Genesys Mobile Services Deployment Guide

Configure an External Cassandra

12/17/2025

Contents

Configure an External Cassandra

Genesys Mobile Services (GMS) is packaged with an embedded Cassandra; however, GMS also supports deployments with an external Cassandra(s). An external Cassandra might be used in the following scenarios:

- You already have a Cassandra/Datastax deployment.
- You are securing your data in segregated networks, cages, and racks.
- You want multiple redundancy features, such as distinct data centers, rack, and chassis awareness.
- You are installing GMS on a production server or on a Windows host; GMS does not support embedded Cassandra on a Windows host or for production usage.

Configuring GMS for an external Cassandra is a multi-step process to enable connection, authentication, and authorization. The steps include setting configuration options in Configuration Manager (or Genesys Administrator), changing configuration settings in the `cassandra.yaml` file, and executing Cassandra Query Language (CQL) commands.

> **Important**
>
> - All external Cassandra nodes must be of the same version.
> - Cassandra is not required if you deploy Genesys Mobile Environment for Chat API V2, Email API V2, and Open Media API V2.
> - At times, Cassandra nodes get desynchronized and may create outdated events and data. In this scenario, if synchronizing nodes is not enough to solve the issue, you need to repair them, as described in the Official Cassandra documentation.

Install External Cassandra Nodes

If you have not already installed Cassandra, then follow these guidelines.

Prerequisites

- GMS supports Cassandra 2.x:
  - The latest tested version is 2.2.9
  - For Cassandra 2.0 and 2.1, use CQL 3.1.
- For the Network Topology, use one or several clusters.

Install Cassandra on All Nodes

Download Apache Cassandra. Use the right Unix user to install the application on all of your nodes (or host):

```
$  wget http://archive.apache.org/dist/cassandra/2.1.8/apache-cassandra-2.1.8-bin.tar.gz
```

Extract

```
$ tar xf  apache-cassandra-2.1.8-bin.tar.gz
```

Configure Cassandra

1. Edit the `conf/cassandra.yaml` file for all of the nodes that you installed:

```
cluster_name: '<the same name for all the cluster/datacenter>'
     -seeds "<cassandra_seed_host_ip_address_cluster_datacenter1>,
     <cassandra_seed_host_ip_address_cluster_datacenter_2>"

 # For example DC1 and DC2 (see below): -seeds "192.168.1.172,10.100.198.143"
 listen_address: <this cassandra host ip address>
 rpc_address: <this cassandra host ip address>
 start_rpc: true # must be set to true for cassandra 2.2
 rpc_port: 9160 # to be changed if needed
 endpoint_snitch: PropertyFileSnitch
```

2. Edit the `conf/cassandra-topology.properties` file for all of the nodes:

```
# Cassandra Node IP=Data Center:Rack
 # DC1
 192.168.1.172=DC1:RAC1 # seed node DC1
 192.168.1.215=DC1:RAC1
 192.168.1.234=DC1:RAC1

  # DC2
 10.100.198.143=DC2:RAC1        # seed node DC2
 10.100.198.94=DC2:RAC1
 10.100.198.136=DC2:RAC1
 # default for unknown nodes
 default=DC1:RAC1
```

3. Start all of the cassandra nodes.

Now, your cluster of cassandra nodes is functional.

Deploy GMS schemas in Cassandra

Create Schemas for your First Deployment

In a new fresh deployment, you need to create GMS keyspaces and tables in **one** of your nodes (not in all of them). Before you run the CQL scripts that perform these operations, edit the following files:

- <GMS Home>/scripts/cassandra_gsg_schema.cql
- <GMS Home>/scripts/cassandra_gsg_dd_schema.cql

Replace the text placeholders [ToBeChanged:<keyspace_name>] with your keyspace names and set the replication factor to NetworkTopologyStrategy.

