GENESYS™

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

Web Services and Applications 8.5.2

11/27/2023

Table of Contents

Web Services and Applications Deployment Guide

Welcome to the *Web Services and Applications Deployment Guide*. This document provides information about deploying Web Services API, Workspace Web Edition, and Gplus Adapters.

About Web Services and Applications

Planning your Web Services and Applications Deployment

Deploying Web Services and Applications

Configuring Web Services and Applications

Deploying and Configuring Gplus Adapters

Overview

Web Services and Applications is a set of REST APIs and user interfaces that provide a web-based client interface to access Genesys services. The following UIs are currently offered:

- Workspace Web Edition
- Gplus Adapter for Salesforce

To work with the API and these applications, you must first follow the steps in this Deployment Guide to install and configure the Web Services API server component of the product. After that, you can work directly with the APIs or provide specific configuration for the applications you plan to use.

Web Services API

Web Services are the REST APIs that can be used by developers to create custom agent applications that integrate with Genesys. These applications can include features such as state management, call control, supervisor monitoring, and call recording.

- Related documentation: Web Services API Reference

Workspace Web Edition

Workspace Web Edition is an HTML 5 thin-client application that provides agents and knowledge workers with non-intrusive access to the information, processes, and applications that they need to perform their jobs more efficiently and to ensure increased customer satisfaction.

- Related documentation: Web Services and Applications Configuration Guide, Workspace Web Edition Help, and Workspace Web Edition Developer's Guide and API Reference.

Gplus Adapter for Salesforce

Gplus Adapter for Salesforce is an integrated solution that enables Salesforce users to handle contact center interactions seamlessly within Salesforce.

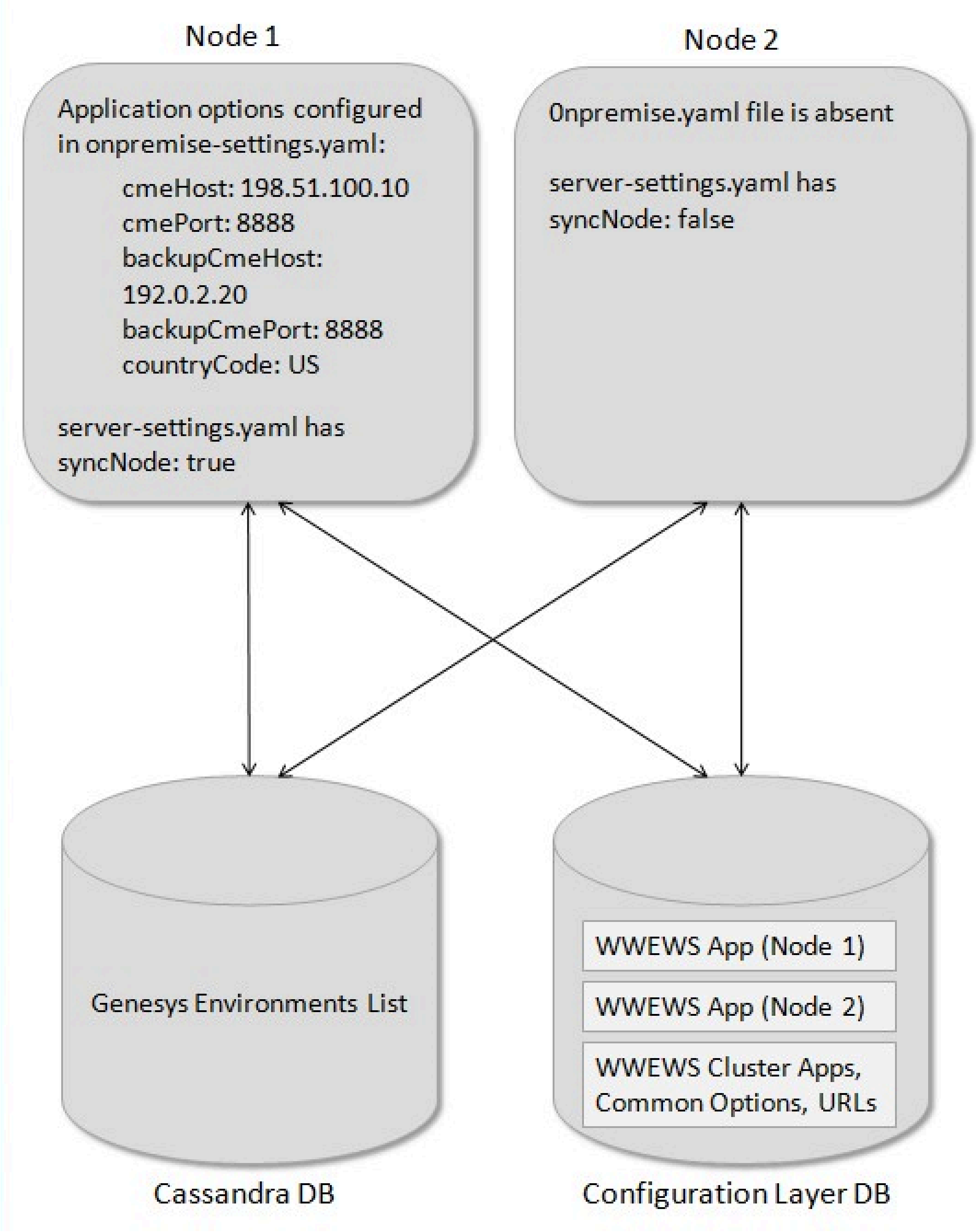
- Related documentation: Gplus Adapter for Salesforce section in this guide, Web Services and Applications Configuration Guide, and Gplus Adapters User Guide

Architecture

This configuration assumes that you have one contact center for each Web Services cluster. You can have any number of Client nodes. The following example depicts a two-node configuration:

- one node for the Synchronization Node (SyncNode)
- one Client node

Each node must have a connection to the Cassandra database and the Configuration Layer database.


Node 1
Node 2
Application options configured in onpremise-settings.yaml:
cmeHost: 198.51.100.10
cmePort: 8888
backupCmeHost: 192.0.2.20
backupCmePort: 8888
countryCode: US
server-settings.yaml has syncNode: true
0npremise.yaml file is absent
server-settings.yaml has syncNode: false
Genesys Environments List
Cassandra DB
WWEWS App (Node 1)
WWEWS App (Node 2)
WWEWS Cluster Apps, Common Options, URLs
Configuration Layer DB

Web Services On-Premise Architecture

Load balancing

Web Services supports any third-party load balancer that supports sticky sessions. You should configure session affinity (sticky sessions) based on `JSESSIONID`. Once your load balancer is set up, you can use the following URL for health checks: `http://<Web_Services_Host>:<Web_Services_Port>/api/v2/diagnostics/version`.

If you are configuring your solution to use Hypertext Transfer Protocol over Secure Socket Layer, you do not need to set up HTTPS between your load balancer and Web Services.

> **Important**
>
> Web Services and Applications does not currently support web sockets.

CometD

Web Services uses CometD version 3, an HTTP-based routing bus that uses an Ajax Push technology pattern known as Comet. Comet is a web application model that allows an HTTP request to push data to a browser, even if the browser has not requested it.

Web Services uses CometD to deliver unsolicited notifications to clients for real-time events, such as getting a new call or chat message. At runtime, CometD delivers messages, providing clients with a consistent approach while maintaining support for multiple browsers.

> **Important**
>
> Web Services and Applications does not currently support web sockets.

For details about CometD, see http://cometd.org/.

For details about how to configure CometD in Web Services, see cometDSettings.

Planning your deployment

Web Services requires the Cassandra NoSQL database management system, so when planning your deployment you also need to consider the Cassandra topology. Three common Cassandra topologies are generally used for Workspace Web Edition & Web Services:

- **Development:** One Cassandra node (appropriate for a development or lab environment)
- **Single Data Center:** One data center with a minimum of three Cassandra nodes
- **Two Data Centers:** Two data centers with a minimum of three Cassandra nodes in each data center

The number of Web Services nodes in your deployment scenario depends on a variety of factors. Please consult with Genesys for help to determine the correct number of nodes for your solution.

For development scenarios, you can install Cassandra and Web Services on the same host.

> ### Important
>
> - For single data center and two data center scenarios, you must install Cassandra and Web Services on separate hosts.
> - For production environments, each Genesys Web Services, Cassandra, and Elastic Search node must run on a dedicated Virtual Machine. For optimal performance, at least 4 cores and 16 GB of RAM for each VM with strict resources reservation is recommended.

Whichever deployment scenario you decide to implement, follow the same basic deployment steps:

Deployment tasks

Complete the following steps to deploy Web Services and Applications.

> ### Important
>
> If you are upgrading from one version of Web Services and Applications to another, see the Web Services and Applications Migration Guide.

1. Review the Prerequisites. Make sure all supporting components are installed and configured.

2. Install Web Services and Applications.
3. Initialize Cassandra.
4. Deploy the web application. If you are installing a version of Web Services and Applications that does not have Jetty embedded, go to Deploying the web application for 8.5.201.09 or earlier.
5. Configure Web Services.
6. Customize your required features, as necessary.
7. Configure additional security (optional).
8. Start and test Web Services, Workspace Web Edition, and Gplus Adapter for Salesforce.
9. Follow up with the appropriate configuration or developer information, depending on which components you plan to use:
    - Configure Workspace Web Edition
    - Work with Web Services (takes you to a new guide)
    - Deploy Gplus Adapter for Salesforce
    - Configure Gplus Adapter for Salesforce (takes you to a new guide)

Next step

- Review the Prerequisites

Prerequisites

To work with Web Services and Applications, your system must meet the software and browser requirements established in the Genesys Supported Operating Environment Reference Guide, and meet the following minimum requirements.

> **Important**
> In multi-tenant Configuration Server deployments, Web Services and Applications supports only one tenant that contains all configuration data.

Server Requirements

OS requirements

- Red Hat Enterprise Linux, version 6
- Red Hat Enterprise Linux, version 7

Java requirements

- If you are installing Web Services and Applications version 8.5.201.09 or earlier, you must also install the latest Java 1.7 JDK 64bit for Linux. For more information, refer to the Java documentation.
- If you are installing Web Services and Applications version 8.5.201.29 or later, you must also install Java JDK 1.8.0_92 or newer 64bit for Linux. For more information, refer to the Java documentation. If you're upgrading from JDK 7 to JDK 8, check out the Java Adoption Guide.
- If you are installing Web Services and Applications version 8.5.202.69 or later, you must also install OpenJDK 1.8.0_232 or newer 64bit for Linux. For more information, refer to OpenJDK documentation. To upgrade from Oracle JDK 8 to OpenJDK 8, refer to Migrating to OpenJDK 1.8.0.

Jetty requirements

- Jetty 9.2.14.v20151106 is now embedded in the Web Services and Applications installation. For Web Services and Applications version 8.5.201.09 or earlier, you must install and configure Jetty 9 separately.

Cassandra requirements

- Before you deploy Web Services and Applications, you must install and configure Cassandra. Web

Services supports Cassandra version 1.2 and 2.2.

Client Workstation Requirements

The following table shows the recommended hardware requirements for the client workstations.

Recommended Hardware Requirements - Client Workstation

Processor	Memory	Hard Drive	Graphic Card	Network
Intel Core 2 Duo CPU 2.8 GHz	2 GB	200 MB	DirectX 9.0+	xDSL / LAN

NOTE: 8 GB of RAM is recommended when non-Genesys applications are being run concurrently, or to improve performance.

Next step

* Installing Web Services and Applications

Installing and configuring Jetty

> **Important**
>
> Complete the steps on this page if you're installing Web Services and Applications version 8.5.201.09 or earlier; Jetty is embedded in newer versions of Web Services and Applications.

Jetty version 9 is a mandatory component that must be installed and configured on each Web Services node prior to starting the installation and configuration of Web Services and Applications.

> **Important**
>
> Jetty version 8 is no longer supported by Web Services. For more information about Jetty, refer to the Jetty documentation.

Install Jetty

Prerequisites

- You have installed the latest Java 1.7 JDK 64bit for Linux. For more information, refer to the Java documentation.

1. Download the latest Jetty version available in stable-9 from Eclipse.
2. Copy the Jetty archive to the installation directory. For example: `/opt/jetty`
3. Use a tar utility to extract the files. For example: `tar -zxvf jetty-distribution-9.2.1.v20140609.tar.gz`
4. Start Jetty to confirm it has been installed correctly:

   ```
   [java_path]/java -jar [jetty_path]/start.jar
   ```

   - [java_path] — The path to your Java installation. For example, `/usr/bin`.
   - [jetty_path] — The path to your Jetty installation. For example, `/opt/jetty`.

- Test Jetty by entering the following URL in a web browser: `http://[host]:8080`
  - [host] — The host name (fully qualified domain name) or IP address where you installed Jetty.

You should see a 404 error page from Jetty.

- Stop Jetty by pressing `Ctrl+c`.

> **Important**
>
> If you plan to use the Gplus Adapter for Salesforce or WebRTC for Workspace Web Edition, you should also configure Jetty for SSL.

Next step

- Installing and configuring Cassandra

Installing and configuring Cassandra

Before you start installing and configuring Web Services and Applications, first you have to install and configure Cassandra. Web Services support Cassandra version 1.2 or 2.2.

For new deployments starting from version 8.5.201.41, we recommend Cassandra 2.2. The procedures below are meant to serve as a quick guide on how to do this. For more detailed information, see the Cassandra 2.2 documentation.

For general instructions and guidelines, select one of the following links:

- Installing and Deploying Cassandra 2.2
- Upgrading to Cassandra 2.2

Installing and Deploying Cassandra 2.2

Installing Cassandra

Complete this procedure for each Cassandra node.

Prerequisites

- If you are using Web Services and Application v8.5.2.41 or earlier, a 2.1.4 version of the Cassandra distribution needs to be downloaded first. This package includes the cassandra-cli tool that you need to load the schema into your Cassandra 2.2 cluster.
- You have installed the latest Java SE Development Toolkit 8. For more information, refer to the Java documentation.

Start

1. Download the latest 2.2.x version of Cassandra.
2. Copy the Cassandra archive to the installation directory. For example, **/usr/local**
3. Use a tar utility to extract the files. For example, `tar -zxvf apache-cassandra-2.2.7-bin.tar.gz`
4. Add directories for data, commitlog, and saved_caches. You can create these directories anywhere or in the default locations configured in the ***Cassandra_install_dir*/conf/cassandra.yaml** file. For example:
   - **/var/lib/cassandra/data**
   - **/var/lib/cassandra/commitlog**
   - **/var/lib/cassandra/saved_caches**
5. Add a directory for logging. You can create this directory anywhere, such as **/var/log/cassandra/**.

End

Configuring Cassandra

The procedures below describe how to create the Cassandra keyspace for the following scenarios:

- Development: 1 Cassandra node (appropriate for a development or lab environment)
- Single Data Center: 1 data center with a minimum of three Cassandra nodes

> **Important**
>
> For more complex Cassandra deployments, please consult with Genesys

Select a tab below for the procedure that matches your deployment scenario.

Development

Configuring Cassandra (1 Cassandra node)

> **Important**
>
> The files modified in this procedure are typically found in the ***Cassandra_install_dir*/conf** directory.

Prerequisites

- Installing Cassandra

Start

1. Modify the **cassandra.yaml** file:
   a. Set seeds to the list of host name of the node. For example: `-seeds: “127.0.0.1”`
   b. Set `listen_address` and `rpc_address` to the host name.
   c. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of Installing Cassandra.
   d. Set the `start_rpc` parameter to `true`.
5. Save your changes and close the file.
6. Open the **log4j-server.properties** file and set the `log4j.appender.R.File` property to the directory you created in Step 5 of Installing Cassandra.
7. Save your changes and close the file.

End

Single Data Center

Configuring Cassandra (1 data center)

Complete the steps below for each node.

> Important
>
> The files modified in this procedure are typically found in the ***Cassandra_install_dir*/conf** directory.

Prerequisites

- Installing Cassandra

Start

1. Modify the **cassandra.yaml** file:
    a. Set the `cluster_name`. It must be the same name on all nodes.
    b. Set the `initial_token` according to the node's place in ring. It must be one of the following:

    ```
    Node #1:  -9223372036854775808
     Node #2:  -3074457345618258603
     Node #3:   3074457345618258602
    ```

    > Important
    >
    > The tokens shown here can be used for a three-node Cassandra cluster in a single data center. If you are using a different topology or cluster size, consult the Cassandra documentation.

    c. Set seeds to the list of host names of all nodes. For example: `-seeds: "node1, node2, node3"`
    d. Set `listen_address` and `rpc_address` to the host name.
    e. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of Installing Cassandra.
    f. Change `endpoint_snitch` to `PropertyFileSnitch`.

7. Save your changes and close the file.
8. Open the **log4j-server.properties** file and set the `log4j.appender.R.File` property to the directory you created in Step 5 of Installing Cassandra.
9. Save your changes and close the file.
10. Open the **cassandra-topology.properties** file and update for your cluster topology. For each node in your cluster, add the following line:

    ```
    [node]=[datacenter]:[rack]
    ```

Where:

- *[node]* is the IP address of the node.
- *[datacenter]* is the name of the data center for this node.
- *[rack]* is the name of the rack for this node.

