

# Troubleshooting

On this page

## Cassandra fails with the "java.lang.OutOfMemoryError: Direct buffer memory" error

With some large databases, Cassandra may fail with the following error:

```
java.lang.OutOfMemoryError: Direct buffer memory
  at java.nio.Bits.reserveMemory(Bits.java:694)
  at java.nio.DirectByteBuffer.<init>(DirectByteBuffer.java:123)
  at java.nio.ByteBuffer.allocateDirect(ByteBuffer.java:311)
  at org.apache.cassandra.utils.memory.BufferPool.allocate(BufferPool.java:110)
  at org.apache.cassandra.utils.memory.BufferPool.access$1000(BufferPool.java:46)
  at org.apache.cassandra.utils.memory.BufferPool$LocalPool.allocate(BufferPool.java:407)
  at org.apache.cassandra.utils.memory.BufferPool$LocalPool.access$000(BufferPool.java:334)
  at org.apache.cassandra.utils.memory.BufferPool.takeFromPool(BufferPool.java:122)
  at org.apache.cassandra.utils.memory.BufferPool.get(BufferPool.java:94)
  at org.apache.cassandra.cache.ChunkCache.load(ChunkCache.java:155)
  at org.apache.cassandra.cache.ChunkCache.load(ChunkCache.java:39)
```

To solve the problem do one of the following:

- Increase the memory (Xmx VM option) for Cassandra until the migration is successful.
- Set `file_cache_size_in_mb: 0` in the `cassandra.yaml` file to disable the chunk cache completely.

## "All commit element IDs" index is corrupted

If the "all commit element IDs" index is corrupted, migration may fail and print an error of the following format into the Migrator's log file:

```
It has been detected that "all commit element IDs" index is corrupted at revision <revision> of resource
<resource.id>
```

To solve the problem

1. Wait until all selected projects finish migrating.
2. Stop the Migrator (backend) process.
3. At the end of the migrator's `application.conf` file, add the following line:

```
esi.migration.source-revision-all-object-ids-getter.prefer-bfs=true
```

4. Start the Migrator and retry the migration of the failed project from the Migrator GUI.

With this setting in place, it is expected that the migration progress will be much slower so it is recommended to finish the affected project with this setting and then revert back.

## The "java.net.SocketException: Too many open files" error appears during migration

This error usually means that the Cassandra or Teamwork Cloud process is running into system-imposed limits on a number of open files.

To solve the problem

1. Set the following limits in `/etc/security/limits.conf`.

```
localUser hard nofile 50000
localUser hard nproc 50000
localUser soft nofile 40000
localUser soft nproc 40000
cassandra hard nofile 50000
cassandra hard nproc 50000
cassandra soft nofile 40000
cassandra soft nproc 40000
root hard nofile 50000
root hard nproc 50000
root soft nofile 40000
root soft nproc 40000
```

2. Restart the processes.

Limits are configurable to separate users (like the Cassandra user that runs Cassandra or the Teamwork Cloud user that runs the Teamwork Cloud service). For more information, see <https://linux.die.net/man/5/limits.conf>.

## The "`com.datastax.driver.core.exceptions.AuthenticationException`" exception is shown in the log file

The following exception is shown in the log file:

```
Error while trying to activate [com.nomagic.esi.server.core.a.d.c, main]
com.datastax.driver.core.exceptions.AuthenticationException: Authentication error on host FQDN/IP: Provided
username cassandra and/or password are incorrect
```

To solve the problem:

- Add the following configuration block to **application.conf** of the migrator artifact:

### **application.conf**

```
authentication-enabled = true
authentication {
    username = your_cassandra_username
    password = your_cassandra_password
}
```