For example, if you edit the cassandra_gsg_schema.cql file, here are the first lines that you will see:

```
CREATE KEYSPACE [ToBeChanged:<keyspace_name>] WITH replication =
{'class': 'SimpleStrategy', 'replication_factor': '2'}
AND durable_writes = true;

CREATE TABLE [ToBeChanged:<keyspace_name>].gsg_principal_roles (
    key blob,
    column1 blob,
    value blob,
    PRIMARY KEY (key, column1)
)
```

If your GMS keyspace name is gsg, you should edit these lines as follows:

```
CREATE KEYSPACE gsg WITH replication =
{ 'class' : 'NetworkTopologyStrategy', 'DC1' : 2, 'DC2' : 2 }
AND durable_writes = true;

CREATE TABLE gsg.gsg_principal_roles (
    key blob,
    column1 blob,
    value blob,
    PRIMARY KEY (key, column1)
)
```

After you have edited the CQL scripts, launch them to create the GMS schemas in the Cassandra node:

```
$ <CASSANDRA_HOME>/bin/cqlsh <cassandra_host> <cassandra_port> \
 -f <GMS_HOME>/scripts/cassandra_gsg_schema.cql
$ <CASSANDRA_HOME>/bin/cqlsh <cassandra_host> <cassandra_port> \
 -f <GMS_HOME>/scripts/cassandra_gsg_dd_schema.cql
```

> **Important**
>
> Create these schemas in one node only.

Update Schema after Upgrading

Starting in 8.5.102, Cassandra schemas are compatible with GMS 8.5.105+ and do **not** require any upgrade. But if you upgrade from GMS versions older than 8.5.102, you will need to manually update the Cassandra schemas in **one** of your nodes (not all).

Before running the CQL scripts that perform these operations, edit the following files:

- <GMS Home>/scripts/update_cassandra_gsg_schema.cql

- <GMS Home>/scripts/update_cassandra_gsg_dd_schema.cql

Replace the text placeholders [ToBeChanged:<keyspace_name>] with your keyspace names and set the replication factor to NetworkTopologyStrategy.

For example, if you edit the update_cassandra_gsg_schema.cql file, here are the first lines that you will see:

```
CREATE KEYSPACE IF NOT EXISTS [ToBeChanged:<keyspace_name>] WITH replication = {'class':
'SimpleStrategy', 'replication_factor': '2'}
AND durable_writes = true;

CREATE TABLE IF NOT EXISTS [ToBeChanged:<keyspace_name>].gsg_principal_roles (
    key blob,
    column1 blob,
    value blob,
    PRIMARY KEY (key, column1)
)
```

If your GMS keyspace name is gsg, you should edit these lines as follows:

```
CREATE KEYSPACE IF NOT EXISTS gsg WITH replication = {'class' : 'NetworkTopologyStrategy',
'DC1' : 2, 'DC2' : 2}
AND durable_writes = true;

CREATE TABLE IF NOT EXISTS gsg.gsg_principal_roles (
    key blob,
    column1 blob,
    value blob,
    PRIMARY KEY (key, column1)
)
```

After you edited the CQL scripts, launch them to update the schemas:

```
$ <CASSANDRA_HOME>/bin/cqlsh <cassandra_host> <cassandra_port> -f <GMS_HOME>/scripts/
update_cassandra_gsg_schema.default.cql
$ <CASSANDRA_HOME>/bin/cqlsh <cassandra_host> <cassandra_port> -f <GMS_HOME>/scripts/
update_cassandra_gsg_dd_schema.default.cql
```

> **Important**
> Update the schemas in one node only.

Configuration Options

The cassandra and cassandra-authentication-security sections list the configuration options applicable to an external Cassandra deployment. Changes take effect after restart.

Connection to an External Cassandra

The following steps are required to enable GMS to connect to an external Cassandra.