The following is a sample **cassandra-topology.properties** file for a Single Data Center scenario:

```
192.0.2.10=datacenter1:rack1
 192.0.2.11=datacenter1:rack1
 192.0.2.12=datacenter1:rack1
```

11. Save your changes and close the file.

End

Verifying the Cassandra installation

Prerequisites

- Configuring Cassandra

Start

1. Start all Cassandra nodes using the following command: `Cassandra_install_dir/bin/cassandra`
2. Use the nodetool utility to verify that all nodes are connected by entering the following command: `Cassandra_install_dir/bin/nodetool -h Cassandra_host ring`

   The following is sample output for a Single Data Center scenario with three Cassandra nodes:

```
/genesys/apache-cassandra-1.2/bin$ ./nodetool ring
 Address     DC           Rack   Status  State   Load      Owns     Token
 192.0.2.10  datacenter1  rack1  Up      Normal  14.97 MB  100.00%  -9223372036854775808
 192.0.2.11  datacenter1  rack1  Up      Normal  14.97 MB  100.00%  -3074457345618258603
 192.0.2.12  datacenter1  rack1  Up      Normal  14.97 MB  100.00%   3074457345618258602
```

The following is sample output for a Development scenario with a single Cassandra node:

```
/genesys/apache-cassandra-2.2/bin$ ./nodetool ring
 Address         DC          Rack        Status State   Load            Effective-
 Ownership Token
 127.0.0.1       datacenter1 rack1       Up     Normal  1.89 MB
 100.00%             76880863635469966884037445232169973201
```

End

Next step

- Installing Web Services and Applications

Upgrading Cassandra to 2.2

Web Services support Cassandra versions 2.2 and 1.2. If you are using Cassandra 1.2, you can maintain this version or upgrade to Cassandra 2.2.

Directly upgrading from 1.2 to 2.2 is not supported, therefore you need to upgrade your Cassandra versions in several steps. For example 1.2 > 2.0 > 2.1 > 2.2. For more information about upgrading Cassandra, see Datastax documentation.

1. Stop all Web Services nodes.
2. Perform Cassandra upgrade according to the Datastax.
3. When configuring Cassandra 2.2 according to the Datastax instructions, you must enable the thrift interface. Set the the **start_rpc** parameter to `true` in the **cassandra.yaml** file
4. Start all Web Services nodes.

Installing

To install Web Services, first you need to set up the two application objects it uses in the Genesys configuration environment:

- An application of type `Genesys Generic Server` that is called the WS Cluster Application.
- An application of type `Genesys Generic Client` that is called the WS Node Application.

Configuring the Web Services applications

Perform the following procedures to configure the Web Services applications. Select a tab to configure the applications in *either* Configuration Manager or Genesys Administrator.

Configuration Manager

Creating and importing the application templates

The Web Services installation package includes a template for `Genesys Generic Server` (the WS Cluster Application), but you must create a new template for `Genesys Generic Client` (the WS Node Application).

Start

1. To create the `Genesys Generic Client` template, navigate to the **Application Templates** folder. Right-click and select **New > Application Template**.
2. Configure the **General** tab of the template as shown below:
   - Name: WS_Node
   - Type: Genesys Generic Client
   - Version: 8.5
   - State Enabled: Yes
3. Click **OK**.
4. To import the `Genesys Generic Server` template, navigate to the **Application Templates** folder in Configuration Manager. Right-click and select **Import Application Template**.
5. Navigate to the **templates** folder in your installation package.
6. Select the **Web_Services_and_Applications_852** template file.
7. Click **OK**.

End

Creating the WS Cluster Application

Start

1. Navigate to the **Applications** folder. Right-click and select **New > Application**.
2. Select the **Web_Services_and_Applications_852** template and click **OK**.
3. Configure the **General** tab as shown below:
   - Name: WS_Cluster
   - Template: Workspace_Web_Edition_Web_Services_852
   - Component Type: [Unknown]
   - State Enabled: Yes
4. On the **Tenants** tab, click **Add**.
   1. Choose the `Environment` tenant (or any other tenant that has a connection to your Configuration Server).
   2. Click **OK**.
5. On the **Server Info** tab, choose a `Host` object. See Create Host in the *Management Framework Deployment Guide* for more information about Host objects. This automatically adds a corresponding port entry. The port value is ignored by the server and does not need to be modified.
6. On the **Start Info** tab, add a "." to the `Working Directory`, `Command Line`, and `Command Line Arguments` fields. These values are mandatory for all applications and must be entered to save the application object. Web Services does not use these values, so the "." is used as a placeholder.
7. On the **Connections** tab, add the following connections:
   - Configuration Server
   - T-Server/SIP Server (if supporting voice)
   - Interaction Server (if supporting multimedia)
   - Universal Contact Server (if supporting multimedia)
   - Stat Server (if supporting reporting)

> ### Important
> KPIs and Statistics are reported only for the voice channel. Web Services does not support real-time statistics for mixed media (voice/multimedia) environments. If a mixed media environment is used, voice statistics are not accurate.

8. Click **OK** to save the `WS_Cluster` application.

End

Creating the WS Node Application

Start

1. Navigate to the **Applications** folder. Right-click and select **New > Application<**.
2. Select the `WS_Node` template and click **OK**.
3. Configure the **General** tab as shown below:
   - Name: WS_Node
   - Template: WS_Node
   - State Enabled: Yes
4. On the **Connections** tab, add the following connections:
   - Cluster application that was configured in the previous procedure.
5. Click **OK** to save the `WS_Node` application.

End

Genesys Administrator

Creating and importing the application templates

The Web Services installation package includes a template for `Genesys Generic Server` (the WS Cluster Application), but you must create a new template for `Genesys Generic Client` (the WS Node Application).

Start

1. To create the `Genesys Generic Client` template, navigate to **PROVISIONING > Environment> Application Templates**.
2. Select **New...** and configure the properties of the template as shown below:
   - Name: WS_Node
   - Type: Genesys Generic Client
   - Version: 8.5
   - State: Enabled
3. Click **Save & Close**.
4. To import the `Genesys Generic Server` template, click **Upload Template** in the **Tasks** panel. The **Click 'Add' and choose application template (APD) file to import** dialog opens.
5. Click **Add** and navigate to the **templates** folder in your installation package.
6. Select the **Web_Services_and_Applications_852** template file and click **Open**.
7. Click **Save & Close**.

End

Creating the WS Cluster Application

Start

1. Navigate to **PROVISIONING > Environment > Application** and click **New...**.
2. In the **General** section, configure the properties of the application as shown below:
   - Name: WS_Cluster
   - Template: Web_Services_and_Applications_852
   - State: Enabled
3. Add the following connections:
   - Configuration Server
   - T-Server/SIP Server (if supporting voice)
   - Interaction Server (if supporting multimedia)
   - Universal Contact Server (if supporting multimedia)
   - Stat Server (if supporting reporting)

   > ### Important
   > KPIs and Statistics are reported only for the voice channel. Web Services does not support real-time statistics for mixed media (voice/multimedia) environments. If a mixed media environment is used, voice statistics are not accurate.

4. In the **Server Info** section, select a `Tenant`:
   1. Click **Add**.
   2. Choose the `Environment` tenant (or any other tenant that has a connection to your Configuration Server).
   3. Click **OK**.
5. Choose a `Host` object. See Create Host in the *Management Framework Deployment Guide* for more information about Host objects.
6. Add a default Listening Port:
   1. Click **Add**.
   2. Enter the application's Port. For instance `7000`.
   3. Click **OK**.
7. Add a "." to the `Working Directory`, `Command Line`, and `Command Line Arguments` fields. These values are mandatory for all applications and must be entered to save the application object. Web Services does not use these values, so the "." is used as a placeholder.

End

Creating the WS Node Application

Start

1. Navigate to **PROVISIONING > Environment > Application** and click **New...**.
2. In the **General** section, configure the properties of the application as shown below:
   - Name: WS_Node
   - Template: WS_Node
   - State: Enabled
3. Add the following connections:
   - Cluster application that was configured in the previous procedure.
4. Click **Save & Close**.

End

Next step

- Deploying the web application

Deploying the web application

Important

- If you're deploying Web Services and Applications version 8.5.201.50 or earlier, complete the steps on Deploying the web application for 8.5.201.50 or later instead.
- If you're deploying Web Services and Applications version 8.5.201.09 or earlier, complete the steps on Deploying the web application for 8.5.201.09 or earlier instead.

Complete the following steps for each Web Services node.

Start

1. Copy the **installation_CD/ gws-<version>.noarch.rpm** file to a local folder.
2. Perform the installation using the RPM package manager:
   ```
   rpm -ivh [--prefix installation_location] gws-<version>.noarch.rpm
   ```
   If --prefix is not specified, the default installation location is **/usr/share** folder.
3. Copy the *.sample files from the **/installation_path/gws/config/config-templates** directory to the **/installation_path/gws/config** directory and remove the **.sample** extension.
4. Run following command to register the new service on your host: `chkconfig gws on`

End

Next Steps

1. Configuring Web Services
2. Initializing Cassandra
3. Starting and Testing Web Services

Installation package files

Web Services configuration and initialization files (Red Hat Enterprise Linux 6)

File/folder name	Description
/etc/default/gws	The configuration file for the Web Services service.
/etc/init.d/gws	The initialization script for the Web Services service.

Web Services configuration and initialization files (Red Hat Enterprise Linux 7)

File/folder name	Description
/usr/bin/gws	The initialization script for the Web Services service.
/usr/lib/systemd/system/gws.service	The default configuration file for the Web Services service. The following scripts extend service functionality on the Red Hat Enterprise Linux 7 platform: • /usr/libexec/initscripts/legacy-actions/gws/config • /usr/libexec/initscripts/legacy-actions/gws/version
/usr/lib/systemd/system/gws.service.d/gws.conf	The custom configuration file for the Web Services service. The settings defined in this file override the default settings.

Web Services and Applications files

File/folder name	Description
/installation_path/gws	The home directory of Web Services and Applications.
/installation_path/gws/gws.jar	The application .jar file.

GWS Configuration files

File/folder name	Description
/installation_path/gws/config	The folder containing configuration files of Web Services and Applications, including: • /installation_path/gws/config/feature-definitions.json

File/folder name	Description
	• /installation_path/gws/config/hystrix.properties • /installation_path/gws/config/logback.xml
/installation_path/gws/config-templates	The folder that contains templates of the Web Services and Applications configuration files. After installation, files from the folder must be modified according to *Web Services and Applications Configuration Guide* and moved to **/installation_path/gws/config**. The templates include: • /installation_path/gws/config-templates/application.yaml.sample • /installation_path/gws/config-templates/elasticsearch.yml.sample • /installation_path/gws/config-templates/statistics.yaml.sample

Cassandra schema

File/folder name	Description
/installation_path/gws/data	The folder that contains scripts to deploy database schema. Files include: • Scripts for Cassandra schema deployment using the **cqlsh** tool: • /installation_path/gws/data/ks-schema-local.cql • /installation_path/gws/data/ks-schema-prod.cql • /installation_path/gws/data/ks-schema-prod_HA.cql • /installation_path/gws/data/cf-schema.cql • Scripts for Cassandra schema deployment using the **cassandra-cli** tool: • /installation_path/gws/data/ks-schema-local.txt • /installation_path/gws/data/ks-schema-prod.txt • /installation_path/gws/data/ks-schema-prod_HA.txt • /installation_path/gws/data/cf-schema.txt

Elastic search templates and routing templates

- /installation_path/gws/elasticsearch
- /installation_path/gws/routing-templates

Deploying the web application for 8.5.201.50 or later

> **Important**
>
> If you're deploying Web Services and Applications version 8.5.201.09 or earlier, complete the steps on Deploying the web application for 8.5.201.09 or earlier instead.

As of version 8.5.201.18, you can install Web Services as a service. Complete the following steps for each Web Services node.

Start

1. Create a new folder on your Web Services node. For example, **web-services**. This is the home folder for the web application.
2. Copy the **gws.jar** file from the installation CD to your new Web Services home folder.
3. Copy the ***installation_CD*/etc/default/gws** file to the **/etc/default** folder on your Web Services host.
4. Copy the ***installation_CD*/etc/init.d/gws** file to the **/etc/init.d** folder on your host.
5. Open **/etc/default/gws** on your host and update the following environment variables to values appropriate for your Web Services node:
   - `GWS_HOST` — This value should match the Jetty host you'll define later in the jetty section of the **application.yaml** configuration file.
   - `GWS_PORT` — This value should match the Jetty port you'll define later in the jetty section of the **application.yaml** configuration file.
   - `GWS_HOME` — The Web Services home folder you created in Step 1.
   - `GWS_LOGS` — The location where you want Web Services to store log files. If this folder doesn't exist, Web Services creates it during application startup.
   - `GWS_TEMP` — The location where you want Web Services to store temp files. If this folder doesn't exist, Web Services creates it during application startup.
   - `GWS_CONF` — The location where you want to store the Web Services configuration files. If you don't specify a value for `GWS_CONF`, Web Services uses `GWS_HOME`/**etc**.
6. Create the `GWS_CONF` folder you specified in Step 5. If you didn't set `GWS_CONF`, create a folder called **etc** in `GWS_HOME` — for example, **web-services/etc**.
7. Create the following configuration files in the folder you created in Step 6. You can simply copy the files from ***installation_CD*/config-templates** and remove the **.sample** extension. You'll learn more about the settings in these files as you go through the configuration steps for Web Services and its features later in this guide.
   - **application.yaml**
   - **elasticsearch.yml**

- **hystrix.properties**
- **logback.xml**
- **statistics.yaml**

8. Copy the **routing-templates** folder from ***installation_CD*/routing-templates** to your `GWS_HOME` folder.
9. Create the user group **gws**.
10. Create the **gws** user in the **gws** user group. The user should have read and write permissions for the folders defined in the `GWS_HOME`, `GWS_LOGS`, `GWS_TEMP`, and `GWS_CONF` environment variables.
11. Use the following command to register the new service on your host: `chkconfig gws on`

End

Next Step

- Configuring Web Services

Deploying the web application for 8.5.201.09 or earlier

> **Important**
>
> Complete the steps on this page if you're installing Web Services and Applications version 8.5.201.09 or earlier; otherwise, go to Deploying the web application.

The Web Services web application uses the Jetty root context. If other web applications served by the same instance of Jetty also use the root context, this can prevent the Web Services web application from getting routed requests. If you are working with a fresh install of Jetty, you should remove the default Jetty files from the **$JETTY_HOME/webapps** and **$JETTY_HOME/contexts** folders.

Complete the following steps for each Web Services node.

Start

1. Stop Jetty.
2. Copy the **jetty.xml** file from ***installation_CD*/jetty** to **$JETTY_HOME/etc**.
3. Copy the **cloud-web.xml** file from ***installation_CD*/jetty** to **$JETTY_HOME/webapps**.
4. Copy the **logback.xml** file from ***installation_CD*/jetty** to **$JETTY_HOME/resources**.
5. Copy the **cloud-web.war** file from ***installation_CD*/webapp** to **$JETTY_HOME/webapps**.
6. Copy the configuration files from ***installation_CD*/conf** to a local folder of your choosing. For example, **$JETTY_HOME/genconfig**.

End

Next Step

- Configuring Web Services for 8.5.201.09 or earlier

Initializing Cassandra

Creating the Cassandra keyspace

The procedures below describe how to create the Cassandra keyspace for the following scenarios:

- Development — 1 Cassandra node (appropriate for a development or lab environment)
- Single Data Center — 1 data center with a minimum of three Cassandra nodes
- Two Data Centers — 2 data centers with a minimum of three Cassandra nodes in each data center

> **Important**
>
> For more complex Cassandra deployments, please consult with Genesys.

Select a tab below for the procedure that matches your deployment scenario.

Development

Creating the Cassandra keyspace (1 Cassandra node)

Start

1. Copy the **ks-schema-local.cql** file from ***/installation_path/gws/data*** to the Cassandra node host.
2. By default, the replication factor is set to 1. Since this is a single node deployment, you don't need to modify this value. Refer to the Cassandra documentation for more information about replication factors.

   ```
   and strategy_options = {replication_factor : 1}
   ```

3. Create the Cassandra schema. Choose one of the following options:
   - If you are using Web Services and Applications v8.5.2.## or later, run the following command: `cqlsh` *`cassandra_host`* `--file ks-schema-local.cql`
   - If you are using Web Services and Applications v8.5.2.41 or earlier, run the following command: *`cassandra_install_dir`*`/bin/cassandra-cli -h` *`cassandra_host`* `--file ks-schema-local.txt`
     ...where *`cassandra_host`* is the host name (fully qualified domain name) or IP address of the Cassandra node.

End

Single Data Center

Creating the Cassandra keyspace (1 data center)

Complete the following procedure on one node in your Cassandra cluster.

