



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.

Web Services and Applications Deployment Guide

Multiple Data Center Deployment

Contents

- 1 Multiple Data Center Deployment
 - 1.1 Overview
 - 1.2 Architecture
 - 1.3 Incoming traffic distribution
 - 1.4 Configuration
 - 1.5 GWS Cluster Management
 - 1.6 Limitations

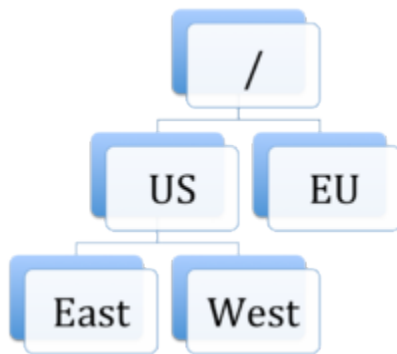
Multiple Data Center Deployment

Starting in release 8.5.2, GWS supports a deployment with multiple (two or more) data centers. This section describes this type of deployment.

Overview

A multiple data center deployment implies a logical partitioning of all GWS nodes into segregated groups that are using dedicated service resources, such as T-Server, StatServers, and so on.

The topology of a GWS Cluster can be considered as a standard directory tree where a leaf node is a GWS data center. The following diagram shows a GWS Cluster with 2 geographical regions (US and EU), and 3 GWS data centers (East and West in the US region, and EU as its own data center).



For data handling and distribution between GWS data centers, the following third-party applications are used:

- Cassandra—a NoSQL database cluster with multiple data centers with data replication between each other.
- Elasticsearch—a search engine which provides fast and efficient solution for pattern searching across Cassandra data. Genesys recommends that each GWS data center have an independent, standalone Elasticsearch cluster.

