Genesys Knowledge Center Deployment Guide

Deploying Cassandra Cluster

12/16/2025

Contents

Deploying Cassandra Cluster

> **Important**
>
> To ensure strong consistency of the data (no matter the configuration of the underlying storage) it was decided to deprecate Cassandra 2.2.x support in 8.5.3 release of the product. Going forward, **support of Cassandra will be discontinued in 9.0 release of the product**.

Genesys recommends using an external Cassandra as the persistent storage for the data stored in Knowledge Center CMS. This chapter describes sample procedure of deploying and configuring Cassandra nodes. For more information please refer to Cassandra documentation.

> **Important**
>
> Genesys recommends that you use Linux when deploying an external Cassandra cluster.
> If you plan to establish secure communications with your Cassandra cluster, Genesys recommends that you carefully evaluate the related security considerations.

Deploy a Cassandra Cluster Node

Linux

Installation

1. Download version 2.2.3 or higher from the Cassandra 2.2 stream.
2. Unpack the archive into the installation directory, for example:

   ```
   cd /genesys tar xzf apache-cassandra-2.2.x-bin.tar.gz
   ```

   > **Important**
   >
   > Do not use paths with spaces when installing Cassandra 2.2

Configuration

1. Go to the directory where you installed your Cassandra node.

2. Edit conf/cassandra.yaml, using the following custom values:
    a. cluster_name: cluster name without spaces, for example GKC_Cassandra_Cluster
    b. seeds: <comma-separated list of fully qualified domain names (FQDN) or IP addresses of one or more Cassandra nodes>
    Note: This value must be the same for all nodes. Here are two examples:
        - 192.168.0.1,192.168.3
        - host1.mydomain.com, host2.mydomain.com
    c. storage_port: 7000 (default value)
    d. ssl_storage_port: 7001 (default value)
    e. listen_address: <current node host name>
    Note: This address is used for inter-node communication, so it must be available for use by other Cassandra nodes in your cluster.
    f. native_transport_port: 9042 (default value)
    g. rpc_address: <current node host name> Note: This address is used by Knowledge Center CMS to connect to
    h. Cassandra, so it must be available to all Knowledge Center CMS hosts.
    i. rpc_port: 9160 (default value)
    j. start_rpc: true
    k. endpoint_snitch: GossipingPropertyFileSnitch
    Note: Make sure that each Cassandra node has access to the ports specified for the other nodes.
3. Edit conf/cassandra-rackdc.properties.
4. Verify that the required communication ports are opened.

Setting Up a Cassandra Service

The sample script described in the following procedure should give you an idea of how to set up Cassandra as a service process.

1. Create the /etc/init.d/cassandra startup script.
2. Edit the contents of the file:

```
#!/bin/sh # # chkconfig: - 80 45 # description:
 Starts and stops Cassandra # update daemon path to point to the
 cassandra executable DAEMON=<Cassandra_installation_dir>
 /bin/cassandra start() { echo -n "Starting Cassandra...
 " $DAEMON -p /var/run/cassandra.pid echo "OK"
 return 0 } stop() { echo -n "Stopping Cassandra... "
 kill $(cat /var/run/cassandra.pid) echo "OK"
 return 0 } case "$1" in start) start ;; stop) stop ;;
 restart) stop start ;; *) echo $"Usage: $0
 {start|stop|restart}" exit 1 esac exit $?
```

3. Make the file executable: sudo chmod +x /etc/init.d/cassandra
4. Add the new service to the list: sudo chkconfig --add cassandra
5. Now you can manage the service from the command line:

  - sudo /etc/init.d/cassandra start
  - sudo /etc/init.d/cassandra stop

6. Configure the service to be started automatically together with the VM: sudo chkconfig --level 2345 cassandra on

Windows

Installation

1. Download version 2.2.3 or higher from the Cassandra 2.2 stream.
2. Unpack the archive into a path without spaces.

Configuration

1. Go to the directory where you installed your Cassandra node.
2. Edit cassandra.yaml, using the following custom values:
   a. cluster_name: cluster name without spaces, for example GKC_Cassandra_Cluster
   b. seeds: <comma-separated list of fully qualified domain names (FQDN) or IP addresses of one or more Cassandra nodes>
      Note: This value must be the same for all nodes. Here are two examples:
      - 192.168.0.1,192.168.3
      - host1.mydomain.com, host2.mydomain.com
   c. storage_port: 7000 (default value)
   d. ssl_storage_port: 7001 (default value)
   e. listen_address: <current node host name>
      Note: This address is used for inter-node communication, so it must be
   f. available for use by other Cassandra nodes in your cluster.
   g. native_transport_port: 9042 (default value)
   h. rpc_address: <current node host name>
      Note: This address is used by Knowledge Center CMS to connect to
   i. Cassandra, so it must be available to all Knowledge Center CMS hosts.
   j. rpc_port: 9160 (default value)
   k. start_rpc: true
   l. endpoint_snitch: GossipingPropertyFileSnitch
3. Edit conf/cassandra-rackdc.properties.
4. Verify that the required communication ports are opened.
5. Start Cassandra.

Tuning Cassandra Configuration

Configuring cassandra-rackdc.properties

For a single data center, use the following as a guide:

```
dc=<Data Center name>
rack=<RACK ID>
```

Example:

```
dc=OperationalDC
rack=RAC1
```

Important

Genesys recommends that you use the same rack ID if you do not have a clear understanding of your servers' rack usage. For more information about cassandra-rackdc.properties, refer to http://docs.datastax.com/en/cassandra/2.2/cassandra/architecture/archsnitchGossipPF.html

Communication Ports

Cassandra use the following ports for external and internode communication. Note: Either or both of them may not work as expected unless you ensure that these ports are opened for communication between all servers that host Cassandra nodes.

Port	Default	Where to change the value
Cassandra Storage port	7000	storage_port in cassandra.yaml
Cassandra SSL Storage port	7001	ssl_storage_port in cassandra.yaml
Cassandra Thrift port	9160	rpc_port in cassandra.yaml (Knowledge Center CMS uses Thrift protocol to communicate to Cassandra)
Cassandra CQL port	9042	native_transport_port in cassandra.yaml

Working with Cassandra

Starting the Cassandra Cluster Nodes

Your Cassandra nodes must be started in a certain order:

1. Start the seed nodes.
2. Start the other non-seed nodes.

The seed node is one of the nodes specified in the seeds option.

Verifying Your Cassandra Cluster

After you have deployed your Cassandra Cluster, you may want to verify that all of the nodes can communicate with each other. To do this, execute the following command on any Database VM:

Linux

```
cd <Cassandra_installation_dir>/bin ./nodetool -h <hostname> status
```

Windows

```
cd <Cassandra_installation_dir >/bin nodetool -h <hostname> status
```

Sample output

This command should produce output that looks something like this:

```
Datacenter: DC1 ========================== Status=Up/Down |/
State=Normal/Leaving/Joining/Moving -- Address Load Tokens Owns Host ID
Rack UN 10.51.XX.XXX 106,36 KB 256 ? 380d02fb-da6c-4f6a-820e-14538bd24a39
RAC1 UN 10.51.XX.XXX 108,22 KB 256 ? 601f05ac-aa1d-417b-911f-22340ae62c38
RAC1 UN 10.51.XX.XXX 107,61 KB 256 ? 171a15cd-fa4d-410e-431b-51297af13e96
RAC1 Datacenter: DC2 ========================== Status=Up/Down |/
State=Normal/Leaving/Joining/Moving -- Address Load Tokens Owns Host ID
Rack UN 10.51.XX.XXX 104,06 KB 256 ? 48ad4d08-555b-4526-8fab-d7ad021b14af
RAC1 UN 10.51.XX.XXX 109,56 KB 256 ? 8ca0fb45-aef7-4f0a-ac4e-a324ceea90c9
RAC1 UN 10.51.XX.XXX 105,18 KB 256 ? 1c45e1fa-9f82-4bc4-a896-5575bad53808
RAC1
```