Start

1. Copy the **ks-schema-prod.cql** file from ***/installation_path/gws/data*** to the Cassandra node host.
2. For fault tolerance, Genesys recommends that you use at least 3 Cassandra nodes and set the replication factor to 3. Refer to the Cassandra documentation for more information about replication factors. To modify this value, change the following line:

```
and strategy_options = {replication_factor : <replication-factor-in-your-lab>}
```

3. Create the Cassandra schema. Choose one of the following options:
   - If you are using Web Services and Applications v8.5.2.## or later, run the following command: `cqlsh cassandra_host --file ks-schema-prod.cql`
   - If you are using Web Services and Applications v8.5.2.41 or earlier, run the following command: `cassandra_install_dir/bin/cassandra-cli -h cassandra_host --file ks-schema-prod.txt`

   cassandra_host is the host name (fully qualified domain name) or IP address of the Cassandra node.

End

Two Data Centers

Creating the Cassandra keyspace (2 data centers)

Complete the following procedure on one node in your Cassandra cluster.

Start

1. Copy the **ks-schema-prod_HA.cql** file from ***/installation_path/gws/data*** to the Cassandra node host.
2. Modify the following line:

```
with strategy_options ={ AZ1 : 3, AZ2 : 3 }
```

   a. Add the data center name. You can use nodetool to find the name of the data center by examining the output of "nodetool ring" (the tool is located in the **bin** directory of Cassandra). The following is sample output from the nodetool:

```
nodetool ring
 Address        DC           Rack   Status  State   Load      Owns     Token
 192.0.2.10     datacenter1  rack1  Up      Normal  14.97 MB  100.00%  0
 198.51.100.10  datacenter2  rack1  Up      Normal  14.97 MB  100.00%  100
 192.0.2.11     datacenter1  rack1  Up      Normal  14.97 MB  100.00%
 56713727820156410577229101238628035242
```

```
198.51.100.11  datacenter2  rack1  Up      Normal  14.97 MB  100.00%
56713727820156410577229101238628035242
192.0.2.12     datacenter1  rack1  Up      Normal  14.97 MB  100.00%
113427455640312821154458202477256070484
198.51.100.12  datacenter2  rack1  Up      Normal  14.97 MB  100.00%
113427455640312821154458202477256070484
```

b. Add the replication factor. Refer to the Cassandra documentation for more information about replication factors.

Based on the nodetool output above, your line might be:

```
with strategy_options ={ datacenter1 : 3, datacenter2 : 3 }
```

3. Create the Cassandra schema. Choose one of the following options:
   - If you are using Web Services and Applications v8.5.2.## or later, run the following command: `cqlsh cassandra_host --file ks-schema-prod_HA.cql`
   - If you are using Web Services and Applications v8.5.2.41 or earlier, run the following command: `cassandra_install_dir/bin/cassandra-cli -h cassandra_host --file ks-schema-prod_HA.txt`

   cassandra_host is the host name (fully qualified domain name) or IP address of the Cassandra node.

End

Creating the column families

Complete the following procedure on one node in your Cassandra cluster.

Start

1. Copy the **cf-schema.cql** file from ***/installation_path/gws/data*** to the Cassandra node host.
2. Run one of the following commands to create the Cassandra schema:
   - If you are using Web Services and Applications v8.5.2.## or later, run the following command:

     ```
     cqlsh cassandra_host --file cf-schema.cql
     ```
3. If you are using Web Services and Applications v8.5.2.41 or earlier, run the following command:

   ```
   cassandra_install_dir/bin/cassandra-cli -h cassandra_host --file
   cf-schema.txt
   ```

End

Next step

- Starting and Testing Web Services

Support for Interaction Server Cluster

Web Services supports Interaction Server Cluster deployment for load balancing. For more information on cluster deployments, suggested configurations, and the configuration procedure, see Interaction Server Cluster.

Special considerations

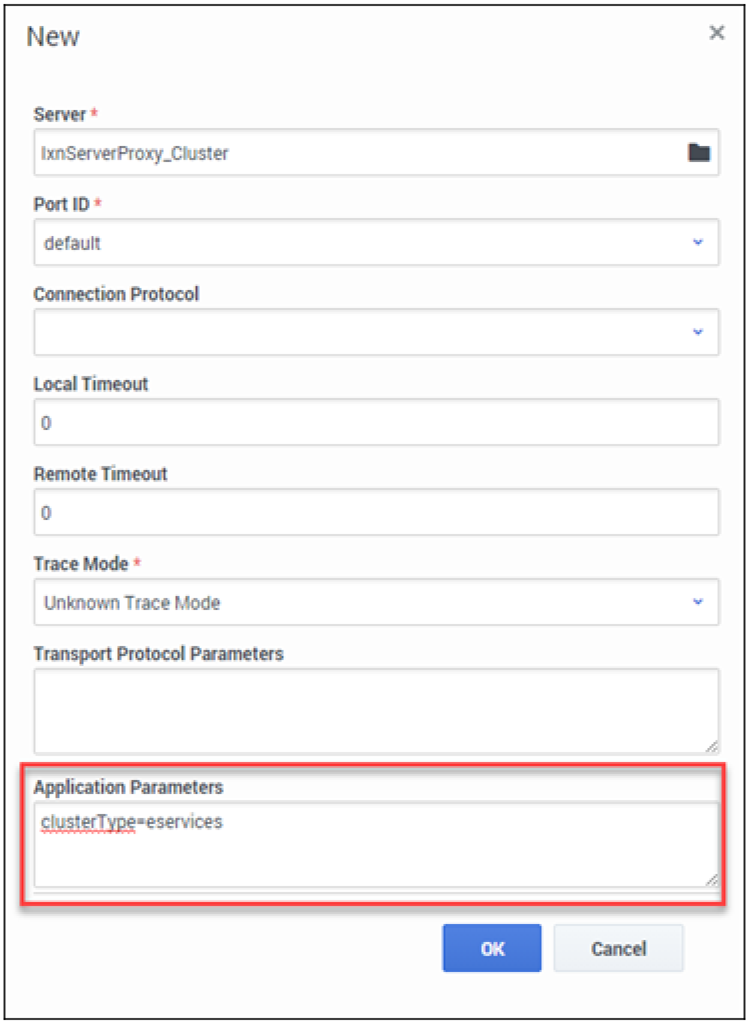
The Cloud Cluster can have one of the following connections:

- Direct connections to Interaction Server Pair
- Connection to an Application Cluster that contains a list of Interaction Server Proxies

The Cloud Cluster cannot support both connection options listed above.

When you configure the connection, the **Application Parameters** field must include the following string:

```
clusterType=eservices
```



New

Server *
IxnServerProxy_Cluster

Port ID *
default

Connection Protocol

Local Timeout
0

Remote Timeout
0

Trace Mode *
Unknown Trace Mode

Transport Protocol Parameters

Application Parameters
clusterType=eservices

Support for Universal Contact Server Cluster

Web Services supports a Universal Contact Server Cluster deployment for load balancing. For more information on cluster deployments, suggested configurations, and the configuration procedure, please see Universal Contact Server Cluster and see Creating the UCS 9.1 Cluster Application Object.

Special Considerations

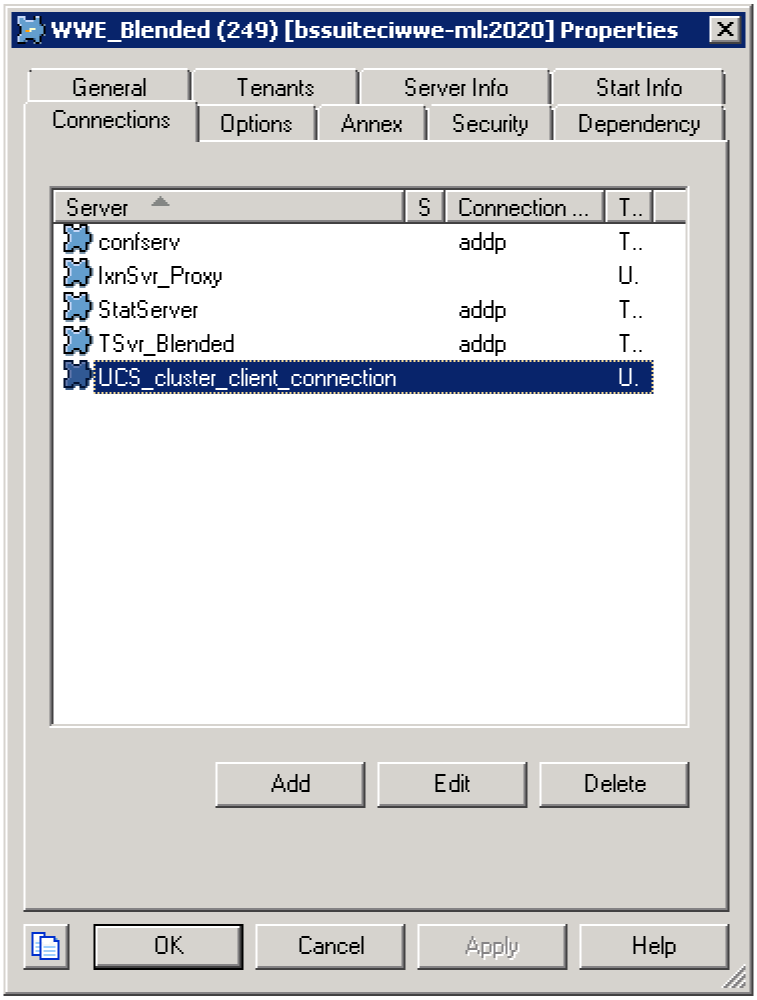
The Cloud Cluster can have one of the following connections:

- Direct connections to Universal Contact Server pair
- Connections to an Application Cluster that contains a list of Universal Contact Servers 9.x

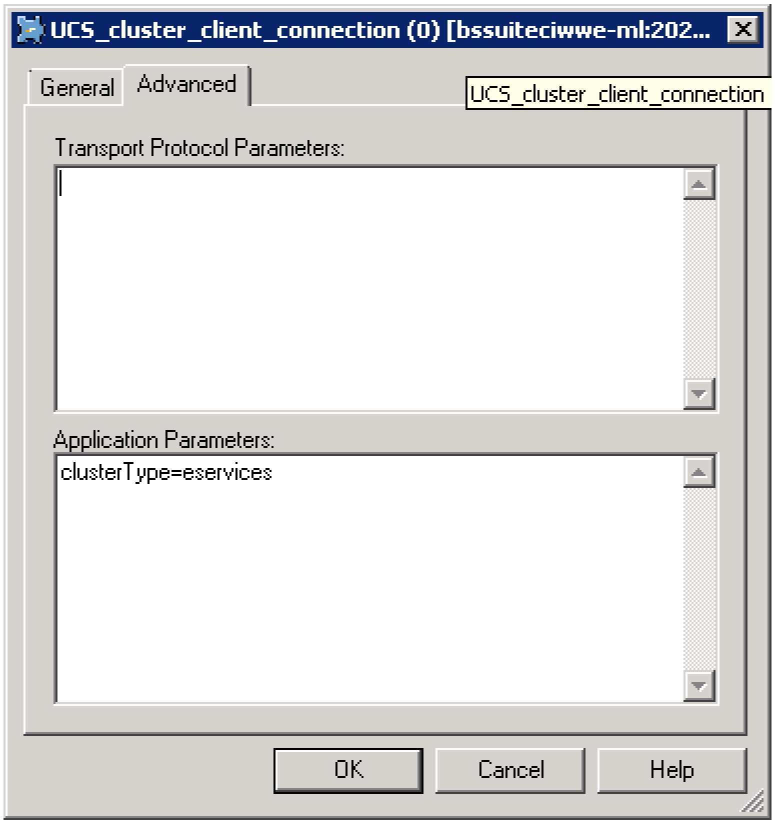
The Cloud Cluster cannot support both connections listed above.

When you configure the connections, the **Applications Parameters** field must include the following string:

```
clusterType=eservices
```



Server	S	Connection ...	T..
confserv		addp	T..
IxnSvr_Proxy			U.
StatServer		addp	T..
TSvr_Blended		addp	T..
UCS_cluster_client_connection			U.

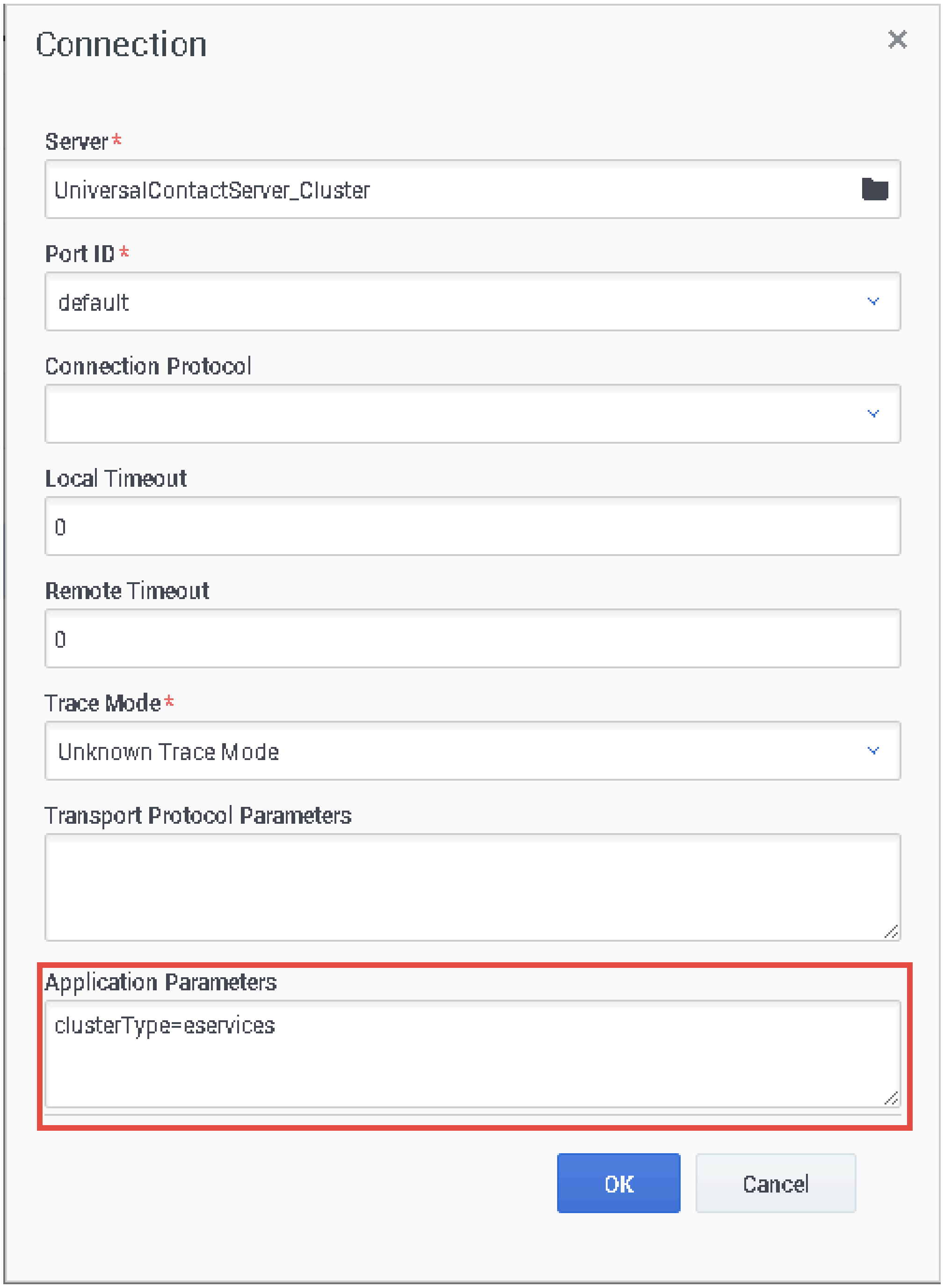


UCS_cluster_client_connection (0) [bssuiteciwwe-ml:202...

Transport Protocol Parameters:

Application Parameters:

clusterType=eservices



Server*
UniversalContactServer_Cluster

Port ID*
default

Connection Protocol

Local Timeout
0

Remote Timeout
0

Trace Mode*
Unknown Trace Mode

Transport Protocol Parameters

Application Parameters
clusterType=eservices

Configuring Web Services settings

> **Important**
>
> If you are deploying Web Services and Applications version 8.5.201.09 or earlier, complete the steps on Configuring Web Services for 8.5.201.09 or earlier instead.

As part of Deploying the web application, you created the **application.yaml** file (or Web Services created it for you). To configure basic Web Services and Applications settings, you need to update the **application.yaml** file on each of your Web Services nodes. In later topics, you'll learn more about modifying this file to configure additional features and security. For now, review the contents below for details about each section in the **application.yaml** configuration file.

Logging settings

The purpose of the **logging** section is to tell Web Services where to find the **logback.xml** file you created (or Web Services created for you) as part of Deploying the web application and where to save logs.

The **application.yaml.sample** file includes the following default **logging** section:

```
logging:
  config: logback.xml
  file: cloud.log
  path: /var/log/jetty9
```

See logging for details about all supported configuration settings for this section.