1. In Configuration Manager, locate and open your GMS Application object.
2. On the Options tab, **[cassandra]** section, set the following options:
   - **nodes** = <your Cassandra hosts or IP addresses for the local datacenter>
   - **port** = <your Cassandra port>
   - **create-embedded-server** = `false`
   - **strategy-class** = `NetworkTopologyStrategy`
   - **strategy-option** = `DC1:2;DC2:2`
3. Restart GMS.

> **Important**
> The strategy-class and strategy-option options must match the replication factor that you set when creating or updating your Cassandra schemas.

Authentication on External Cassandra

> **Important**
> Supports Cassandra version 2.0.x and higher (≤ 2.2).

The following steps are prerequisites prior to enabling authentication.

Configure cassandra.yaml File

1. Stop the Cassandra nodes.
2. Edit the `conf/cassandra.yaml` file for all nodes.
3. Ensure that `cluster_name` is identical for all nodes.
4. Locate the seed nodes. This is the field for all Cassandra nodes; change it accordingly:
   - For the seed node, this will be its own port.
   - For the non-seed nodes, this will be the IP address of the seed node.

5. Ensure that `listen_address` is changed from `127.0.0.1` to the current IP address.
6. Ensure that `rpc_address` is changed from `127.0.0.1` to the current IP address.
7. Locate the `authenticator` field.
8. Change the value from `AllowAllAuthenticator` to `PasswordAuthenticator`.

   ```
   authenticator: PasswordAuthenticator
   ```

   Note: The full classname is `org.apache.cassandra.auth.PasswordAuthenticator`.
9. Save the file.
10. Repeat these steps on each external Cassandra instance.
11. Start all the cassandra nodes.

Execute CQL Commands

1. On the external Cassandra, using the cqlsh utility (included with Cassandra), create your username and password for **one** of the nodes (not all). The following example shows the creation of a genesys user with genesys password.

   ```
   $ cqlsh -u cassandra -p cassandra cassandra_host cassandra_port
   > CREATE USER genesys WITH PASSWORD 'genesys';
   > LIST USERS;
       name | super
       -----------+-------
        genesys | False
        cassandra | True
   ```

   > **Important**
   >
   > The default superuser is `cassandra` with password `cassandra`. This step is required to be completed on only one external Cassandra instance. It will then be replicated to the other nodes.

2. On Windows OS / Cassandra versions 2.1 or 2.2, replace:

   ```
   $ cqlsh -u cassandra -p cassandra cassandra_host cassandra_port
   ```

   with:

   ```
   {path_to_cassandra}\bin>{path_to_python}\python.exe cqlsh cassandra_host -u cassandra -p
    cassandra
   ```

3. Set the options of `cqlsh` before parameters or set your python 2.7 path in PATH environment variable like this:

   ```
   PATH={path_to_python};%PATH%
   ```

   Therefore, you can launch the `cqlsh` script using the `cqlsh.bat` command:

   ```
   cqlsh.bat -u cassandra –p cassandra cassandra_host
   ```

   Using the default cassandra port of `native_transport_port` (default is `9042`). Otherwise you will need to add the port parameter to the `cqlsh` script.

4. If you use Cassandra 2.2, you should change the consistency level of the `system_auth` table and apply the CREATE command above according to the Cassandra version:

```
$ cqlsh -u cassandra -p cassandra cassandra_host cassandra_port
> ALTER KEYSPACE system_auth WITH REPLICATION = { 'class' : 'NetworkTopologyStrategy',
'DC1' : 2, 'DC2' : 2 };

>
```

Set Configuration Options

1. In Configuration Manager, locate and open your GMS Application object.
2. On the Options tab, **[cassandra-authentication-security]** section, set the following options with the same username and password that you just created on the external Cassandra.
   - **username**, for example, genesys
   - **password**, for example, genesys
3. **Restart GMS**. The Pelops and Hector clients connect to the external Cassandra using the login and password.

Authorization on External Cassandra

> **Important**
> Supports Cassandra version 2.0.x and higher (≤ 2.2).

After creating the authentication, you must enable authorization and create keyspaces.