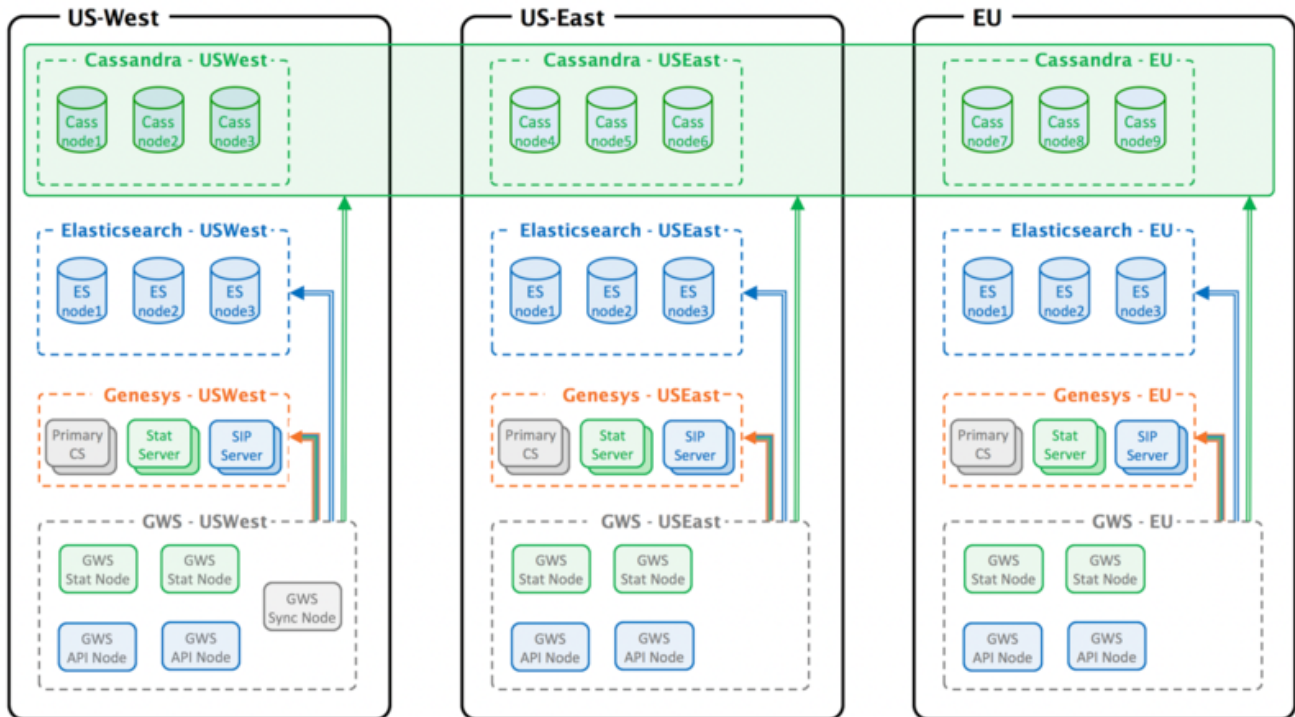
Architecture

A typical GWS data center in a multiple data center deployment consists of the following components:

- 2 GWS API nodes
- 2 GWS Stat nodes
- 3 Cassandra nodes

- 3 Elasticsearch nodes
- 1 GWS Sync node (only for Primary region)

The following diagram illustrates the architecture of this sample multiple data center deployment.



Note the following restrictions of this architecture:

- Only 1 Sync node is deployed within a GWS Cluster
- Each data center must have a dedicated list of Genesys servers, such as Configuration Servers, Stat Servers, and T-Servers.
- The Cassandra Keyspace definition must comply with the number of GWS data centers.
- Each GWS data center must have its own standalone and dedicated Elasticsearch Cluster.
- The GWS node identity must be unique across the entire Cluster.

Incoming traffic distribution

GWS does not support traffic distribution between GWS nodes natively. To enable this, any third-party reverse proxy can be used that can provide a session stickiness based on association with sessions on the backend server; this rule is commonly referred to as *Application-Controlled Session Stickiness*. In other words, when a GWS node creates a new session and returns Set-Cookie in response, the load-balancer should issue its own stickiness cookie. GWS uses a JSESSIONID cookie by default, but this can be reconfigured by using the following option in the **application.yaml** file:

```
jetty:
  cookies:
    name: <HTTP Session Cookie Name>
```

Configuration

This section describes the additional configuration required to set up a multiple data center deployment.

Important

In a DR environment (for instance, having two separate SIP Servers connected to a WS Cluster Application with different locations specified), it is possible that the **Place** can contain two DNs of the type **Extension**. However, the two DNs must belong to different switches and the API server will be mapped only to one of them during the **StartContactCenterSession** operation. In such cases, the DN visibility depends on the **nodePath** parameter in the **application.yaml** file in the API server and on the **location** attribute of the SIP Server connection in the WS Cluster Application.

Cassandra

Configure Cassandra in the same way as for a single data center deployment (described earlier in this document), making sure that the following conditions are met:

- All Cassandra nodes must have the same cluster name in **application.yaml**.
- The same data center name must be assigned to all Cassandra nodes across the GWS data center (specified in **cassandra-network.properties** or **cassandra-rackdc.properties**, depending on the Cassandra deployment).
- The Keyspace definition must be created based on **ks-schema-prod_HA.cql** from the Installation Package, changing only the following:
 - The name and ReplicationFactor of each.
 - The number of data centers between which the replication is enabled.

For example:

```
CREATE KEYSPACE sipfs WITH replication = {'class': 'NetworkTopologyStrategy', 'USWest': '3', 'USEast': '3', 'EU': '3'} AND durable_writes = true;
```

Genesys Web Services and Applications

The position of each node inside the GWS Cluster is specified by the mandatory property **nodePath** provided in **application.yaml**. The value of this property is in the standard file path format, and uses the forward slash (/) symbol as a delimiter. This property has the following syntax:

```
nodePath: <path-to-node-in-cluster>/<node-identity>
```

Where:

- <path-to-node-in-cluster> is the path inside the cluster with all logical sub-groups.
- <node-identity> is the unique identity of the node. Genesys recommends that you use the name of the host on which this data center is running for this parameter.