Jetty settings

You use the **jetty** section of the **application.yaml.sample** file to tell Web Services how Jetty should behave. The **application.yaml.sample** file includes the following default **jetty** section:

```
jetty:
  host:
  port: 8090
  idleTimeout: 30000
  soLingerTime: -1
  sessionMaxInactiveInterval: 1800
  enableWorkerName: true
  enableRequestLog: true
  requestLog:
    filename: yyyy_mm_dd.request.log
    filenameDateFormat: yyyy_MM_dd
    logTimeZone: GMT
    retainDays: 90
    append: true
    extended: false
    logCookies: false
    logLatency: true
    preferProxiedForAddress: true
  enableSsl: false
```

See jetty for details about supported configuration settings for this section.

Cassandra cluster settings

The settings in the **cassandraCluster** section correspond to the contents of the **cassandra-cluster.yaml** file in version 8.5.201.09 or earlier of Web Services and Applications. This section tells Web Services how your Cassandra cluster should be managed and accessed.

The **application.yaml.sample** file includes the following default **cassandraCluster** section:

```
cassandraCluster:
  thrift_port: 9160
  jmx_port: 7199
  nodes: [ToBeChanged: <CASSANDRA_PRIMARY_DC_NODES>]
  backup_nodes: [ToBeChanged: <CASSANDRA_BACKUP_DC_NODES>]
  replication_factor: [ToBeChanged: <REPLICATION_FACTOR>]
  write_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
"CL_QUORUM" for single-DC env.]
  read_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
"CL_QUORUM" for single-DC env.]
  max_conns_per_host: 16
  max_cons: 48
  max_pending_conns_per_host: 80
  max_blocked_threads_per_host: 160

  cassandraVersion: [ToBeChanged: "1.1" | "1.2"]
  useSSL: [ToBeChanged: sopporting only for 1.2 Cassandra "false" | "true"]
```

Make sure that you update all settings marked as `[ToBeChanged]`. See cassandraCluster for details about all supported configuration settings for this section.

Server settings

The settings in the `serverSettings` section correspond to the contents of the **server-settings.yaml** file in version 8.5.201.09 or earlier of Web Services and Applications. This section provides the core settings Web Services needs to run your node.

The **application.yaml.sample** file includes the following default **serverSettings** section:

```
serverSettings:
  # URLs
  externalApiUrlV2: [ToBeChanged: public URL including protocol, address and port,
<PUBLIC_SCHEMA_BASE_URL>]/api/v2
  internalApiUrlV2: [ToBeChanged: internal URL including protocol, address and port,
<INTERNAL_SCHEMA_BASE_URL>]/internal-api
```

```
  undocumentedExternalApiUrl: [ToBeChanged: public URL including protocol, address and port,
<PUBLIC_SCHEMA_BASE_URL>]/internal-api

  # Paths
  pathPrefix: [ToBeChangedOrRemoved: <PATH_PREFIX>]
  internalPathPrefix: [ToBeChangedOrRemoved: <INTERNAL_PATH_PREFIX>]

  # General
  iwsDispositionCodeSync: [ToBeChanged: "true"|"false"]
  temporaryAuthenticationTokenTTL: [ToBeChangedOrRemoved:
<TEMPORARY_AUTHENTICATION_TOKEN_TTL>]
  enableCsrfProtection: [ToBeChanged: "true"|"false"]
  salesforceAuthenticationMode: [ToBeChanged: "true"|"false"]
  enableOpenIDConnect: [ToBeChanged: "true"|"false"]

  # Timeouts
  activationTimeout: 12000
  configServerActivationTimeout: 35000
  configServerConnectionTimeout: 15000
  connectionTimeout: 4000
  contactCenterSynchronizationTimeout: 60000
  inactiveUserTimeout: [ToBeChangedOrRemoved: <INACTIVE_USER_TIMEOUT>]
  reconnectAttempts: 1
  reconnectTimeout: 10000

  # OPS account
  opsUserName: [ToBeChanged: <OPS_USER_NAME>]
  opsUserPassword: [ToBeChanged: <OPS_USER_PASSWORD>]

  # Configuration Server credentials
  applicationName: Cloud
  applicationType: CFGGenericClient
  cmeUserName: [ToBeChanged: <CONFIG_SERVER_USER_NAME>]
  cmePassword: [ToBeChanged: <CONFIG_SERVER_USER_PASSWORD>]
  syncNode: [ToBeChanged: "true"|"false"]
  synchronizationCmeEventsPrefilterEnabled: [ToBeChanged: "true"|"false"]
  enableVirtualQueueSynchronization: [ToBeChanged: "true"|"false"]

  # Statistics
  statConnectionTimeout: [ToBeChangedOrRemoved: <STAT_CONNECTION_TIMEOUT>]
  statReconnectAttempts: [ToBeChangedOrRemoved: <STAT_RECONNECT_ATTEMPTS>]
  statReconnectTimeout: [ToBeChangedOrRemoved: <STAT_RECONNECT_TIMEOUT>]
  statOpenTimeout: [ToBeChangedOrRemoved: <STAT_OPEN_TIMEOUT>]
  statisticsWritesCL: [ToBeChangedOrRemoved: <STATISTICS_WRITE_SCL>]
  reportingSyncInterval: [ToBeChangedOrRemoved: <REPORTING_SYNC_INTERVAL>]
  enableElasticSearchIndexing: [ToBeChanged: "true"|"false"]
  statisticsOpenRetryInterval: [ToBeChangedOrRemoved: <STATISTICS_OPEN_RETRY_INTERVAL>]

  # Multi regional supporting
  nodePath: [ToBeChanged: node position in cluster, example: /<REGION>/HOST
  nodeId: [ToBeChangedOrRemoved: unique value in cluster <NODE_ID>]

  # SSL and CA
  caCertificate: [ToBeChangedOrRemoved: <CA_CERTIFICATE>]
  jksPassword: [ToBeChangedOrRemoved: <JKS_PASSWORD>]

  # SAML
  samlSettings:
    encryptionKeyName: [ToBeChangedOrRemoved: <SAML_ENCRYPTION_KEY_NAME>]
    signingKeyName: [ToBeChangedOrRemoved: <SAML_SIGNING_KEY_NAME>]
    identityProviderMetadata: [ToBeChangedOrRemoved: <SAML_IDENTITY_PROVIDER_METADATA>]
    serviceProviderEntityId: [ToBeChangedOrRemoved: <SAML_SERVICE_PROVIDER_ENTITY_ID>]
    encryptionKeyPassword: [ToBeChangedOrRemoved: <SAML_ENCRYPTION_KEY_PASSWORD>]
    signingKeyPassword: [ToBeChangedOrRemoved: <SAML_SIGNING_KEY_PASSWORD>]
    tlsKeyName: [ToBeChangedOrRemoved: <SAML_TLS_KEY_NAME>]
    tlsKeyPassword: [ToBeChangedOrRemoved: <SAML_TLS_KEY_PASSWORD>]
    responseSkewTime: [ToBeChangedOrRemoved: <SAML_RESPONSE_SWEW_TIME>]

  # CORS
  crossOriginSettings:
    allowedOrigins: [ToBeChangedOrRemoved: <CROSS_ALLOWED_ORIGINS>]
    allowedMethods: [ToBeChangedOrRemoved: <CROSS_ALLOWED_METHODS>]
    allowedHeaders: [ToBeChangedOrRemoved: <CROSS_ALLOWED_HEADERS>]
    exposedHeaders: [ToBeChangedOrRemoved: <CROSS_EXPOSED_HEADERS>]
    allowCredentials: [ToBeChangedOrRemoved: <CROSS_ALLOW_CREDENTIALS>]
    corsFilterCacheTimeToLive: [ToBeChangedOrRemoved:
<CROSS_ORIGIN_CORS_FILTER_CACHE_TIME_TO_LIVE>]

  # Elastic Search
  elasticSearchSettings:
    clientNode: [ToBeChangedOrRemoved: "true"|"false"]
    indexPerContactCenter: [ToBeChangedOrRemoved: "true"|"false"]
    enableScheduledIndexVerification: [ToBeChangedOrRemoved: "true"|"false"]
    indexVerificationInterval: [ToBeChangedOrRemoved:
<ELASTIC_SEARCH_INDEX_VERIFICATION_INTERVAL>]
    retriesOnConflict: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_RETRIES_ON_CONFICT>]
    waitToIndexTimeout: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_WAIT_TO_INDEX_TIMEOUT>]
    enableIndexVerificationAtStartUp: [ToBeChangedOrRemoved: "true"|"false"]

  # Caching Settings
  cachingSettings:
    enableSystemWideCaching: [ToBeChangedOrRemoved: "true"|"false"]
    agentStatesTTL: [ToBeChangedOrRemoved: <CACHING_AGENT_STATES_TTL>]
    businessAttributesTTL: [ToBeChangedOrRemoved: <CACHING_BUSINESS_ATTRIBUTES_TTL>]
    transactionsTTL: [ToBeChangedOrRemoved: <CACHING_TRANSACTIONS_TTL>]
    skillsTTL: [ToBeChangedOrRemoved: <CACHING_SKILLS_TTL>]
    virtualAgentGroupsTTL: [ToBeChangedOrRemoved: <CACHING_VIRTUAL_AGENT_GROUPS_TTL>]
    contactCenterFeaturesTTL: [ToBeChangedOrRemoved: <CACHING_CONTACT_CENTER_FEATURES_TTL>]
    contactCenterSettingsTTL: [ToBeChangedOrRemoved: <CACHING_CONTACT_CENTER_SETTINGS_TTL>]
    voiceContextCaching: [ToBeChangedOrRemoved: "true"|"false"]
    voiceContextRefreshInterval: [ToBeChangedOrRemoved:
<CACHING_VOICE_CONTEXT_REFRESH_INTERVAL>]

    dedicatedCacheSettings:
      - cacheName: ContactServerCategoriesCache
        timeToLiveSeconds: [ToBeChangedOrRemoved: <TTL_FOR_CATEGORIES_CACHE>]
        maxEntriesLocalHeap: [ToBeChangedOrRemoved: <MAX_LOCAL_HEAP_FOR_CATEGORIES_CACHE>]
      - cacheName: ContactServerStandardResponsesCache
        timeToLiveSeconds: [ToBeChangedOrRemoved: <TTL_FOR_STANDARD_RESPONSES_CACHE>]
        maxEntriesLocalHeap: [ToBeChangedOrRemoved:
<MAX_LOCAL_HEAP_FOR_STANDARD_RESPONSES_CACHE>]

  # DoS Filter Settings
  enableDosFilter: [ToBeChanged: "true"|"false"]
  dosFilterSettings:
    maxRequestsPerSec: [ToBeChangedOrRemoved: <DOS_FILTER_MAX_REQUESTS_PER_SEC>]
    delayMs: [ToBeChangedOrRemoved: <DOS_FILTER_DELAY_MS>]
    maxWaitMs: [ToBeChangedOrRemoved: <DOS_FILTER_MAX_WAIT_MS>]
    throttledRequests: [ToBeChangedOrRemoved: <DOS_FILTER_THROTTLED_REQUESTS>]
    throttleMs: [ToBeChangedOrRemoved: <DOS_FILTER_THROTTLE_MS>]
    maxRequestMs: [ToBeChangedOrRemoved: <DOS_FILTER_MAX_REQUEST_MS>]
    maxIdleTrackerMs: [ToBeChangedOrRemoved: <DOS_FILTER_MAX_IDLE_TRACKER_MS>]
    insertHeaders: [ToBeChangedOrRemoved: <DOS_FILTER_INSERT_HEADERS>]
    trackSessions: [ToBeChangedOrRemoved: <DOS_FILTER_TTACK_SESSIONS>]
    remotePort: [ToBeChangedOrRemoved: <DOS_FILTER_REMOTE_PORT>]
    ipWhitelist: [ToBeChangedOrRemoved: <DOS_FILTER_IP_WHITE_LIST>]
```

```
# Statistics Settings

# Account Management
accountManagement:
  forgotPasswordEmailTemplate:
    from: [ToBeChangedOrRemoved: <PASSWORD_MESSAGE_FROM>]
    subject: [ToBeChangedOrRemoved: <PASSWORD_MESSAGE_SUBJECT>]
    body: [ToBeChangedOrRemoved: <PASSWORD_MESSAGE_BODY>]
  accountCreatedEmailTemplate:
    from: [ToBeChangedOrRemoved: <ACCOUNT_MESSAGE_FROM>]
    subject: [ToBeChangedOrRemoved: <ACCOUNT_MESSAGE_SUBJECT>]
    body: [ToBeChangedOrRemoved: <ACCOUNT_MESSAGE_BODY>]
  smtpServer:
    host: [ToBeChangedOrRemoved: <SMTP_SERVER_HOST>]
    port: [ToBeChangedOrRemoved: <SMTP_SERVER_PORT>]
    userName: [ToBeChangedOrRemoved: <SMTP_SERVER_USER_NAME>]
    password: [ToBeChangedOrRemoved: <SMTP_SERVER_PASSWORD>]
    timeout: [ToBeChangedOrRemoved: <SMTP_SERVER_TIMEOUT>]

# CometD Settings
cometDSettings:
  maxSessionsPerBrowser: [ToBeChangedOrRemoved: <MAX_SESSIONS_PER_BROWSER>]
  multiSessionInterval: [ToBeChangedOrRemoved: <MULTI_SESSION_INTERVAL>]

# OAuth2 Settings

# Session Persistence Settings

# Multimedia Disaster Recovery
drMonitoringDelay: [ToBeChangedOrRemoved: <DR_MONITORING_DELAY>]

# Stale CometD Session monitoring

# Node Settings Refresh

# Log Header Settings
logHeaderSettings:
  enableLogHeader: [ToBeChangedOrRemoved: "true"|"false"]
  updateOnPremiseInfoInterval: [ToBeChangedOrRemoved: <UPDATE_ON_PREMISE_INFO_INTERVAL>]

# Update on startup settings
updateOnStartup:
  opsCredentials: true
  features: true
  statistics: true
```

Make sure that you update all settings marked as `[ToBeChanged]`. You must also do the following:

- Set the applicationName to the name of the application you created in Configuring the Web Services applications — for example, `WS_Node`.
- In each Web Services cluster, you must configure one node as the synchronization node — `syncNode: true`. All other nodes in the cluster must have `syncNode: false`.

See serverSettings for details about supported configuration settings for this section.

On-premises settings

The settings in the **onPremiseSettings** section correspond to the contents of the **onpremise-settings.yaml** file in version 8.5.201.09 or earlier of Web Services and Applications. This section tells Web Services where Configuration Server is located.

For example:

```
onPremiseSettings:
  cmeHost: localhost
  cmePort: 8888
  countryCode: US
```

The **application.yaml.sample** file doesn't include a default **onPremiseSettings** section, so you'll need to add it yourself.

> **Warning**
>
> Ensure that you add the **onPremiseSettings** section to the top of the **application.yaml.sample** file. Web Services does not read the section if it is located elsewhere in the file.

See onPremiseSettings for details about all supported configuration settings for this section.

Tuning the Web Services host performance

Complete the steps below on each Web Services node to tune the performance of the host environment.

1. To optimize TCP/IP performance, you can run the following commands:

   ```
   sudo sysctl -w net.core.rmem_max=16777216
   sudo sysctl -w net.core.wmem_max=16777216
   sudo sysctl -w net.ipv4.tcp_rmem="4096 87380 16777216"
   sudo sysctl -w net.ipv4.tcp_wmem="4096 16384 16777216"
   sudo sysctl -w net.core.somaxconn=4096
   sudo sysctl -w net.core.netdev_max_backlog=16384
   sudo sysctl -w net.ipv4.tcp_max_syn_backlog=8192
   sudo sysctl -w net.ipv4.tcp_syncookies=1
   sudo sysctl -w net.ipv4.tcp_congestion_control=cubic
   ```

2. After providing for some means of starting Jetty, determine the user or group that will start Jetty and increase the file descriptors available to that user or group by adding the following to the **/etc/security/limits.conf** file:

   ```
   <user_name>                hard nofile        100000
   <user_name>                soft nofile        100000
   ```

   Where `<user_name>` is the name of the user or group that is starting Jetty.

Configuring Web Services as a System Service on Red Hat Linux Enterprise Linux 6

1. Open the **/etc/default/gws** file.
2. Update the following environment variables to values appropriate for your Web Services node:
    - **GWS_HOST**: Match this value to the Jetty host that you configured in the **jetty** section of the **application.yaml** configuration file.
    - **GWS_PORT**: Match this value to the Jetty port that you configured in the **jetty** section of the **application.yaml** configuration file.

SameSite cookies

To handle sameSite cookie attribute, you must configure options for both Jetty and CometD.

If the value of **SameSite** is set to `None`, Chrome browser also checks if the `Secure` cookie attribute is present, and if not, then warn user.

To mitigate this issue, make the following edits in `application.yaml`:

```
...
jetty:
  ...
  cookies:
    ...
    secure: true
    sameSite: None
...
serverSettings:
  ...
  cometDSettings:
    ...
    cookieSecure: true
    cookieSameSite: None
```

> **Important**
>
> If cookies are configured to be secure, the browser applies them to a secure connection only (https); therefore, these options take effect only if **enableSsl** is set to `true`.

