SIP Feature Server Deployment Guide

Migrate data from Embedded to External Cassandra and between Cassandra versions

12/15/2025

Contents

Migrate data from Embedded to External Cassandra and between Cassandra versions

Cassandra versions 2.x and higher do not support backward compatibility with Cassandra versions 1.x. The data migration is required when upgrading Feature Server's Cassandra database backend from embedded Cassandra version 1.x to external versions 2.x and/or 3.x.

Feature Server release 8.1.202.02 includes the following Python scripts for migrating data from embedded Cassandra database to Cassandra versions 2.x or 3.x:

- **copyKeyspaceSchema.py**—Creates a keyspace and its column families in the destination Cassandra cluster.

- **copyKeyspaceColumnFamilies.py**—Copies content of source keyspace column families to the destination keyspace column families.

Cassandra 4.x migration If your current deployment environment uses Embedded Cassandra and you want to migrate to Cassandra 4.x, the scripts provided above are not compatible. Note that it is also not possible to directly migrate an embedded Cassandra to Cassandra 4.x.

If you want to move SIP Feature Server's database to Cassandra 4.x,

- Have your externally deployed Cassandra 2.x/3.x migrated to latest 3.11 per Cassandra's official recommendations or

- Migrate the embedded Cassandra to Cassandra 3.11 using the scripts provided in the previous section and

- Perform an in-place upgrade from Cassandra 3.11 to Cassandra 4.x. per Cassandra's official recommendations.

Prerequisites for data migration to external Cassandra

The following are the prerequisites for the data migration from versions 1.x to versions 2.x and/or 3.x.

- Destination Cassandra cluster must be deployed and all the nodes must be up and running.

- In terms of Feature Server deployment, the destination Cassandra cluster must be deployed in external mode.

- The destination Cassandra cluster must not have any Feature Servers assigned to it before the copying of data from the source Cassandra cluster is completed.

- SIP Feature Server must run in the **ReadOnly** mode to ensure proper data copy during migration with running Feature Servers. The **ReadOnly** mode must be turned on before deploying the Python migration scripts. Use the following configuration:

```
[Cassandra]readOnly=true
```

Migrating data from Embedded to External Cassandra

The following steps show how to migrate data from Cassandra v1.x to v2.x and v3.x

1. Deploy the Python scripts.

2. Run the Python scripts.

3. Connect Feature Server nodes to migrated Cassandra cluster.

Deactivate Embedded Cassandra module for version 8.1.203 and later by referring to the procedure here.

Deploy the Python scripts

1. Install Python 2.7.5 32-bit version and Pycassa libraries on the destination Cassandra host where the scripts must be run.

2. The Python scripts **copyKeyspaceSchema.py**, **copyKeyspaceColumnFamilies.py** and the sample json input file, **copyKeyspaceInput.json** are present in the Python utilities folder of Feature Server deployment: *FS installation path*/**Python/util/**. Copy these script files to a directory on the destination Cassandra host.

3. Navigate to the directory location and run the scripts.

For more details, refer to Python Scripts.

Run the Python scripts

Following is a sample **copyKeyspaceInput.json** input json file:

```
{"sourceHostPort": "FsNode01:9160",
"sourceHostUserName": "",
"sourceHostPassword": "",
"sourceHostTls": "false",
 "destinationHostPort": "CassNode01:9160",
 "destinationHostUserName": "",
 "destinationHostPassword": "",
 "destinationHostTls": "false",
 "replicationStrategyClassName": "NetworkTopologyStrategy",
 "replicationOptions": {"DC1":"2", "DC2":"2"},
 "sourceKeyspace": "sipfs",
 "destinationKeyspace": "sipfs",
 "excludedCFs": [ ],
 "includedCFs": [ ] }
```

Copy keyspace schema

The following steps show the procedure to copy the keyspace schema:

1. Verify that the input json file has the following parameters:

Parameters	Description	Sample	Mandatory
sourceHostPort	Host and the Thrift port of source Cassandra DB in the URL format: *host IP:port*	`FsNode01:9160`	Yes
destinationHostPort	Host and the Thrift port of destination Cassandra database in the URL format: *host IP:port*	`CassNode01:9160`	Yes
sourceKeyspace	Name of the source keyspace	`sipfs`	Yes
destinationKeyspace	Name of the destination keyspace	`sipfs`	Yes
replicationStrategyClassName	Replication Strategy Class Name	`NetworkTopologyStrategy`	Yes
replicationOptions	Replication Options for the destination keyspace Ensure to configure this value according to the **cassandra-toplogy.properties** file.	`{"DC1": "2", "DC2": "2"}`	Yes
sourceHostUserName	The username of source Cassandra.	`FSadmin`	Yes, if authentication is enabled in the source Cassandra Cluster.
sourceHostPassword	The password of source Cassandra.	`FSadmin`	Yes, if authentication is enabled in the source Cassandra Cluster.
sourceHostTls	Set this option to `true` when SSL is enabled for the source Cassandra connection.	`true`	Yes, if SSL is enabled for the source Cassandra.
destinationHostUserName	The username of destination Cassandra.	`FSadmin`	Yes, if authentication is enabled in the destination Cassandra Cluster.
destinationHostPassword	The password of destination Cassandra.	`FSadmin`	Yes, if authentication is enabled in the destination Cassandra Cluster.
destinationHostTls	Set this option to `true` when	`true`	Yes, if SSL is enabled for the

	SSL is enabled for the destination Cassandra connection.		destination Cassandra.

2. Run the **copyKeyspaceSchema.py** script.

   Sample command line
   ```
   python ./copyKeyspaceSchema.py -i ./copyKeyspaceInput.json -o
   ./copyKeyspaceSchema_`date +%y%m%d-%H:%M`.log
   ```

Copy keyspace column families

1. Verify that the input json file has the following parameters:

Parameters	Description	Sample	Mandatory
sourceHostPort	Host and the Thrift port of source Cassandra database in the URL format: *host IP:port*	FsNode01:9160	Yes
destinationHostPort	Host and the Thrift port of destination Cassandra database in the URL format: *host IP:port*	CassNode01:9160	Yes
sourceKeyspace	Name of the source keyspace	sipfs	Yes
destinationKeyspace	Name of the destination keyspace	sipfs	Yes
excludedCFs	List of comma-separated column family names to be excluded from copying while running the **copyKeyspaceColumnFamilies.py** script.	message_bytes, device	No
includedCFs	List of comma-separated column family names to be copied while running the **copyKeyspaceColumnFamilies.py** script.	message_bytes, device	No
sourceHostUserName	The username of source Cassandra.	FSadmin	Yes, if authentication is enabled in the source Cassandra Cluster.

sourceHostPassword	The password of source Cassandra.	FSadmin	Yes, if authentication is enabled in the source Cassandra Cluster.
sourceHostTls	Set this option to `true` when SSL is enabled for the source Cassandra connection.	`true`	Yes, if SSL is enabled for the source Cassandra.
destinationHostUserName	The username of destination Cassandra.	FSadmin	Yes, if authentication is enabled in the destination Cassandra Cluster.
destinationHostPassword	The password of destination Cassandra.	FSadmin	Yes, if authentication is enabled in the destination Cassandra Cluster.
destinationHostTls	Set this option to `true` when SSL is enabled for the destination Cassandra connection.	`true`	Yes, if SSL is enabled for the destination Cassandra.

If one or more source column families contain huge volumes of data, then run the **copyKeyspaceColumnFamilies.py** script to copy these column families separately from the rest of the source column families. Use the excludedCFs and includedCFs parameters to exclude or include a specific column family. When the includedCFs list is not empty, the excludedCFs parameter is ignored and only the column families in the includedCFs list are copied.

For example, provide the following json file as the input to the **copyKeyspaceColumnFamilies.py** script to copy the content of all column families except message_bytes column family.

```
{"sourceHostPort": "FsNode01:9160",
 "sourceHostUserName": "",
 "sourceHostPassword": "",
 "sourceHostTls": "false",
 "destinationHostPort": "CassNode01:9160",
 "destinationHostUserName": "",
 "destinationHostPassword": "",
 "destinationHostTls": "false",
 "replicationStrategyClassName": "NetworkTopologyStrategy",
 "replicationOptions": {"DC1": "2", "DC2": "2"},
 "sourceKeyspace": "sipfs",
 "destinationKeyspace": "sipfs",
```

```
"excludedCFs": [ "message_bytes" ],
"includedCFs": [ ] }
```

For example, provide the following json file as input to the **copyKeyspaceColumnFamilies.py** script to copy the content of only the message_bytes column family.

```
{"sourceHostPort": "FsNode01:9160"
"sourceHostUserName": "",
"sourceHostPassword": "",
"sourceHostTls": "false",
"destinationHostPort": "CassNode01:9160",
"destinationHostUserName": "",
"destinationHostPassword": "",
"destinationHostTls": "false",
"replicationStrategyClassName": "NetworkTopologyStrategy",
"replicationOptions": {"DC1": "2", "DC2": "2"},
"sourceKeyspace": "sipfs",
"destinationKeyspace": "sipfs",
"excludedCFs": [],
"includedCFs": [ "message_bytes" ] }
```