Upgrading Cassandra Nodes

You can upgrade your Cassandra version without interrupting service if:

- The version you are upgrading to is in the same stream (for example, from one 2.2.x version to another)
- You are not changing your database schema

Use the following steps for this task:

1. Stop the first Cassandra seed node.
2. Preserve your database storage.
3. Upgrade your Cassandra version, following the instructions in the Release Notes for the new version.
4. Be sure that your database storage is in the preserved state (the same set of files).
5. Start the first Cassandra seed node.
6. Execute steps 1 through 5 for the other seed nodes.
7. Execute steps 1 through 5 for the other non-seed nodes.
8. Verify that the Cassandra cluster is working, as shown above in Verifying Your Cassandra Cluster.

If your upgrade plans include changing your database schema or changing Cassandra versions between streams (for example, from 2.0 to 2.2), then you will have to interrupt service. Use the following steps for this task:

1. Stop all of your Cassandra nodes.
2. If your database schema has been changed since you installed the previous version, update the Cassandra database, following the instructions in the Release Notes for the new version.
3. Configure each node, following the instructions in the Release Notes for the new version.
4. Start the Cassandra seed nodes.
5. Start the other nodes.
6. Verify that the Cassandra cluster is working, as shown above in Verifying Your Cassandra Cluster.

Maintenance

Because Cassandra is a critical component of Knowledge Center CMS, it is essential to keep track of its health. The Datastax documentation provides some really good information about how to do this at http://docs.datastax.com/en/cassandra/2.0/cassandra/tools/toolsNodetool_r.html.
Genesys recommends that you use the nodetool utility that is bundled with your Cassandra installation package and that you make a habit of using the following nodetool commands to monitor the state of your Cassandra cluster.

ring

Displays node status and information about the cluster, as determined by the node being queried. This can give you an idea of the load balance and whether any nodes are down. If your cluster is not properly configured, different nodes may show a different cluster; this is a good way to check that every node views the cluster the same way.

```
nodetool -h <HOST_NAME> -p <JMX_PORT> ring
```

status

Displays cluster information.

```
nodetool -h <HOST_NAME> -p <JMX_PORT> status
```

compactionstats

Displays compaction statistics.

```
nodetool -h <HOST_NAME> -p <JMX_PORT> compactionstats
```

getcompactionthroughput \ setcompactionthhroughput

Displays the compaction throughput on the selected Cassandra instance. By default it is 32 MB/s. You can increase this parameter if you observe permanent growth of database size after the TTL and grace periods are passed. Note that increasing compaction throughput will affect memory and CPU consumption. Because of this, you need make sure to have sufficient hardware to support the rate that you have selected.

```
nodetool -h <HOST_NAME> -p <JMX_PORT> getcompactionthroughput
```

To increase compaction throughput to 64 MB/s, for example, use the following command:

```
nodetool -h <HOST_NAME> -p <JMX_PORT> setcompactionthroughput 64
```

Recovery

Depending on the replication factor and consistency levels of a Cassandra cluster configuration, the Knowledge Center CMS can handle the failure of one or more Cassandra nodes in the data center without any special recovery procedures and without interrupting service or losing functionality. When the failed node is back up, the Knowledge Center CMS automatically reconnects to it. If an eligible number of nodes have failed, you should just restart them.

If too many of the Cassandra nodes in your cluster have failed or stopped, you will lose functionality. To ensure a successful recover from failure of multiple nodes, Genesys recommends that you:

1. Stop every node, one at a time, with at least two minutes between operations.
2. Restart the nodes one at a time, with at least two minutes between operations.