If the value of **SameSite** is set to `Lax` or `Strict`, a secured connection is not required, for example:

```
...
jetty:
  ...
  cookies:
```

```
...
    ...
    httpOnly: true
    secure: false
    sameSite: Lax
...
serverSettings:
  ...
  cometDSettings:
    ...
    cookieHttpOnly: true
    cookieSecure: false
    cookieSameSite: Lax
```

However, it is important to note the following:

- If **SameSite** is set to Lax, the cookie is sent only on same-site requests or by top-level navigation with a safe HTTP method. That is, it will not be sent with cross-domain POST requests or when loading the site in a cross-origin frame, but it will be sent when the user navigates to the site via a standard top-level <a href=...> link.
- If **SameSite** is set to `Strict`, the cookie is never sent in cross-site requests. Even if the user clicks a top-level link on a third-party domain to your site, the browser refuses to send the cookie.

> **Important**
>
> You can choose an insecure connection by specifying a different type of SameSite (Lax or Strict), but this means that it will be impossible to embed Workspace Web Edition in an iframe or use it for any other cross-domain integrations. For example, applications like Genesys CRM Workspace/Adapter will not work with this configuration.

Next step

- Configuring features

Configuring Web Services for 8.5.201.09 or earlier

> **Important**
> Complete the steps on this page if you're installing Web Services and Applications version 8.5.201.09 or earlier; otherwise, go to Configuring Web Services.

To configure your Web Services nodes, you'll need to create three different configuration files: **server-settings.yaml**, **onpremise-settings.yaml**, and **cassandra-cluster.yaml**. You can find sample versions of these files in the main config folder you created in Step 6 of Deploying the web application for 8.5.201.09 or earlier; to use them, just remove the **.sample** extension.

Review the procedures on this page for details about the options you should configure in each of these files.

Modifying the server settings

Complete the steps below on each Web Services node.

Start

1. Open the **server-settings.yaml** file and review the options. This file contains a number of core parameters that are used by the server.

   The following is an unmodified file:

```
# URLs
externalApiUrlV2: [ToBeChanged: public URL including protocol, address and port,
<PUBLIC_SCHEMA_BASE_URL>]/api/v2
internalApiUrlV2: [ToBeChanged: internal URL including protocol, address and port,
<INTERNAL_SCHEMA_BASE_URL>]/internal-api
undocumentedExternalApiUrl: [ToBeChanged: public URL including protocol, address and
port, <PUBLIC_SCHEMA_BASE_URL>]/internal-api

# General
iwsDispositionCodeSync: [ToBeChanged: "true"|"false"]
temporaryAuthenticationTokenTTL: [ToBeChanged: "true"|"false"]
enableCsrfProtection: [ToBeChanged: "true"|"false"]
salesforceAuthenticationMode: [ToBeChanged: "true"|"false"]

# Timeouts
activationTimeout: 12000
configServerActivationTimeout: 35000
configServerConnectionTimeout: 15000
connectionTimeout: 4000
contactCenterSynchronizationTimeout: 60000
```

```
#inactiveUserTimeout: [ToBeChangedOrRemoved: <INACTIVE_USER_TIMEOUT>]
logoutAgentWhenNoActiveCometSessionTimeout: [ToBeChanged: "true"|"false"]
reconnectAttempts: 1
reconnectTimeout: 10000

# OPS account
opsUserName: [ToBeChanged: <OPS_USER_NAME>]
opsUserPassword: [ToBeChanged: <OPS_USER_PASSWORD>]

# Configuration Server credentials
applicationName: Cloud
applicationType: CFGGenericClient
cmeUserName: [ToBeChanged: <CONFIG_SERVER_USER_NAME>]
cmePassword: [ToBeChanged: <CONFIG_SERVER_USER_PASSWORD>]
syncNode: [ToBeChanged: "true"|"false"]
synchronizationCmeEventsPrefilterEnabled: [ToBeChanged: "true"|"false"]
enableVirtualQueueSynchronization: [ToBeChanged: "true"|"false"]

# Statistics
#statConnectionTimeout: [ToBeChangedOrRemoved: <STAT_CONNECTION_TIMEOUT>]
#statReconnectAttempts: [ToBeChangedOrRemoved: <STAT_RECONNECT_ATTEMPTS>]
#statReconnectTimeout: [ToBeChangedOrRemoved: <STAT_RECONNECT_TIMEOUT>]
#statOpenTimeout: [ToBeChangedOrRemoved: <STAT_OPEN_TIMEOUT>]
#statisticsWritesCL: [ToBeChangedOrRemoved: <STATISTICS_WRITE_SCL>]
#reportingSyncInterval: [ToBeChangedOrRemoved: <REPORTING_SYNC_INTERVAL>]
enableElasticSearchIndexing: [ToBeChanged: "true"|"false"]
#statisticsOpenRetryInterval: [ToBeChangedOrRemoved: <STATISTICS_OPEN_RETRY_INTERVAL>]

# Call Recording
createCallRecordingCF: [ToBeChanged: "true"|"false"]
#crClusterName: [ToBeChangedOrRemoved: <CR_CLUSTER_NAME>]
#crRegion: [ToBeChangedOrRemoved: <CR_REGION>]
#awsS3AccessKey: [ToBeChangedOrRemoved: <AWS_S3_ACCESS_KEY>]
#awsS3SecretKey: [ToBeChangedOrRemoved: <AWS_S3_SECRET_KEY>]
#awsS3BucketName: [ToBeChangedOrRemoved: <AWS_S3_BUCKET_NAME>]
#awsS3SocketTimeout: [ToBeChangedOrRemoved: <AWS_S3_SOCKET_TIMEOUT>]
#awsS3MaxErrorRetry: [ToBeChangedOrRemoved: <AWS_S3_MAX_ERROR_RETRY>]
#awsS3MaxConnection: [ToBeChangedOrRemoved: <AWS_S3_MAX_CONNECTION>]
#awsS3ConnectionTimeout: [ToBeChangedOrRemoved: <AWS_S3_CONNECTION_TIMEOUT>]
#cryptoSecurityKey: [ToBeChangedOrRemoved: <CRYPTO_SECURITY_KEY>]
#webDAVMaxConnection: [ToBeChangedOrRemoved: <WEBDAV_MAX_CONNECTION>]
#webDAVMaxTotalConnection: [ToBeChangedOrRemoved: <WEBDAV_MAX_TOTAL_CONNECTION>]

# CDR
#voiceMonitorNodeId: [ToBeChangedOrRemoved: <VOICE_MONITOR_NODE_ID>]

# Multi regional supporting
nodePath: [ToBeChanged: node position in cluster, example: /<REGION>/HOST
#nodeId: [ToBeChangedOrRemoved: unique value in cluster <NODE_ID>]

# SSL and CA
#caCertificate: [ToBeChangedOrRemoved: <CA_CERTIFICATE>]
#jksPassword: [ToBeChangedOrRemoved: <JKS_PASSWORD>]

# SAML
#samlSettings:
  #encryptionKeyName: [ToBeChangedOrRemoved: <SAML_ENCRYPTION_KEY_NAME>]
  #signingKeyName: [ToBeChangedOrRemoved: <SAML_SIGNING_KEY_NAME>]
  #identityProviderMetadata: [ToBeChangedOrRemoved: <SAML_IDENTITY_PROVIDER_METADATA>]
  #serviceProviderEntityId: [ToBeChangedOrRemoved: <SAML_SERVICE_PROVIDER_ENTITY_ID>]
  #encryptionKeyPassword: [ToBeChangedOrRemoved: <SAML_ENCRYPTION_KEY_PASSWORD>]
  #signingKeyPassword: [ToBeChangedOrRemoved: <SAML_SIGNING_KEY_PASSWORD>]
  #tlsKeyName: [ToBeChangedOrRemoved: <SAML_TLS_KEY_NAME>]
```

```
    #tlsKeyPassword: [ToBeChangedOrRemoved: <SAML_TLS_KEY_PASSWORD>]
    #responseSkewTime: [ToBeChangedOrRemoved: <SAML_RESPONSE_SWEW_TIME>]

  # CORS
  #crossOriginSettings:
    #allowedOrigins: [ToBeChangedOrRemoved: <CROSS_ALLOWED_ORIGINS>]
    #allowedMethods: [ToBeChangedOrRemoved: <CROSS_ALLOWED_METHODS>]
    #allowedHeaders: [ToBeChangedOrRemoved: <CROSS_ALLOWED_HEADERS>]
    #exposedHeaders: [ToBeChangedOrRemoved: <CROSS_EXPOSED_HEADERS>]
    #allowCredentials: [ToBeChangedOrRemoved: <CROSS_ALLOW_CREDENTIALS>]
    #corsFilterCacheTimeToLive: [ToBeChangedOrRemoved:
  <CROSS_ORIGIN_CORS_FILTER_CACHE_TIME_TO_LIVE>]

  # Elastic Search
  #elasticSearchSettings:
    #clientNode: [ToBeChangedOrRemoved: "true"|"false"]
    #indexPerContactCenter: [ToBeChangedOrRemoved: "true"|"false"]
    #enableScheduledIndexVerification: [ToBeChangedOrRemoved: "true"|"false"]
    #indexVerificationInterval: [ToBeChangedOrRemoved:
  <ELASTIC_SEARCH_INDEX_VERIFICATION_INTERVAL>]
    #retriesOnConflict: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_RETRIES_ON_CONFICT>]
    #waitToIndexTimeout: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_WAIT_TO_INDEX_TIMEOUT>]

  # Screen Recording
  #screenRecordingSettings:
    #screenRecordingEServicesEnabled: [ToBeChangedOrRemoved: "true"|"false"]
    #screenRecordingVoiceEnabled: [ToBeChangedOrRemoved: "true"|"false"]
    #clientSessionManagerCacheTTL: [ToBeChangedOrRemoved:
  <SCREEN_RECORDING_CLIENT_SESSION_MANAGER_CACHE_TTL>]
    #recordingInteractionEventsTTL: [ToBeChangedOrRemoved:
  <SCREEN_RECORDING_RECORDING_INTERACTION_EVENTS_TTL>]
    #contactCenterInfoManagerCacheTTL: [ToBeChangedOrRemoved:
  <SCREEN_RECORDING_CONTACT_CENTER_INFO_MANAGER_CACHE_TTL>]

  # Caching Settings
  #cachingSettings:
    #enableSystemWideCaching: [ToBeChangedOrRemoved: "true"|"false"]
    #agentStatesTTL: [ToBeChangedOrRemoved: <CACHING_AGENT_STATES_TTL>]
    #contactCenterFeaturesTTL: [ToBeChangedOrRemoved:
  <CACHING_CONTACT_CENTER_FEATURES_TTL>]
    #contactCenterSettingsTTL: [ToBeChangedOrRemoved:
  <CACHING_CONTACT_CENTER_SETTINGS_TTL>]
    #voiceContextCaching: [ToBeChangedOrRemoved: "true"|"false"]
    #voiceContextRefreshInterval: [ToBeChangedOrRemoved:
  <CACHING_VOICE_CONTEXT_REFRESH_INTERVAL>]

  # DoS Filter Settings
  enableDosFilter: [ToBeChanged: "true"|"false"]
  #dosFilterSettings:
    #maxRequestsPerSec: [ToBeChangedOrRemoved: <DOS_FILTER_MAX_REQUESTS_PER_SEC>]
    #delayMs: [ToBeChangedOrRemoved: <DOS_FILTER_DELAY_MS>]
    #maxWaitMs: [ToBeChangedOrRemoved: <DOS_FILTER_MAX_WAIT_MS>]
    #throttledRequests: [ToBeChangedOrRemoved: <DOS_FILTER_THROTTLED_REQUESTS>]
    #throttleMs: [ToBeChangedOrRemoved: <DOS_FILTER_THROTTLE_MS>]
    #maxRequestMs: [ToBeChangedOrRemoved: <DOS_FILTER_MAX_REQUEST_MS>]
    #maxIdleTrackerMs: [ToBeChangedOrRemoved: <DOS_FILTER_MAX_IDLE_TRACKER_MS>]
    #insertHeaders: [ToBeChangedOrRemoved: <DOS_FILTER_INSERT_HEADERS>]
    #trackSessions: [ToBeChangedOrRemoved: <DOS_FILTER_TTACK_SESSIONS>]
    #remotePort: [ToBeChangedOrRemoved: <DOS_FILTER_REMOTE_PORT>]
    #ipWhitelist: [ToBeChangedOrRemoved: <DOS_FILTER_IP_WHITE_LIST>]

  # Account Management
  #accountManagement:
    #forgotPasswordEmailTemplate:
      #from: [ToBeChangedOrRemoved: <PASSWORD_MESSAGE_FROM>]
      #subject: [ToBeChangedOrRemoved: <PASSWORD_MESSAGE_SUBJECT>]
      #body: [ToBeChangedOrRemoved: <PASSWORD_MESSAGE_BODY>]
    #accountCreatedEmailTemplate:
      #from: [ToBeChangedOrRemoved: <ACCOUNT_MESSAGE_FROM>]
      #subject: [ToBeChangedOrRemoved: <ACCOUNT_MESSAGE_SUBJECT>]
      #body: [ToBeChangedOrRemoved: <ACCOUNT_MESSAGE_BODY>]
    #smtpServer:
      #host: [ToBeChangedOrRemoved: <SMTP_SERVER_HOST>]
      #port: [ToBeChangedOrRemoved: <SMTP_SERVER_PORT>]
      #userName: [ToBeChangedOrRemoved: <SMTP_SERVER_USER_NAME>]
      #password: [ToBeChangedOrRemoved: <SMTP_SERVER_PASSWORD>]
      #timeout: [ToBeChangedOrRemoved: <SMTP_SERVER_TIMEOUT>]
```

2. Make sure to update all the options marked [ToBeChangedOrRemoved]. You must also set the applicationName to the name of the application you created in Configuring the Web Services applications — for example, WS_Node.
   You can review the Configuring features and Configuring security pages for more information about enabling specific functionality in your Web Services solution or for details about all the options available to you in the **server-settings.yaml** file, see Web Services configuration options options.

   > ### Important
   > In each Web Services cluster, one node must be configured as the synchronization node: syncNode = true. All other nodes in the cluster must have syncNode = false.

3. Save your changes and close the file.

End

Modifying the premise settings

Complete the steps below on each Web Services node.

Start

1. Open the **onpremise-settings.yaml** file and review the options. This file contains parameters that are used to connect to Configuration Server.

   The following is an unmodified file:

```
cmeHost: localhost
 cmePort: 8888
 countryCode: US
```

   The following options are valid in this file:

- **cmeHost** — The Configuration Server host name (FQDN) or IP.
- **cmePort** — The Configuration Server port.
- **backupCmeHost** — The backup Configuration Server host name (FQDN) or IP.

- **backupCmePort** — The backup Configuration Server port.
- **countryCode** — The premise contact center's country code.

> Important
>
> Configure the **backupCmeHost** and **backupCmePort** options if there is a backup Configuration Server in the Genesys environment and you want high-availability support.

- Save your changes and close the file.

End

Modifying the Cassandra cluster settings

Complete the steps below on each Web Services node.

Start

1. Open the **cassandra-cluster.yaml** file and review the options.

   The following is an unmodified file:

```
thrift_port: 9160
 jmx_port: 7199
 keyspace: sipfs
 nodes: [ToBeChanged: <CASSANDRA_PRIMARY_DC_NODES>]
 backup_nodes: [ToBeChanged: <CASSANDRA_BACKUP_DC_NODES>]
 replication_factor: [ToBeChanged: <REPLICATION_FACTOR>]
 write_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
 "CL_QUORUM" for single-DC env.]
 read_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
 "CL_QUORUM" for single-DC env.]
 max_conns_per_host: 16
 max_cons: 48
 max_pending_conns_per_host: 80
 max_blocked_threads_per_host: 160

 cassandraVersion: [ToBeChanged: "1.1" | "1.2"]
 useSSL: [ToBeChanged: supporting only for 1.2 Cassandra "false" | "true"]
```

2. Modify the settings as needed, making sure to update all the options marked `[ToBeChanged]`:
   - **thrift_port** — The port for Thrift to listen for clients. It should be the same as the `rpc_port` you set in the **cassandra.yaml** file as part of the Configuring Cassandra procedure.
   - **jmx_port** — The port Cassandra uses for Java Manage Extension (JMX).
   - **keyspace** — The name of the Cassandra keyspace. This name should be the same as the keyspace name you set while Creating the Cassandra keyspace. If you used the keyspace creation scripts that come with F, then you can leave this value as `sipfs`.

- **nodes** — A comma-separated list of Cassandra node IPs or host names.
- **backup_nodes** — A comma-separated list of backup Cassandra node IPs or host names. This option is intended for deployments that have two separate Cassandra data centers — Web Services switches from primary to backup if the primary nodes are unavailable. If your deployment is small with only one data center, you can ignore this option.
- **replication_factor** — A replication factor appropriate for your Cassandra topology. This value should be the same as the replication factor you set in Step 2 of the Creating the Cassandra keyspace procedure.
- **read_consistency_level** — Set this value according to your Cassandra topology:

Development (1 Cassandra node)	Single Data Center (1 data center with a minimum of three Cassandra nodes)	Two Data Centers (data centers with a minimum of three Cassandra nodes in each data center)
CL_ONE	CL_QUORUM	CL_LOCAL_QUORUM

- **write_consistency_level** — Set this value according to your Cassandra topology:

Development (1 Cassandra node)	Single Data Center (1 data center with a minimum of three Cassandra nodes)	Two Data Centers (data centers with a minimum of three Cassandra nodes in each data center)
CL_ONE	CL_QUORUM	CL_LOCAL_QUORUM

- **cassandraVersion** — Possible values are 1.1 (for 1.1.x versions) and 1.2 (for 1.2.x versions).
- **useSSL** — Set to `true` to enable Cassandra to use SSL. This option is only valid for Cassandra 1.2.x.