2. Run the **copyKeyspaceColumnFamilies.py** script.

   Sample command line
   ```
   python ./copyKeyspaceColumnFamilies.py -i ./copyKeyspaceInput.json -o
   ./copyKeyspaceContent_`date +%y%m%d-%H:%M`.log
   ```

   > ## Important
   >
   > If there are regional keyspaces to be copied, all the keyspaces, the global keyspace and all regional keyspaces must be copied one after the other. To copy all keyspaces, the scripts must be run for each keyspace: the global keyspace and each regional keyspace.

Connecting Feature Server nodes to migrated Cassandra cluster

The following steps should be performed for every Feature Server node involved:

1. Disable the **ReadOnly** mode in Feature Server. Use the configuration: `[Cassandra]readOnly=false`

2. Stop Feature Server node.

3. Edit *<FS installation path>*\launcher.xml file and set the property `startCassandra` to **False**.

   ```
   <parameter name="startCassandra"
   displayName="com.genesyslab.common.application.cassandraServer" hidden="true"
   mandatory="false">
   <description><![CDATA[ Start Cassandra Server]]></description>
   <valid-description><![CDATA[]]></valid-description>
   <effective-description/>
   <format type="string" default="false"/>
   <validation>
   </validation>
   ```

```
</parameter>
```

4. Update the **[Cassandra]** section of the Feature Server application as shown in the following table:

[Cassandra] section Option	Default Value	Feature Server Application Value	Mandatory
nodes	NA	Configure all the Cassandra nodes IP addresses that belong to the data center where Feature Server is installed.	Yes
nodeFailureTolerance		Replication factor of Feature Server data center is 1. If the regional keyspace is used, then the least value (keyspace, regional keyspace) replication_factor of its data center is 1. For example, if the DC1 contains 4 nodes and the replication_factor for the global keyspace is 3 and the regional keyspace is 2, then the value is 1.	No
keyspace	sipfs	Name of the 'global' keyspace This option must have the same value as the keyspace name parameter for the **copyKeyspaceSchema.py** script when copying the global keyspace.	No
replicationStrategyClassName	NA	This option must have the same value as the replication options parameters for the **copyKeyspaceSchema.py** script when copying both the global keyspace and the regional keyspace values.	Yes
replicationOptions	NA	This option must have the same value as the replication options parameters for the **copyKeyspaceSchema.py** script.	Yes
regionalKeyspace	sipfs_<region>	Name of the regional keyspace This option must have the same value as the	Mandatory if regional keyspace(s) is enabled.

		replication options parameters for the **copyKeyspaceSchema.py** script when copying the regional keyspace.	
regionalReplicationOptions	NA	This option must have the same value as the replication options parameters for the **copyKeyspaceSchema.py** script.	Mandatory if regional keyspace(s) is enabled.
username	cassandra	Cassandra Username	Mandatory if authentication is enabled in Cassandra Cluster.
password	cassandra	Cassandra Password	Mandatory if authentication is enabled in Cassandra Cluster.

5. Start Feature Server node.

Upgrading external Cassandra cluster to Cassandra 4.x

Prerequisites

1. Ensure that SIP Feature Server already works with the external Cassandra and the connection mode between Feature Server and external Cassandra was switched from the `Thrift` to `CQL` mode. Switching Feature Server's connection mode to **CQL** can be done by configuring the options mentioned in Provisioning of Cassandra Parameters.

2. Enable read-only mode of the SIP Feature Server application by setting the **readOnly** option to `true` in the Cassandra section of the application options.

3. Follow Cassandra's official recommendations to migrate your Cassandra 3.11 cluster to Cassandra 4.X. Genesys provides only a sample migration procedure that would help you to plan steps for your own specific deployment.

Sample migration procedure

Start the migration by upgrading the seed node first and then proceed with other nodes.