For example:

```
nodePath: /US/West/api-node-1
```

In addition to the configuration options set in the standard deployment procedure, set the following configuration options in **application.yaml** for all GWS nodes to enable the multiple data center functionality:

```
cassandraCluster:
  write_consistency_level: CL_LOCAL_QUORUM
  read_consistency_level: CL_LOCAL_QUORUM

serverSettings:
  nodePath: <path-to-node-in-cluster>/<node-identity>

statistics:
  locationAwareMonitoringDistribution: true
  enableMultipleDataCenterMonitoring: true
```

Important

- If the replication factor is changed in the keyspace definition (that is, if additional Cassandra nodes are added) then `replication_factor` in `cassandraCluster` of **application.yaml** should be adjusted to agree with the keyspace definition.
- Set both **locationAwareMonitoringDistribution** and **enableMultipleDataCenterMonitoring** on all nodes. Without them, the statistics information in a Multiple Data Center environment will not be displayed properly.

In addition, set the following options on all Stat nodes:

```
serverSettings:
  elasticSearchSettings:
    enableScheduledIndexVerification: true
    enableIndexVerificationAtStartup: true
```

GWS Sync Node

A Synchronization Node is a special node, and as indicated elsewhere in this Guide, in Deployment Guide, this node imports existing data from Configuration Server and keeps track of all changes.

Warning

Only one Sync Node can be running at any point of time even if you have two sync

nodes in your architecture. Genesys strongly recommends that you deploy Sync node in the same data center where the primary Configuration Server is located.

If any disaster causes this node to terminate or become unavailable because of network issues or the whole data center goes down, provisioning of any object in Configuration Server will not be reflected in the GWS cluster until the Sync node is recovered. Other functionality related to Agent activity is not affected in this case.

Configuration Server

The GWS Cluster Application object (typically named **CloudCluster**) in the Configuration Database must be configured with a specified location for each connection to Genesys servers, like Configuration Server, Stat Server, T-Server, and so on. This setting defines which server instance is used by the GWS node based on its position in the GWS Cluster. The visibility resource rule is based on comparing the **nodePath** attribute and the specified specification in connections.

Set these locations as Application Parameters of each connection, as follows:

```
locations=<path-to-node-in-cluster>
```

where <path-to-node-in-cluster> is the same path to the data center specified by the **nodePath** property in **application.yaml**.

For example:

```
locations=/US/West
```

GWS Cluster Management

Add a New Data Center

Before deploying new GWS nodes, you must extend the Cassandra cluster by adding new nodes into the data ring and updating the keyspace definition with a replication strategy for this new data center. Using the CQLSH utility, run the following command to update the existing Cassandra keyspace:

```
ALTER KEYSPACE sipfs WITH REPLICATION = {'class': 'NetworkTopologyStrategy', 'USWest': '3', 'USEast': '3', 'EU': '3', 'CA': '3'};
```

After you have deployed the new Cassandra data center, you can use the normal procedure to deploy additional GWS nodes.

Remove an Existing Data Center

Before removing a Cassandra data center, you must stop all GWS nodes in this data center to avoid writing data into Cassandra.

1. Stop all GWS nodes, and remove them if necessary.
2. Update the keyspace definition by removing the appropriate data center from the replication strategy. Use the same CQL command as you used for adding a new data center.
3. Run the following command on each Cassandra node in the data center being removed:

```
nodetool decommission
```
4. Stop all Cassandra nodes, and remove them if necessary.

Limitations

The cross-site monitoring is only supported via Team Communicator, using the WWE option [teamlead.monitoring-cross-site-based-on-activity-enabled](#).
The My Agents view is not supported for cross-site operations.