The following options tune the Cassandra database access. The default values were used by Genesys during internal load tests.

- **max_conns_per_host** — Maximum number of connections to allocate for a single host's pool.
- **max_cons** — Maximum number of connections in the pool.
- **max_pending_conns_per_host** — Maximum number of pending connect attempts per host.
- **max_blocked_threads_per_host** — Maximum number of blocked clients for a host.

3. Save your changes and close the file.

End

Tuning the Web Services host performance

Complete the steps below on each Web Services node to tune the performance of the host environment.

Start

1. To optimize TCP/IP performance, you can run the following commands:

```
sudo sysctl -w net.core.rmem_max=16777216
 sudo sysctl -w net.core.wmem_max=16777216
 sudo sysctl -w net.ipv4.tcp_rmem="4096 87380 16777216"
 sudo sysctl -w net.ipv4.tcp_wmem="4096 16384 16777216"
 sudo sysctl -w net.core.somaxconn=4096
 sudo sysctl -w net.core.netdev_max_backlog=16384
 sudo sysctl -w net.ipv4.tcp_max_syn_backlog=8192
 sudo sysctl -w net.ipv4.tcp_syncookies=1
 sudo sysctl -w net.ipv4.tcp_congestion_control=cubic
```

2. Increase the file descriptors by adding the following to the **/etc/security/limits.conf** file:

```
<user_name>                hard nofile        100000
 <user_name>                soft nofile        100000
```

- `<user_name>` — The name of the user or group that is starting Jetty.

End

Next step

- Configuring features

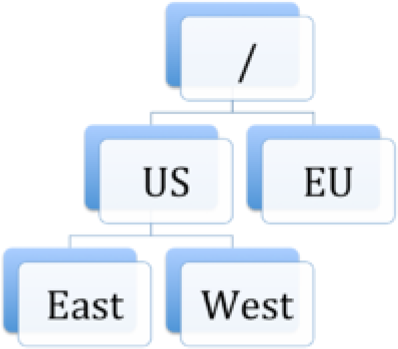
Multiple Data Center Deployment

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.

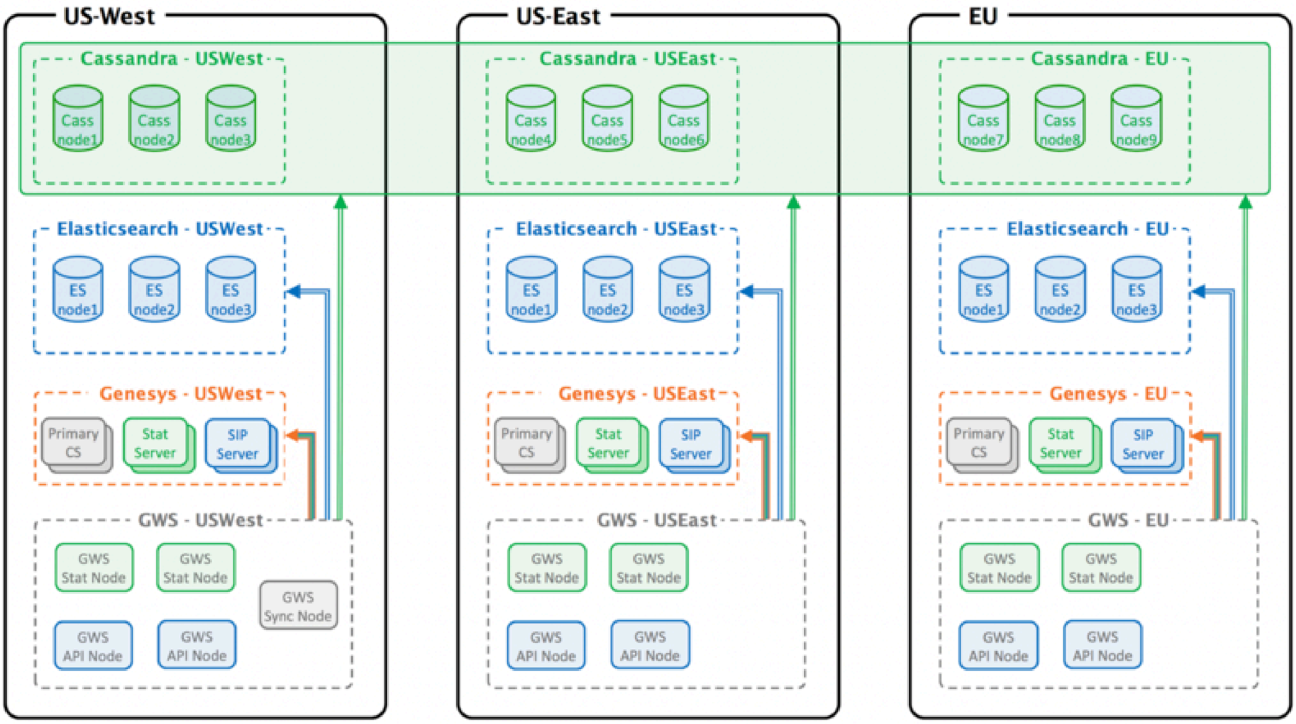
Architecture

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.

Configuring and enabling Web Services features

You can configure the following Web Services features:

- Reporting
- Elasticsearch
- Chat and email conference and transfer through a queue
- Contact availability
- Agent Group availability (for Voice only)

Some features are enabled by default. Other features, such as eService features, are disabled and you can enable the features that are applicable for your deployment. For more information, see Enable features in the Feature Definition file.

Next step

- Configuring security

Reporting

You can configure Web Services to use, store, and expose statistical data for agents, skills, and queues. This data is used for reporting in Workspace Web Edition and you can also access it through the API. Complete the following steps to set up reporting in Web Services:

1. Configure Elasticsearch to support real-time statistics.
2. Enable reporting.

Enabling reporting

Start

1. Open the **application.yaml** file (**server-settings.yaml** if you're installing Web Services and Applications version 8.5.201.09 or earlier).
2. Configure the Web Services node by setting the nodeId option to a unique identifier, such as the host name or IP address of the node.
3. Review the statistics configuration options for details about setting the connection to Stat Server. Adjust the default settings for these options as required for your deployment.
4. Save your changes and close the file.
5. Confirm that the **statistics.yaml** file is in present your Web Services home folder (the $JETTY_HOME/genconfig folder if you're installing Web Services and Applications version 8.5.201.09 or earlier). This file defines which statistics Web Services requests from Stat Server.

End

Next step

- Back to Configuring features

Elasticsearch

Web Services uses Elasticsearch — an open-source, full-text search engine with a RESTful web interface — to store both real-time and historical statistics. Real-time statistics reflect the current state of the object (User, Queue, Skill), while historical statistics are stored as time-based events.

For new deployments, Genesys recommends that you set up a cluster of Elasticsearch nodes that is separate from your Web Services nodes. See Configuring Web Services to use a standalone Elasticsearch cluster for details. It's possible to set up an embedded Elasticsearch cluster, which means that Elasticsearch is included in your Web Services nodes. **Note: Genesys does not recommend using this approach in a production environment.** See Configuring Web Services to use an embedded Elasticsearch cluster for details.

Configuring Web Services to use a standalone Elasticsearch cluster

You can configure Web Services to work with a standalone Elasticsearch cluster by completing the steps below.

> ### Important
> Genesys recommends that you use a standalone Elasticsearch cluster for new deployments of Web Services. Contact your Genesys representative for information about how to migrate from embedded to standalone.

Prerequisites

- You have deployed and configured a cluster of Elasticsearch nodes. Refer to the Elasticsearch documentation for details. **Note:** Genesys recommends that you use the latest stable 1.x version of Elasticsearch.

Start
Complete the following steps for each Web Services node:

1. Copy the **installation_path/config-templates/elasticsearch.yml.sample** file to the configuration folder on each node in your Elasticsearch cluster, rename the file to **elasticsearch.yml**, and then edit the file to set correct option values.
2. Copy all files from the **installation_path/elasticsearch/templates/** folder to the templates directory of the configuration folder, for example, **/etc/elasticsearch/templates**, on each node in your Elasticsearch cluster.
3. Web Services keeps Elasticsearch in sync with Cassandra and removes statistical data for deleted objects by performing index verification. By default, Web Services runs index verification when its node is started, but you can also configure scheduled index verification by setting

enableScheduledIndexVerification to `true` in the elasticSearchSettings option. By default, the index verification takes place every 720 minutes (12 hours), but you can change this timing by setting **indexVerificationInterval** in elasticSearchSettings. **Note:** Genesys recommends that you only configure one Web Services node for index verification to avoid excessive requests to Elasticsearch and Cassandra.

4. If your deployment uses statistics, make sure you complete the reporting configuration steps, including setting the nodeId.
5. Set the crClusterName option to the name of the cluster. All Web Services nodes with the same cluster name will form the cluster.
6. Set the enableElasticSearchIndexing option to `true`. **Note:** In this case, Web Services writes statistics values to both Elasticsearch and Cassandra, but only reads them from Elasticsearch.
7. Set the elasticSearchSettings option to appropriate values for your environment.

End

Configuring Web Services to use an embedded Elasticsearch cluster

You can configure Web Services to work with an embedded Elasticsearch cluster by completing the steps below.

> ### Important
> Genesys does not recommend using an embedded Elasticsearch cluster in a production environment. Contact your Genesys representative for information about how to migrate from embedded to standalone.

Complete the following steps for each Web Services node that you want to host Elasticsearch:

Start

1. Copy **installation_path/config-templates/elasticsearch.yml.sample** to the **installation_path/config/elasticsearch.yml** file, and then edit the file to set correct option values.
2. Web Services keeps Elasticsearch in sync with Cassandra and removes statistical data for deleted objects by performing index verification. By default, Web Services runs index verification when its node is started, but you can also configure scheduled index verification by setting **enableScheduledIndexVerification** to `true` in the elasticSearchSettings option. By default, the index verification takes place every 720 minutes (12 hours), but you can change this timing by setting **indexVerificationInterval** in elasticSearchSettings. **Note:** Genesys recommends that you only configure one Web Services node for index verification to avoid excessive requests to Elasticsearch and Cassandra.
3. If your deployment uses statistics, make sure you complete the reporting configuration steps, including setting the nodeId.
4. Set the crClusterName option to the name of the cluster. All Web Services nodes with the same cluster

name will form the cluster.

5. Set the enableElasticSearchIndexing option to `true`. **Note:** In this case, Web Services writes statistics values to both Elasticsearch and Cassandra, but only reads them from Elasticsearch.
6. Set the elasticSearchSettings option to appropriate values for your environment.

End

Next Step

- Back to Configuring features

Consultation, conference, and transfer through a queue for chat

To ensure consultations, conferences, and transfers through queues are reported correctly for Chat, you must update the **Interaction Subtype** Business Attribute.

Genesys Administrator

Updating the **Interaction Subtype** Business Attribute

Start

1. Navigate to **PROVISIONING > Routing/eServices > Business Attributes**, select **Interaction Subtype**, and click **Edit...**.
2. Select the **Attributes Values** tab and click **New...**.
3. Enter the following:
    - Name: `InternalConferenceInvite`
    - Display Name: `Internal Conference Invite`
4. Click **Save & Close**.

End

Configuration Manager

Updating the **Interaction Subtype** Business Attribute

Start

1. Navigate to **Business Attributes > Interaction Subtype > Attribute Values**.
2. Right-click and select **New > Business Attribute Value**.
3. In the **General** tab, enter the following:
    - Name: `InternalConferenceInvite`
    - Display Name: `Internal Conference Invite`
4. Click **OK**.

End

Next step

- Back to Configuring features

Contact availability

Your Web Services and Applications solution must meet the following requirements to enable contact availability for contact resources of type User in the Contacts API:

- Your environment must include a connection to Stat Server.
- You have enabled statistics monitoring.
- Your **statistics.yaml** file contains the following definitions:

```
---
#internal stats
name: CurrentTargetState
statisticDefinitionEx:
    category: CurrentTargetState
    mainMask: "*"
    subject: DNStatus
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
objectType: AGENT
notificationMode: IMMEDIATE
notificationFrequency: 0
---
name: CurrentAgentState
notificationFrequency: 0
notificationMode: IMMEDIATE
objectType: AGENT
statisticDefinitionEx:
    category: CurrentState
    mainMask: "*"
    subject: DNAction
```

- You have enabled multimedia channel states monitoring (optional).
- The contact must have a device assigned and be logged in; otherwise, Web Services does not include the availability subresource.

Agent Group Availability (for Voice)

Your Web Services and Applications solution must meet the following requirements to enable Agents to view Agent Group availability for Voice in Team Communicator:

- Your environment must include a connection to Stat Server.
- You have enabled statistics reporting.
- Your **statistics.yaml** file contains the following definitions:

```
---
name: TransferAvailability_CurrentReadyAgents
notificationFrequency: 10
notificationMode: IMMEDIATE
objectType: VIRTUAL_AGENT_GROUP
statisticDefinitionEx:
     dynamicFilter: "MediaType=voice"
     category: CurrentNumber
     mainMask: WaitForNextCall
     subject: DNStatus
---
name: TransferAvailability_CurrentReadyAgents
notificationFrequency: 10
notificationMode: IMMEDIATE
objectType: AGENT_GROUP
statisticDefinitionEx:
     dynamicFilter: "MediaType=voice"
     category: CurrentNumber
     mainMask: WaitForNextCall
     subject: DNStatus
---
```

Enabling features in the Feature Definitions file

Some features are enabled by default. Other features, such as eService features, are disabled and you can enable the features that are applicable for your deployment.

The Feature Definitions file contains a list of features that are available for your contact center. By default, all data access APIs and all voice-related functionality is enabled. Therefore, for most voice-only deployments, you do not need to make any changes to the Feature Definitions file.

Procedure

1. Locate the **feature-definitions.json** file, which is located in the **config-templates** folder.
2. Move the file to the **GWS_CONF folder** and then open the file.
3. For each feature that you want to enable, set the **autoAssignOnContactCenterCreate** flag to true.
4. Save the file.
   In the following example, the Facebook API is enabled and GWS will attempt to connect to the interaction server

   ```
   { "id": "api-multimedia-facebook", "displayName": "Multimedia Facebook API",
    "description": "API for Multimedia Facebook", "autoAssignOnContactCenterCreate": true }
   ```

Sample Feature Definitions file

```
[
  {
    "id": "api-provisioning-read",
    "displayName": "API Provisioning Read",
    "description": "General provisioning read",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-provisioning-write",
    "displayName": "API Provisioning Write",
    "description": "General provisioning write",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-voice",
    "displayName": "Voice API",
    "description": "API for Voice",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-voice-predictive-calls",
    "displayName": "Voice API - Predictive calls",
```

```
    "description": "Enables predictive calls for a contact center",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-voice-outbound",
    "displayName": "Voice API Outbound",
    "description": "API for Outbound",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-supervisor-agent-control",
    "displayName": "API Supervisor Agent Control",
    "description": "API for Supervisors to Control Agent State",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-supervisor-monitoring",
    "displayName": "API Supervisor Monitoring",
    "description": "API for Supervisors to Monitor Agents",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-multimedia-chat",
    "displayName": "Multimedia Chat API",
    "description": "API for Multimedia Chat",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-email",
    "displayName": "Multimedia Email API",
    "description": "API for Multimedia Email",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-facebook",
    "displayName": "Multimedia Facebook API",
    "description": "API for Multimedia Facebook",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-twitter",
    "displayName": "Multimedia Twitter API",
    "description": "API for Multimedia Twitter",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-workitem",
    "displayName": "Multimedia Workitem API",
    "description": "API for Multimedia Workitem",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-user-account-management-email",
    "displayName": "User Account Management via Email",
    "description": "API for account management via email",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-devices-webrtc",
    "displayName": "WebRTC Support",
    "description": "API for WebRTC provisioning",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-ucs-voice",
    "displayName": "Support UCS for voice",
    "description": "For support contact center in voice",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-voice-instant-messaging",
    "displayName": "API Voice Instant Messaging",
    "description": "API for Internal Agent-to-Agent Chat",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-platform-configuration-read",
    "displayName": "Platform Configuration API - read",
    "description": "Low-level configuration API",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-platform-configuration-write",
    "displayName": "Platform Configuration API - write",
    "description": "Low-level configuration API",
    "autoAssignOnContactCenterCreate": true
  }
]
```

Configuring security

Web Services adheres to the standards described in the Open Web Application Security Project (OWASP) Top 10 — see the OWASP website for details about the Top 10 — and has adopted several methods of ensuring security, for example:

- Errors are logged locally to prevent information leakage through API requests.
- User sessions have a timeout option.
- Cross Site Request Forgery Protection

Web Services includes additional security configurations that you can use with your installation:

- Transport Layer Security (TLS)
- Security Assertion Markup Language (SAML) authentication
- Cross-Site Request Forgery (CSRF) protection
- Cross-Origin Resource Sharing (CORS) filter

For details about how Web Services handles authentication, see Web Services authentication flow.