Pre-upgrade checks

1. Confirm that all nodes are up and normal by running the following command:

```
# nodetool status | grep -v UN      => Returns nodes that are not marked as UN (U-UP
N-Normal)
```

```
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address     Load       Tokens      Owns (effective)  Host ID
```

2. Confirm that you don't receive any unresolved errors after you run the following command:

```
sudo grep -e "WARN" -e "ERROR" <path to cassandra installed folder>/logs/
system.log      => Returns Warning and Error messages in cassandra system logs -
should not return any error
```

3. Confirm that gossip information is stable by running the following command:

```
# nodetool gossipinfo | grep STATUS | grep -v NORMAL      => Returns gossipinfo
status that are not Normal - should return empty
```

4. Confirm that there are no dropped messages by running the following command:

```
# nodetool tpstats | grep -A 12 Dropped
        Message type            Dropped
        READ                       0
        RANGE_SLICE                0
        _TRACE                     0
        HINT                       0
        MUTATION                   0
        COUNTER_MUTATION           0
        BATCH_STORE                0
        BATCH_REMOVE               0
        REQUEST_RESPONSE           0
        PAGED_RANGE                0
        READ_REPAIR                0
```

5. Repair each node before upgrading by running the following command:

```
# nodetool repair -pr
```

Running the above command does not give any results. However, the time it runs might be long depending on the size of data.

Create Snapshot

Create a pre-upgrade snapshot backup by running the following command.

```
# nodetool snapshot --tag pre-upgrade
        Requested creating snapshot(s) for [all keyspaces] with snapshot name [pre-upgrade]
and options {skipFlush=false}
        Snapshot directory: pre-upgrade
```

Backup

1. Shut down Cassandra by running the following commands.

```
a. # nodetool drain             => No response expected. To restrict requests from
clients
b. # nodetool netstats          => To check drain status - Mode should be marked
DRAINED
        Mode: DRAINED
        Not sending any streams.
        Read Repair Statistics:
        Attempted: 0
```

```
            Mismatch (Blocking): 0
            Mismatch (Background): 0
            Pool Name                    Active    Pending      Completed    Dropped
            Large messages                n/a         2              0          0
            Small messages                n/a         2              5          0
            Gossip messages               n/a         2            122          0
```

2. Stop Cassandra by running the following commands.

   ```
   a. # sudo kill $(sudo lsof -t -i:7199)
   b. # ps auwx | grep CassandraDaemon
   ```

3. Back up Cassandra configuration and data files by running the following commands.

   ```
   cd <path to cassandra installed folder> && tar czfv <user defined path>/cassandra-
   config-backup.tgz ./conf
   Note: Below commands are needed to place data directory in common path, if not
   already (time consumption depends on data size)
   cd <path to cassandra installed folder> && tar czfv <user defined
   path>/cassandra-data-backup.tgz ./data/data
   cd <user defined path>/ && tar xzf cassandra-data-backup.tgz        => To
   extract data files
   ```

Install and Configure the new Cassandra

1. Install the new Cassandra package by running the following commands:

   ```
   curl -OL https://archive.apache.org/dist/cassandra/4.x.x/apache-cassandra-4.x.x-
   bin.tar.gz
   Note: Extract zip file and move to expected path
   tar xzf apache-cassandra-4.1.2-bin.tar.gz
   mv apache-cassandra-4.1.2 /<user defined path>
   ```

2. Configure user roles for Cassandra 4.x and its data directory.

   ```
   sudo chown -R <fs_admin_role>:<fs_admin_role> <path to cassandra 4.x installed
   folder>          => Extracted folder of cassandra 4.x.x
   sudo chown -R <fs_admin_role>:<fs_admin_role> <user defined path>          =>
   Extracted folder of backup data from older version
   ```

3. Update Cassandra configuration files of new version.

   ```
   Copy the cassandra-topology.properties file from older to new version.
   cp <path to cassandra 3.x installed folder>/conf/cassandra-topology.properties
   <path to cassandra 4.x installed folder>/conf

   Update the conf/cassandra.yaml file in cassandra 4.x.x extracted folder with the
   following options.
   cluster_name: <cluster_name> (default:FeatureServerCluster)
   num_tokens: 256
   data_file_directories: <user defined path>/data/data
   - seeds: "<seed_node_ip>"
   listen_address: <node_ip>
   rpc_address: (empty)
   endpoint_snitch: PropertyFileSnitch
   ```

Upgrade

1. Start Cassandra from Cassandra 4.x.x extracted folder by running the following command:

```
<path to cassandra 4.x installed folder>/conf/bin/cassandra -f
```

2. Verify if Cassandra latest version has started from logs.

```
sudo tail -n 50 -f <path to cassandra 4.x installed folder>/logs/system.log
INFO  [main] 2024-04-18 10:05:32,432 SystemKeyspace.java:1729 - Detected
version upgrade from 3.11.16 to 4.1.2, snapshotting system
keyspaces
INFO  [main] 2024-04-18 10:05:37,489 StorageService.java:864 - Cassandra
version: 4.1.2
```

