



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# SIP Feature Server Deployment Guide

Configure SIP Feature Server to work with the Cassandra cluster

12/19/2025

# Configure SIP Feature Server to work with the Cassandra cluster

Perform the following steps on each SIP Feature Server instance:

1. During installation, ensure that you select the Cassandra cluster type as **External Cassandra**. If you didn't select this type of installation or if you want to verify the current mode, verify the following:
  - Make sure the value of the parameter **startServer** (**com.genesyslab.common.application.cassandraServer**) is set to false in **<Feature Server installed directory>\launcher.xml** (or **launcher\_64.xml** on Linux).
  - Make sure that you set **connection-type** in the **Cassandra** section to a value (cql or thrift) according to the protocol you are planning to use.
2. In the **SIP Feature Server application > [Cassandra]** section, configure the options as follows:
  - nodes=IPAddress of all the Cassandra nodes that are available in that data center.
  - keyspace=<keyspace\_name> Keyspace name for SIP Feature Server application. The default is sipfs.
  - nodeFailureTolerance=replication\_factor Value of its data center-1.  
For example, if DC1 is the data center where SIP Feature Server is connected and the replication factor is DC1=3, DC2=3, then configure nodeFailureTolerance=2.
3. If Authentication is enabled in Cassandra, then configure the following options in the **[Cassandra]** section under SIP Feature Server application:
  - username=<cassandra\_username>
  - password=<Cassandra\_password>
4. If Cassandra TLS encryption is enabled on the CQL port, then perform the following steps:
  - Set the **cassandra\_encryption** parameter (**com.genesyslab.voicemail.application.cassandraEncryption**) to true in **<Feature Server installed directory>\launcher.xml** (or **launcher\_64.xml** on Linux).
  - Set Feature Server to trust all remote Cassandra server certificates by default.
    - If you want to verify the remote server certificate, configure the option **trusted-ca** in the **Cassandra** section of the Feature Server application object. The value should be the path to a file with trusted certificate authority you want to use to verify the remote server certificate. Note that the file must be in the PEM format and it is stored in a local folder that is accessible by Feature Server process(es).
  - Set Feature Server to skip validation of remote Cassandra server's hostname that matches with the subject of the certificate returned by that server by default.
    - If you want to enforce strict validation, configure the **verify-host** option to true in the **Cassandra** section.

### Important

If you want to use TLS encryption when connecting to legacy Cassandra deployments that uses the Thrift protocol, then, instead of configuring the above options, create a truststore under **<SIP Feature Server installed directory>/etc** and import the public key certificates of the Cassandra nodes. Then, edit the **<SIP Feature Server installed directory>/launcher.xml** file (or **launcher\_64.xml** for Linux) and set **javax.net.ssl.trustStore** to **./etc/<path of the truststore file>**, and **javax.net.ssl.trustStorePassword** to **<truststore password>**