Next step

- Starting and testing Web Services and Applications

Transport Layer Security (TLS)

Configuring TLS between Web Services and Configuration Server

Web Services can use a secured Transport Layer Security (TLS) connection mechanism to connect to Configuration Server. When configured, Web Services connects to a secure port on Configuration Server, verifies the server's authority, and encrypts/decrypts network traffic. You can configure secured connections to Configuration Server in the following ways:

- Minimal configuration
- Validate the certificate against the CA

Prerequisites

Before configuring Web Services, make sure the Configuration Server secure port is configured as described in Introduction to Genesys Transport Layer Security in the *Genesys Security Deployment Guide* and that all certificates for server host and the certificate authority are configured and available.

Minimal configuration

Web Services does not check the server's certificate against the Certificate Authority, but all traffic is encrypted. To configure Web Services with minimal configuration, all you need to do is configure a connection to a secured port on Configuration Server. You can do this using **either** of the following methods:

- For the initial connection to Configuration Server, set the tlsEnabled option to `true` in the onPremiseSettings section of the **application.yaml** file (**onpremise-settings.yaml** file if you are installing Web Services and Applications version 8.5.201.09 or earlier). This creates a secured connection to Configuration Server the first time Web Services starts.
- For an environment that is already configured with Configuration Manager synchronization enabled, you can make changes with Configuration Manager as described in the Genesys Security Deployment Guide. These changes are synchronized back to the Cassandra database from Configuration Manager.

Validate the certificate against the CA

In order to support the client-side certificate check, Web Services needs the public key for the Certificate Authority (CA). Web Services supports the PEM and JKS key storage formats, but recommends using JKS because it's compatible with both Cassandra and HTTPS.

Complete the steps below to validate the certificate against the CA.

> **Important**
>
> The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

Start

1. If you plan to use a JKS file, you can generate if from a PEM file by importing the PEM certificate, as shown here:

   ```
   keytool -importcert -file ca_cert.pem -keystore ca_cert.jks
   ```

2. Once you have the **ca_cert.jks** file, place it in a location available from your Web Services host, such as:
   - A local folder on the Web Services host
   - A network share
3. Configure the following options in the serverSettings section of the **application.yaml** file (**server-settings.yaml** if you're installing Web Services and Applications version 8.5.201.09 or earlier):
   - For a PEM file, set caCertificate to the location of the file. For example:

     ```
     caCertificate: /opt/ca_cert.pem
     ```

   - For a JKS file, set caCertificate to the location of the file and set jksPassword to the password for the key storage. For example:

     ```
     caCertificate: /opt/ca_cert.jks
      jksPassword: pa$$word
     ```

End

Configuring TLS for connections with Cassandra

Genesys supports Transport Layer Security (TLS) for connections from Web Services to Cassandra and between Cassandra nodes. You can configure secured connections for one or both of the following scenarios:

- Secure connections from Web Services to Cassandra
- Secure connections between Cassandra nodes

Secure Connections from Web Services to Cassandra

Prerequisites

- You have installed Bash, Java keytool and OpenSSL

Complete the following steps to configure TLS for connections from Web Services to Cassandra.

> **Important**
>
> The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

Start of procedure

1. Do one of the following:
   - Create the server-side keystore with a self-signed certificate and the client-side truststore — which contains the public part of server certificate. Run the following commands:

     ```
     #!/bin/bash
     #generate keypair
     keytool -genkeypair -alias cassandra -keyalg RSA -keysize 1024 -dname "CN=cassandra,
     OU=Test, O=Test Ltd, C=US" -keystore server.jks
     -storepass password -keypass password

     #export certificate
     keytool -exportcert -alias cassandra -file client.pem -keystore server.jks -storepass
     password -rfc

     #create client truststore and import certificate
     keytool -importcert -alias cassandra -file client.pem -keystore client.jks -storepass
     password -noprompt
     ```

   - Create a self-signed root authority, use it to sign the server certificate, store it to **server.jks** and create the client-side truststore, which trusts all certificates signed with root authority. Run the following commands:

     ```
     #!/bin/sh

     #generate self-signed root certificate
     keytool -genkeypair -alias root -keyalg RSA -keysize 1024 -validity 3650 -dname
     "CN=TestRoot, OU=Dev, O=Company, C=US" -keystore root.jks
     -storepass password -keypass password

     #export root certificate
     keytool -exportcert -alias root -file root.crt -keystore root.jks -storepass password

     #generate server-side certificate
     keytool -genkeypair -alias server -keyalg RSA -keysize 1024 -validity 3650 -dname
     "CN=TestServer, OU=Dev, O=Company, C=US"
     -keystore server.jks -storepass password -keypass password

     #create the sign request for server certificate
     keytool -certreq -alias server -keystore server.jks -file server.csr -storepass
     password -keypass password

     #export private key of root auth: need later for signing the server certificate
     keytool -v -importkeystore -srckeystore root.jks -srcalias root -destkeystore
     root.p12 -deststoretype PKCS12 -noprompt
     -destkeypass password -srckeypass password -destalias root -srcstorepass password
     -deststorepass password
     ```

```
openssl pkcs12 -in root.p12 -out private.pem -password pass:password -passin
pass:password -passout pass:password
rm root.p12

#sign the certificate
openssl  x509  -req  -CA   private.pem  -in server.csr -out server.crt  -days 3650
-CAcreateserial -passin pass:password
rm private.pem
rm private.srl
rm server.csr

#import root certificate to client side trust store
keytool -importcert -alias root -file root.crt -keystore client.jks -storepass
password -noprompt

#import root certificate to server side key store
keytool -importcert -alias root -file root.crt -keystore server.jks -storepass
password -noprompt
rm root.crt

#import certificate sign reply into server-side keystore
keytool -import -trustcacerts -alias server -file server.crt -keystore server.jks
-storepass password -keypass password
rm server.crt
```

2. Configure Cassandra to use your generated certificates for the client connection by setting the `client_encryption_options` in the **cassandra.yaml** file. For example:

```
client_encryption_options:
    enabled: true
    keystore: <absolute path to server.jks file>
    keystore_password: password
    #the password specified in while creating storage
    # For the purpose of the demo the default settings were used.
    # More advanced defaults below:
    #protocol: TLS
    #algorithm: SunX509
    #store_type: JKS
    #cipher_suites: [TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA]
```

> ### Important
> To enable support for encryption, you must have the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction installed.

3. Confirm the Cassandra nodes can start successfully:
   a. Edit the **conf/log4j-server.properties** file and uncomment the following line:

      ```
      log4j.logger.org.apache.cassandra=DEBUG
      ```

   b. Start Cassandra and check the logs. If the configuration was successful, you shouldn't see any errors.
4. Check that SSL-to-client is working successfully using cassandra-cli:
   a. Confirm that unsecured connections aren't possible by starting cassandra-cli locally — this forces it to connect to the Cassandra instance running on localhost. You should expect to see the exception in the cassandra-cli output.

b. Confirm that secured connections are possible by running the following command from the directory where cassandra-cli is installed:

```
./cassandra-cli -tf org.apache.cassandra.cli.transport.SSLTransportFactory -ts
<absolute path to client truststore cass_client.jks> -tspw somePassword
```

c. If the configuration is successful, you should see the "Connected to" welcome message:

```
Connected to: "Test Cluster" on 127.0.0.1/9160
Welcome to Cassandra CLI version 1.2.12
```

5. Configure Web Services to enable secure connections:

a. On the Web Services node, open the **application.yaml** file.

b. In the `cassandraCluster` section, configure the following settings:

Parameter	Type	Default	Description
useSSL	Boolean	`false`	Specify `true` to connect Cassandra to an SSL channel. This paramater only applies to Cassandra 1.2.x.
truststore	String		Specify the absolute path to the trustore file. Ensure that Web Services can read the file. The supported format is **JKS**.
truststorePassword	String		Specify the Truststore password.
sslProtocol	String	TLS	Specify the SSL protocol to be used. This protocol is passed to JAVA security.
cipherSuites	String	`[TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA]`	Specify a list of ciphers in the form of a yaml list. The list must match the list that is configured in Cassandra.

c. Specify your Cassandra version using the cassandraVersion parameter.

> Important
>
> Web Services does not support Cassandra 1.1 for on premise deployments.

End or procedure

Secure connections between Cassandra nodes

When you enable SSL for connections between Cassandra nodes, you ensure that communication between nodes in the Cassandra cluster is encrypted, and that only other authorized Cassandra nodes can join the cluster.

The steps below show you how to create a single certificate to be used by all Cassandra nodes in the cluster. This simplifies cluster management because you don't need to generate a new certificate each time you add a new node to the cluster, which means you don't need to restart all nodes to load the new certificate.

> **Important**
>
> The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

Start

1. Generate a keystore and truststore. See Step 1 of Secure connections from Web Services to Cassandra for details.
2. On each Cassandra node in the cluster, set `server_encryption_options` in the **cassandra.yaml** file. For example:

```
server_encryption_options:
    internode_encryption: all
    keystore: <absolute path to keystore >
    keystore_password: <keystore password - somePassword in our sample>
    truststore: <absolute path to trustore>
    truststore_password: <truststore password - somePassword in our sample>
```

3. Check the Cassandra logs. If the configuration was successful, you shouldn't see any errors.

End

Next Step

- Back to Configuring security

SAML authentication

Web Services supports Security Assertion Markup Language (SAML) for single sign-on (SSO) authentication.

Configuring SAML

To enable SAML, make the following configuration changes in the serverSettings section of the **application.yaml** file on each of your Web Services nodes (**server-settings.yaml** if you're installing Web Services and Applications version 8.5.201.09 or earlier):

Start

1. Set the following options in the SSL and CA section:
    - caCertificate — should point to a JKS key storage that includes the SAML encryption key. See Generating security keys for details.
    - jksPassword — should be the password for the **caCertificate** key storage.
2. Set the following option in the SAML section:
    - samlSettings — the following properties are mandatory:
        - encryptionKeyName
        - signingKeyName
        - identityProviderMetadata
3. Save the changes to the file. Your configuration should look something like this:

```
# SSL and CA
caCertificate: /Users/samluser/Documents/Keys/keystore.jks
jksPassword: password

# SAML
samlSettings:
    serviceProviderEntityId: genesys.staging.GWS
    encryptionKeyName: client
    signingKeyName: client
    identityProviderMetadata: /Users/samluser/Documents/Metadata/idp-metadata.xml
```

4. To activate SAML authentication, append the browser URL for Workspace Web Edition with `?authType=saml`.
5. To enable extended SAML logging, add the following string to **logback.xml** file: `<logger name="org.springframework.security.saml" level="%LEVEL%"/>`, where valid values for LEVEL are INFO (preferred) or DEBUG.

End

Generating security keys

You can use the keytool utility that comes with the Java SDK to generate a JKS key store. Use the following command:

```
keytool -genkey -keystore <path_to_jks_file> -alias <key_name> -keypass <key_password>
-storepass <store_password> -dname <distinguished_name>
```

If you already have a JKS key store, you can add a key to it by executing the command above with the same file name and the new key name and key password. For example:

```
keytool -genkey -keystore /opt/keystore.jks -alias encryption_key -keypass genesys -storepass
genesys -dname "CN=GWS, OU=R&D, O=Genesys, L=Daly City, S=California, C=US"
```

Next step

- Back to Configuring security

Cassandra authentication

Web Services supports Cassandra authentication. Authentication validates incoming user connections to the Cassandra database. Implementing Cassandra authentication requires you to do some configuration in Cassandra and in Web Services.

Configure Cassandra authentication

The user login accounts and their passwords required for authentication are managed inside the **cassandra.yaml** file. Configure Cassandra authentication according to Datastax documentation.

Web Services configuration

To support Cassandra authentication, open the **application.yaml** file and provide the appropriate credentials. For example:

```
cassandraCluster:
  thrift_port: 9160
  jmx_port: 7199
  ...
  userName: <super user name>
  password: <super user password>
  ...
```

Important

- To save backward compatable behavior when the username or password is not provided, GWS will try to connect to Cassandra in anonymous way.
- You might also want to encrypt the password for added security.

CSRF protection

Web Services provides protection against Cross Site Request Forgery (CSRF) attacks. For general information and background on CSRF, see the OWASP CSRF Prevention Cheat Sheet.

To set up Cross Site Request Forgery protection, set the following options in the serverSettings section of the **application.yaml** file on each of your Web Services nodes (**server-settings.yaml** if you're installing Web Services and Applications version 8.5.201.09 or earlier) :

- enableCsrfProtection — determines whether CSRF protection is enabled on the Web Services node.
- crossOriginSettings — specifies the configuration for cross-origin resource sharing in Web Services. Make sure this option has the *exposedHeaders* setting with a value that includes `X-CSRF-HEADER,X-CSRF-TOKEN`.

For example, your configuration might look like this:

```
enableCsrfProtection: true
crossOriginSettings:
    corsFilterCacheTimeToLive: 120
    allowedOrigins: http://*.genesys.com, http://*.genesyslab.com
    allowedMethods: GET,POST,PUT,DELETE,OPTIONS
    allowedHeaders: "X-Requested-With,Content-Type,Accept,
Origin,Cookie,authorization,ssid,surl,ContactCenterId"
    allowCredentials: true
    exposedHeaders: "X-CSRF-HEADER,X-CSRF-TOKEN"
```

For more information about CSRF protection in the Web Services API, see Cross Site Request Forgery Protection.

Next step

- Back to Configuring security

CORS filter

Web Services supports Cross-Origin Resource Sharing (CORS) filter, which allows applications to request resources from another domain. For general information and background on CORS, see Cross-Origin Resource Sharing.

> ### Important
> CORS must be enabled for the screen recording options to be available in the Speechminer Web UI when the using Microsoft Internet Explorer web browser.

To set up Cross-Origin Resource Sharing, make sure you set the crossOriginSettings option in the serverSettings section of the **application.yaml** file on each of your Web Services nodes (**server-settings.yaml** if you're installing Web Services and Applications version 8.5.201.09 or earlier). It specifies the configuration for cross-origin resource sharing in Web Services. Make sure this option has the **exposedHeaders** setting with a value that includes `X-CSRF-HEADER,X-CSRF-TOKEN`.

For example, your configuration might look like this:

```
crossOriginSettings:
    corsFilterCacheTimeToLive: 120
    allowedOrigins: http://*.genesys.com, http://*.genesyslab.com
    allowedMethods: GET,POST,PUT,DELETE,OPTIONS
    allowedHeaders: "X-Requested-With,Content-
Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId,X-CSRF-TOKEN"
    allowCredentials: true
    exposedHeaders: "X-CSRF-HEADER,X-CSRF-TOKEN"
```

For more information about CORS in the Web Services API, see Cross-Origin Resource Sharing.

Next step

- Back to Configuring security

Web Services authentication flow

Web Services provides authentication in the following sequence:

1. Salesforce Authentication

- Enters here if a request contains two specific headers (Salesforce Session ID and Salesforce Identity URL).
- If successful, the user is authenticated and execution flow proceeds to the authorization stage.
- If authentication headers are not present or authentication fails, execution flow proceeds to the next step.

2. Configuration Server Authentication

- Enters here if a request contains basic authentication header and Configuration Server authentication is enabled for this contact center.
- If successful, user is authenticated and execution flow proceeds to the authorization stage.
- If authentication headers are not present, Configuration Server authentication is disabled, or authentication fails, execution flow proceeds to the next step.

3. Web Services Authentication

- Enters here if a request contains basic authentication header.
- If successful, user is authenticated and execution flow proceeds to the authorization stage.
- If authentication headers are not present or authentication fails, execution flow proceeds to the next step.

4. Security Assertion Markup Language (SAML) Authentication

- Enters here if SAML is enabled and configured.
- An attempt is made to authenticate user through the standard SAML authentication flows.
- If successful, the user is authenticated and execution flow proceeds to the authorization stage.
- If not successful, the user receives an anonymous authentication, which means this users is only given access to unprotected endpoints.

Next step

- Back to Configuring security

Password encryption

For added security, consider encrypting your passwords in the **application.yaml** file. This feature is only supported for JAR (Spring Boot) distributables.

The following table identifies which passwords can be encrypted and where you can find them in the **application.yaml** file:

File section	Settings
jetty > ssl	• keyStorePassword • keyManagerPassword • trustStorePassword
serverSettings	• opsUserPassword • cmePassword • jksPassword • webDAVPassword
serverSettings > samlSettings	• encryptionKeyPassword • signingKeyPassword • tlsKeyPassword
serverSettings > accountManagement > smtpServer	• password
cassandraCluster	• password • truststorePassword

Procedure: Encrypting passwords

Start

1. Run the GWS application with the **--encrypt** parameter followed by the password you need to encrypt. For example:

```
 $ java -jar gws.jar --encrypt ops
CRYPT:an03xPrxLAu9p==
```

The GWS application only encrypts and prints the password. The server won't actually start.