3. Check if all nodes are marked as UN, use the following command:

```
nodetool status
```

4. Monitor the thread pool status by running the following command. There should be no pending, blocked, or dropped messages.

```
watch -d nodetool tpstats
```

Update SST Tables (one node at a time)

1. Upgrade SSTables by running the following command:

```
nodetool upgradesstables         => should return empty
watch -d "nodetool compactionstats -H"        => pending tasks should be 0
     Every 2.0s: nodetool compactionstats -H
     pending tasks: 0
```

2. Confirm SSTables have been upgraded by checking the data folder copied to user defined path from older Cassandra.

```
All table files will be modified with 'nb-' prefix. Will return the files that are
not modified.
    sudo find <user defined path>/data/data -type f | grep -v "snapshots" | rev |
cut -d'/' -f1 | rev | grep -v "^nb\-"
        output:
        grep: warning: stray \ before -
        ballot.meta
```

Cleanup

Remove snapshot by running the following command.

```
 nodetool clearsnapshot -t pre-upgrade
Requested clearing snapshot(s) for [all keyspaces] with snapshot name [pre-upgrade]
```

Upgrade other nodes

Repeat all the above steps for remaining nodes.

Reset and restart SIP Feature Server applications

In the Feature Server application options, set the **readOnly** option to `false` and restart the Feature Server applications one by one.

Validation

Verify if Cassandra latest version has started from logs. Use the following command:

```
sudo tail -n 50 -f <path to cassandra 4.x installed folder>/logs/system.log
INFO  [main] 2024-04-18 10:05:32,432 SystemKeyspace.java:1729 - Detected version upgrade
from 3.11.16 to 4.1.2, snapshotting system
keyspaces
INFO  [main] 2024-04-18 10:05:37,489 StorageService.java:864 - Cassandra version: 4.1.2
```

The Cassandra version can also be verified by using the following nodetool command:
<source lang = "bash">
```
nodetool version
    ReleaseVersion: 4.1.2
```

In the Feature Server Cassandra logs, look for similar log information like the following to verify the Cassandra nodes connected to Feature Server:

```
2024-04-24 05:13:28,964 [pool-19-thread-1] - [INFO] New Cassandra host
usw1lbe-35-14-002.usw1.genhtcc.com/10.51.27.108:9042 added
2024-04-24 05:13:28,965 [pool-19-thread-1] - [INFO] New Cassandra host
usw1lbe-35-14-001.usw1.genhtcc.com/10.51.26.107:9042 added
```

In the Feature Server logs, look for similar log information like the following to verify the successful connection of Feature Server with upgraded Cassandra nodes and its functioning.

```
2024-04-24T05:13:26.971 Trc 09900  [INFO] Cassandra connection pool :
usw1lbe-35-14-001.usw1.genhtcc.com,usw1lbe-35-14-002.usw1.genhtcc.com.
...
2024-04-24T05:13:29.091 Trc 09900  [INFO] [Cassandra] cluster name
FeatureServerClusterVoicemail35-14
2024-04-24T05:13:29.130 Dbg 09900  [DEBUG] Syncing schema, keyspace: 'sipfs' ... CQL mode.
2024-04-24T05:13:29.141 Dbg 09900  [DEBUG] Syncing column families: cluster
usw1lbe-35-14-001.usw1.genhtcc.com,usw1lbe-35-14-002.usw1.genhtcc.com:9042 ... CQL mode.
2024-04-24T05:13:29.226 Dbg 09900  [DEBUG] Completed syncing schema, keyspace: sipfs ... CQL
mode.
2024-04-24T05:13:29.228 Dbg 09900  [DEBUG] Repository is activated
2024-04-24T05:13:29.236 Trc 09900  [INFO] Repository activated:
com.genesyslab.feature.component.system.FsSystemRepository, mode: online)
2024-04-24T05:13:29.281 Trc 09900  [INFO] Operational mode: 'Standalone'.
2024-04-24T05:13:29.282 Trc 09900  [INFO] Configuration server id: 'aa3244da-
fa51-4455-af52-a207086d7935'.
2024-04-24T05:13:29.283 Trc 09900  [INFO] Setting cluster node data...
2024-04-24T05:13:29.399 Trc 09900  [INFO] Cluster node data has been set.
2024-04-24T05:13:29.469 Trc 09900  [INFO] Set node switch data.
...
2024-04-24T05:15:02.312 Std 05061  Initialization completed
```