this CPU spike can cause performance issues or even stop opscenterd from properly starting.

If your environment is experiencing excessive CPU consumption, try the available workarounds to alleviate the issue.

Configuring the polling period for CPU issues while running opscenterd

Increasing the nodelist polling period (nodelist_poll_period) can reduce CPU usage when running opscenterd.

1. Open cluster_name.conf for editing.
2. Add a [collection] section and set the nodelist_poll_period value:

```
[collection]
nodelist_poll_period = 43200 # this would be every 12 hours
```

The nodelist_poll_period represents the interval in seconds that OpsCenter polls the nodes and
token lists in a cluster. Polling the node list determines whether there were any topology changes since
the last poll. If you do not anticipate any topology changes, set it to a high value.

3. If there were any topology changes and the polling interval is set to a high value, restart opscenterd.
Otherwise, wait for the next poll.

4. Refresh the browser.

**Configuring a sleep delay for CPU issues when starting opscenterd**

Configuring a delay between starting clusters helps alleviate opscenterd CPU usage on startup, allowing
OpsCenter to function properly. Set the startup sleep time to control how long OpsCenter waits between
connecting to clusters on startup.

1. Open opscenterd.conf for editing.

2. Under the [clusters] section, set the value of startup_sleep to 5:

```
[clusters]
startup_sleep = 5
```

The default value is 0 seconds, which results in no wait time between connecting to each cluster.
Depending on your environment, you might need to adjust the value. After configuring the sleep value
to a value other than zero, wait until all clusters have started before using the OpsCenter UI or API.
Otherwise, OpsCenter can become unresponsive and log multiple errors.

3. Restart opscenterd.

**Troubleshooting collecting or displaying metrics**

If a cluster keyspace has many tables, the number of metrics OpsCenter collects can become very large. For
information about reducing the number of keyspaces or tables that are monitored, see Controlling data collection.

**Restoring materialized views**

When restoring backups on clusters running DataStax Enterprise (DSE) 6.0.0-6.0.2, materialized views might
return no results when queried. To mitigate this issue, choose one of the following options and follow the
instructions.

**Running a second restore**

One option of restoring materialized views is to run a second restore operation with SSTableloader and
truncation enabled.

Running a second restore means running SSTableloader, which impacts performance and can cause slow downs. If performance loss is an issue and you are restoring only a few materialized views, consider
Recreating materialized views.

The initial restore can be completed without using SSTableloader. However, SSTableloader must be selected
when completing the second restore.

**Recreating materialized views**

Another option of restoring materialized views is to describe the materialized view, copy the output, drop the
materialized view, and then recreate it. This option is less involved than running a second restore, but could be
time consuming if restoring a significant number of materialized views.
Troubleshooting DSE OpsCenter

Complete the following steps to recreate the materialized views:

1. Complete a standard restore operation.

2. For each materialized view, describe the data and copy the output.

   DESCRIBE MATERIALIZED VIEW

3. Drop each materialized view.

   DROP MATERIALIZED VIEW

4. Use the copied output to recreate each materialized view.

   CREATE MATERIALIZED VIEW

DataStax Agent port setting conflict

If there are problems with OpsCenter, check for conflicts in port settings. The DataStax Agent uses port 7199 by default. If you have not changed the default port, check that Cassandra or another process on the node is not set up to use port 7199.

If you set the DataStax Agent port to a host name instead of an IP address, the DNS provider must be online to resolve the host name. If the DNS provider is not online, expect some intermittent problems.

Java not installed or JAVA_HOME environment variable not set

If Java is not installed, or if OpsCenter cannot find JAVA_HOME, the following error might display:

```bash
/usr/share/datastax-agent/bin/datastax-agent: line 98:exec: -X: invalid option
exec: usage: exec [-cl ] [-a name ] [ command [arguments ... ] ] [redirection ... ]
```

To correct this problem, install Java or set the JAVA_HOME variable:

```bash
$ export JAVA_HOME=<path_to_java>
```

Where <path_to_java> is the path to where Java is installed.

Insufficient user resource limits errors

Insufficient resource limits might result in an insufficient nofiles error:

```bash
2012-08-13 11:22:51-0400 [] INFO: Could not accept new connection (EMFILE )
```

See Setting user resource limits to view the recommended settings for insufficient user resource limits errors.

Error getting version update information

If an OpsCenter node does not have internet access, or OpsCenter has difficulty with the URL for DataStax updates, the "Error getting version update information" message displays. The message indicates that notifications when new versions of OpsCenter or DataStax Enterprise (DSE) are available will not be shown. As a temporary workaround to hiding the error message, you can disable the latest version check.

Add the following to opscenterd.conf and restart opscenterd:

```conf
[labs]
```
latest_version_check = False

**OpsCenter cannot start**

OpsCenter uses the system temporary directory to load and execute dynamic class files. You might see an error message in the startup.log indicating that OpsCenter cannot read/write/execute from that directory. If permissions on the directory cannot be changed, you can use a JVM argument to change the temporary directory that OpsCenter uses. Refer to the JVM tuning section for more details.