Configure the cassandra.yaml Files

1. Edit the `conf/cassandra.yaml` file for all nodes. Locate the `authorizer` field.
2. Change the value from `AllowAllAuthorizer` to `CassandraAuthorizer`.

```
authorizer: CassandraAuthorizer
```

   Note: The full classname is `org.apache.cassandra.auth.CassandraAuthorizer`.
3. Save the file.
4. Repeat these steps on each external Cassandra instance.
5. Start all the cassandra nodes.

Execute CQL Commands

To authorize actions on the keyspace, you must first create the keyspace(s), then grant them

permissions in **one** of the nodes (not all).

1. On the external Cassandra, using the `cqlsh` utility (included with Cassandra), create your keyspaces in one of the nodes. The following example shows the `gsg` and `gsg_dd` keyspaces.

> **Important**
>
> This step is required to be completed on only one external Cassandra instance. It will then be replicated to the other nodes.

2. Set permissions to GMS keyspaces in Cassandra using CQLSH (example for cassandra 2.0, 2.1). Change the consistency level of GMS keyspaces and apply the CQL commands shown below according to the Cassandra version:

```
$ cqlsh -u cassandra -p cassandra cassandra_host cassandra_port
> LIST USERS;
        name | super
        -----------+-------
        genesys | False
        cassandra | True
> LIST ALL PERMISSIONS OF genesys;
        (0 rows)

> CREATE KEYSPACE gsg WITH REPLICATION = { 'class' : 'NetworkTopologyStrategy', 'DC1' :
2, 'DC2' : 2 };
> CREATE KEYSPACE gsg_dd WITH REPLICATION = { 'class' : 'NetworkTopologyStrategy', 'DC1'
: 2, 'DC2' : 2 };
> GRANT ALTER ON KEYSPACE gsg TO genesys;
> GRANT CREATE ON KEYSPACE gsg TO genesys;
> GRANT DROP ON KEYSPACE gsg TO genesys;
> GRANT MODIFY ON KEYSPACE gsg TO genesys;
> GRANT SELECT ON KEYSPACE gsg TO genesys;
> LIST ALL PERMISSIONS OF genesys;
username | resource | permission
        ----------+---------------+------------
        genesys | <keyspace gsg> | CREATE
        genesys | <keyspace gsg> | ALTER
        genesys | <keyspace gsg> | DROP
        genesys | <keyspace gsg> | SELECT
        genesys | <keyspace gsg> | MODIFY
        (5 rows)
> GRANT ALTER ON KEYSPACE gsg_dd TO genesys;
> GRANT CREATE ON KEYSPACE gsg_dd TO genesys;
> GRANT DROP ON KEYSPACE gsg_dd TO genesys;
> GRANT MODIFY ON KEYSPACE gsg_dd TO genesys;
> GRANT SELECT ON KEYSPACE gsg_dd TO genesys;
> LIST ALL PERMISSIONS OF genesys;
        username | resource | permission
        ----------+------------------+------------
        genesys | <keyspace gsg> | CREATE
        genesys | <keyspace gsg> | ALTER
        genesys | <keyspace gsg> | DROP
        genesys | <keyspace gsg> | SELECT
        genesys | <keyspace gsg> | MODIFY
        genesys | <keyspace gsg_dd> | CREATE
        genesys | <keyspace gsg_dd> | ALTER
        genesys | <keyspace gsg_dd> | DROP
        genesys | <keyspace gsg_dd> | SELECT
```

```
genesys | <keyspace gsg_dd> | MODIFY
(10 rows)
```

3. Add the user to Cassandra by using the same CQLSH commands than for Authentication.
4. Restart GMS. The Pelops and Hector clients connect to the external Cassandra and are authorized to manage the GMS keyspaces (`gsg` and `gsg_dd`).

Final Steps

- In the GMS Application > Security section > Log On As SYSTEM Account.
- The time zone for all nodes must be identical. Make sure that you synchronize the time before testing.