2. Copy the printed encrypted password and paste into the **application.yaml** file. For example:

```
opsUserName: ops
 opsUserPassword: CRYPT:an03xPrxLAu9p==
```

The server only decrypts passwords that start with the **CRYPT:** prefix. Passwords without the **CRYPT:** prefix are considered plain text and remain unmodified.

End

Secure Cookies

Web Services uses the **secure** flag option when sending a new cookie to the user within an HTTP Response. The purpose of the **secure** flag is to prevent cookies from being observed by unauthorized parties due to the transmission of a the cookie in clear text.

Enabling the **secure** flag

Set the **cookies** option in the **jetty** section of the **application.yaml** file on your Web Services nodes. For details, see Configuring Web Services.

```
cookies:
   httpOnly: true
   secure: true
```

Sample Cookie Header when **secure** flag is not set

```
Set-Cookie: MyCookieName=The value of my cookie; path=/; HttpOnly
```

Sample Cookie Header when **secure** flag is set

```
Set-Cookie: MyCookieName=The value of my cookie; path=/; HttpOnly; secure
```

When the cookie is declared as secure in the cookies configuration option, the browser will prevent the transmission of a cookie over an unencrypted channel.

Starting and testing

After you install and configure Web Services, you should start the nodes in the following order:

1. Start the syncNode.
2. Start the remaining nodes.

Starting Web Services nodes

Perform the following for each Web Services node, starting with the syncNode.

- To start the node, enter the following command as one complete command line entry, replacing items in square brackets [ ] with the appropriate values as described below:

  ```
  [java dir]/java -Xmx8G -Xms8G -jar [home dir]/gws.jar
  ```

  - `[java dir]` — The home directory for Java. For example, **/usr/bin/java**.
  - `[home dir]` — The home directory you created for your Web Services web application. For example, **/web-services**.

> Important
>
> **The memory allocation -`Xmx8G` and -`Xms8G` are mandatory for the Web Services node.**

For example, your command line might look like the following:

```
/usr/bin/java -Xmx8G -Xms8G -jar /web-services/gws.jar
```

Testing Web Services

Complete the steps below to verify each Web Services node is up and running.

1. Type the following URL into a web browser:
   `http://`*`ws_host`*`:`*`ws_port`*`/api/v2/diagnostics/version`

   - *`ws_host`* — The host name or IP address for the Web Services node.
   - *`ws_port`* — The port for the Web Services node.

   For example, the URL might be `http://192.0.2.20:8080/api/v2/diagnostics/version`

   If the request is successful, the version is printed in the browser:

```
{"statusCode":0,"version":"8.5.200.96"}
```

Testing Workspace Web Edition

Complete the following steps for each Web Services node to confirm that Workspace Web Edition is working.

1. Launch Workspace Web Edition by navigating to `http://`*`ws_host`*`:`*`ws_port`*`/ui/ad/v1/index.html` in a web browser.
   - *`ws_host`* — The host name or IP address for the Web Services node.
   - *`ws_port`* — The port for the Web Services node.

   For example, the URL might be `http://192.0.2.20:8080/ui/ad/v1/index.html`
2. Enter the credentials for a WWE agent, see Setting Up Agents On The System. Note that the user must be of the agent type.
3. After clicking **Log In**, Workspace Web Edition displays the agent desktop view and the agent is ready to work.

Testing Gplus Adapter for Salesforce

Complete the following steps for each Web Services node to confirm that Gplus Adapter for Salesforce is working.

1. Launch Gplus Adapter for Salesforce by navigating to `http://`*`ws_host`*`:`*`ws_port`*`/ui/crm-adapter/index.html` in a web browser.
   - *`ws_host`* — The host name or IP address for the Web Services node.
   - *`ws_port`* — The port for the Web Services node.

   For example, the URL might be `http://192.0.2.20:8080/ui/crm-adapter/index.html`
2. Enter the credentials for any of the previously created agents. Note that the user must be of the agent type.
3. After clicking **Log In**, Gplus Adapter for Salesforce displays in the web browser. For details about how to deploy the adapter in Salesforce, see Gplus Adapter for Salesforce in this guide.

Web Services configuration options

You can set the configuration options below in the corresponding sections of the **application.yaml** file on your Web Services nodes (**server-settings.yaml** if you are installing Web Services and Applications version 8.5.201.09 or earlier). For details, see Configuring Web Services.

logging

Settings in this section are listed under "logging".

config

Default Value: `logback.xml`
Valid Values: A valid path
Mandatory: No
Specifies the path to the **logback.xml** file. You created this file (or Web Services created it for you) as part of Deploying the web application.

file

Default Value: `cloud.log`
Valid Values: A valid file name
Mandatory: No
Specifies the name of the log file. This value is stored in *${LOG_FILE}* which may be used in **logback.xml**.

path

Default Value: `/var/log/jetty9`
Valid Values: A valid path
Mandatory: No
Specifies the path to the log file. This value is stored in *${LOG_PATH}* which may be used in **logback.xml**.

jetty

Settings in this section are listed under "jetty".

cookies

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
httpOnly	No	true	If true, it sets an HTTP-only flag for session cookies.
secure	No	false	If true, it sets the secure cookie flag for session cookies.
sameSite	Yes	None	Specifies what should be returned as SameSite cookie attribute value in response for Jetty's session cookie. Valid values are None, Lax, or `Strict`.
partitioned	No	false	If true, it sets the `Partitioned` flag for session cookies.

Mandatory: No
Specifies how Jetty should handle cookies. For example:

```
cookies:
    httpOnly: true
    secure: true
    sameSite: None
```

These options only take effect if **enableSsl** is set to `true`.

host

Default Value: `0.0.0.0`
Valid Values: A host name or IP address
Mandatory: No
Specifies the host name or IP address of the Jetty host. In versions 8.5.201.18 and later, this value should be the same as `GWS_HOST` you defined as part of Deploying the web application.

port

Default Value: `8090`
Valid Values: A valid port
Mandatory: No
Specifies the port of the Jetty host. In versions 8.5.201.18 and later, this value should be the same as `GWS_PORT` you defined as part of Deploying the web application.

idleTimeout

Default Value: `30000`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the maximum idle time, in milliseconds, for a connection.

soLingerTime

Default Value: `-1`
Valid Values: An integer greater than 0, or -1 to disable
Mandatory: No
Specifies the socket linger time.

sessionMaxInactiveInterval

Default Value: `1800`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the period, in seconds, after which a session is deemed idle and saved to session memory.

enableWorkerName

Default Value: `true`
Valid Values: `true`, `false`
Mandatory: No
Specifies whether to add the **WorkerName** parameter into the JSESSIONID cookie.

enableRequestLog

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Enables request logging. If you set the value to `true`, you must also set values for the requestLog option.

requestHeaderSize

Default Value: 8192
Valid Values: Any positive integer value greater than the default value

Specifies the allowed request header size for the Jetty servlet container.

responseHeaderSize

Default Value: 8192
Valid Values: Any positive integer value greater than the default value

Specifies the allowed response header size for the Jetty servlet container.

requestLog

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
filename	No	yyyy_mm_dd.cloud-request.log	Specifies the log file name format.
filenameDateFormat	No	yyyy_MM_dd	Specifies the log file name date format.
logTimeZone	No	GMT	Specifies the timestamp time zone used in the log.
retainDays	No	90	Specifies the time interval, in days, for which Jetty should retain logs.
append	No	true	Specifies whether Jetty appends to the request log file or starts a new file.
extended	No	true	Specifies whether Jetty logs extended data.
logCookies	No	true	Specifies whether Jetty logs request cookies.
logLatency	No	true	Specifies whether Jetty logs the request latency.
preferProxiedForAddress	No	true	Specifies whether Jetty logs IP address or the IP address from the X-Forwarded-For request header.

Mandatory: No
Specifies how Jetty should handle request logging. For example:

```
enableRequestLog: true
  requestLog:
    filename: yyyy_mm_dd.cloud-request.log
    filenameDateFormat: yyyy_MM_dd
    logTimeZone: GMT
    retainDays: 90
    append: true
    extended: true
    logCookies: false
    logLatency: true
    preferProxiedForAddress: true
```

These options only take effect if enableRequestLog is set to `true`.

enableSsl

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Enables Secure Sockets Layer support. If you set the value to `true`, you must also set values for the ssl option.

ssl

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
port	No	443	The SSL port. This option is the equivalent of the Jetty "https.port" variable.
securePort	No	8443	The port to which integral or confidential security constraints are redirected. This option is the equivalent of the Jetty "jetty.secure.port" variable.
idleTimeout	No	30000	The maximum idle time, in milliseconds, for a connection.
soLingerTime	No	-1	The socket linger time. A value of -1 disables this option.
keyStorePath	No	None	The keystore path.
keyStorePassword	No	None	The keystore password.
keyManagerPassword	No	None	The key manager password.
keyStoreProvider	No	None	The keystore provider.
keyStoreType	No	JKS	The key store type.
trustStorePath	No	None	The truststore path.
trustStorePassword	No	None	The truststore password.
trustStoreProvider	No	None	The truststore provider.
trustStoreType	No	JKS	The truststore type.
needClientAuth	No	None	Set this option to true if SSL needs client authentication.
wantClientAuth	No	None	Set this option to true if SSL wants client authentication.
certAlias	No	None	The alias of the SSL certificate for the connector.
validateCerts	No	None	Set this option to true if the SSL certificate has to be validated.
validatePeerCerts	No	None	Set this option to true if SSL certificates of the

Name	Mandatory	Default Value	Description
			peer have to be validated.
trustAll	No	None	Set this option to true if all certificates should be trusted if there is no keySstore or truststore.
renegotiationAllowed	No	None	Set this option to true if TLS renegotiation is allowed.
excludeCipherSuites	No	None	Specifies the array of cipher suite names to exclude from enabled cipher suites.
includeCipherSuites	No	None	Specifies the array of cipher suite names to include in enabled cipher suites.
endpointIdentificationAlgorithm	No	None	Specifies the endpoint identification algorithm. Set this option to "HTTPS" to enable hostname verification.
includeProtocols	No	None	The array of protocol names (protocol versions) to include for use on this engine.
excludeProtocols	No	None	The array of protocol names (protocol versions) to exclude from use on this engine.
enableHsts	No	false	If set to `true`, Genesys Web Services (GWS) provides support for HTTP Strict Transport Security (HSTS) protocol. HSTS prevents Man-in-the-Middle attacks that can occur in unsecure (HTTP) browser sessions. The following string is inserted in the response header: `Strict-Transport-Security: max-age=31536000 ; includeSubDomains` This string tells the browser to not accept any untrusted, expired, or revoked TLS certificates from the domain.

Name	Mandatory	Default Value	Description

enableNonSecureToSecureRedirect	No	false	Redirects HTTP requests to HTTPS.
			When enabled, the following string is sent in the response header:
			`HTTP/1.1 302 Found`
			`Location: https://...`

Mandatory: No
Specifies how Jetty should handle support for Secure Sockets Layer. For example:

```
enableSsl: true
 ssl:
   port: 443
   securePort: 8443
   idleTimeout: 30000
   soLingerTime: -1
```

These options only take effect if **enableSsl** is set to `true`.

cookies

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
httpOnly	No	true	If true, it sets an HTTP-only flag for session cookies.
secure	No	false	If true, it sets the secure cookie flag for session cookies.

Mandatory: No
Specifies how Jetty should handle cookies. For example:

```
cookies:
   httpOnly: true
   secure: true
```

These options only take effect if **enableSsl** is set to `true`.

sessionCookieName

Default Value: `JSESSIONID`
Valid Values: Any string which can be used as a cookie name as per RFC 6265
Mandatory: No
Defines the name of the session cookie used by Web Services.

enableXXSSProtection

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Enables XSS Protection and the following header is added to the response.

```
X-XSS-Protection: 1
```

enableXXSSProtectionBlockMode

Default Value: `true`
Valid Values: `true`, `false`
Mandatory: No
When enabled, blocks the page from being rendered by sending the following header in the response:

```
X-XSS-Protection: 1; mode=block
```

enableXContentTypeOptions

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
When enabled, content sniffing is disabled by sending the following header in the response:

```
X-Content-Type-Options: nosniff
```

xFrameOptions

The X-Frame-Options HTTP response header can be used to indicate whether a browser should be allowed to render a page in a <frame>, <iframe>, or <object>. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.

Default Value: None
Valid Values: DENY, SAMEORIGIN, ALLOW-FROM
Mandatory: No
When enabled, the following header is added to the response.

```
X-Frame-Options: DENY
```

xFrameOptionsAllowFromUri

Default Value: None
Valid Values: A valid URI
Mandatory: No
When enabled, **xFrameOptions** is set to ALLOW-FROM, and only iframes from the specified URI are allowed. The following header is added to the response.

```
X-Frame-Options: ALLOW-FROM https://example.com/
```

This header is not supported by all browsers. For the list of browser types and versions that support the X-Frame-Options header, please refer to Clickjacking Defense Cheat Sheet.

xFrameOptionsExcludedUris

Default Value: None
Valid Values: List of URIs
Mandatory: No
Responses for requests from the specified URIs will not contain the X-Frame-Options header. The URIs must be relative and start with a slash (/) symbol. They must not contain the hostname and the port information. This is a workaround for browsers that do not support xFrameOptions.

> #### Important
>
> If you are using a browser that does not fully support xFrameOptions, it is highly recommended to list the URIs in the **xFrameOptionsExcludedUris** option to exclude these pages from the Clickjacking prevention:
>
> ```
> jetty:
>   ...
>   xFrameOptionsExcludedUris:
>     - /ui/crm-adapter/index.html
>     - /ui/crm-workspace/index.html
>     - /ui/dashboard/index.html
>     - /ui/ad/v1/disaster-recovery.html
> ```

cassandraCluster

Settings in this section are listed under "cassandraCluster".

thrift_port

Default Value: `9160`
Valid Values: A valid port
Mandatory: No
Specifies the port for Thrift to listen for clients. It should be the same as the `rpc_port` you set in the **cassandra.yaml** file as part of the Configuring Cassandra procedure.

jmx_port

Default Value: 7199
Valid Values: A valid port
Mandatory: No
Specifies the port Cassandra uses for Java Manage Extension (JMX).

keyspace

Default Value: `sipfs`
Valid Values: A valid keyspace name
Mandatory: No
Specifies the name of the Cassandra keyspace. This name should be the same as the keyspace name you set while Creating the Cassandra keyspace. If you used the keyspace creation scripts that come with Web Services and Applications, then you can leave this value as `sipfs`.

nodes

Default Value: None
Valid Values: A comma-separated list of IP addresses or host names
Mandatory: No
Specifies the Cassandra node IPs or host names.

backup_nodes

Default Value: None
Valid Values: A comma-separated list of IP addresses or host names
Mandatory: No
Specifies the backup Cassandra node IPs or host names. This option is intended for deployments that have two separate Cassandra data centers — Web Services switches from primary to backup if the primary nodes are unavailable. If your deployment is small with only one data center, you can ignore this option.

replication_factor

Default Value: None
Valid Values: An integer less than the number of nodes in the cluster
Mandatory: No
Specifies a replication factor appropriate for your Cassandra topology. This value should be the same as the replication factor you set in Step 2 of the Creating the Cassandra keyspace procedure.

read_consistency_level

Default Value: None
Valid Values: `CL_ONE`, `CL_QUORUM`, `CL_LOCAL_QUORUM`
Mandatory: No
Specifies the read consistency level appropriate for your Cassandra topology:

Development (1 Cassandra node)	Single Data Center (1 data center with a minimum of three Cassandra nodes)	Two Data Centers (data centers with a minimum of three Cassandra nodes in each data center)
CL_ONE	CL_QUORUM	CL_LOCAL_QUORUM

write_consistency_level

Default Value: None
Valid Values: `CL_ONE`, `CL_QUORUM`, `CL_LOCAL_QUORUM`
Mandatory: No
Specifies the write consistency level appropriate for your Cassandra topology:

Development (1 Cassandra node)	Single Data Center (1 data center with a minimum of three Cassandra nodes)	Two Data Centers (data centers with a minimum of three Cassandra nodes in each data center)
CL_ONE	CL_QUORUM	CL_LOCAL_QUORUM

max_conns_per_host

Default Value: `16`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the maximum number of connections to allocate for a single host's pool.

max_cons

Default Value: `48`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the maximum number of connections in the pool.

max_pending_conns_per_host

Default Value: `80`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the maximum number of pending connection attempts per host.

max_blocked_threads_per_host

Default Value: `160`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the maximum number of blocked clients for a host.

cassandraVersion

Default Value: None
Valid Values: `1.1` or `1.2`
Mandatory: No
Specifies the Cassandra version for your Web Services and Applications deployment.

useSSL

Default Value: None
Valid Values: `true`, `false`
Mandatory: No
Specifies whether Cassandra should use Secure Sockets Layer (SSL). This option is only valid for Cassandra 1.2.x.

serverSettings

Settings in this section are listed under "serverSettings".

URLs

externalApiUrlV2

Default Value: None
Valid Values: A public schema-based URL ending with `/api/v2`.
Mandatory: Yes
Specifies the prefix used for resources in the public API. In a development environment, the host and port should be set to the host name or IP address of the Web Services node. In a production environment, the host and port should be set to the host name or IP address of the load balancer in a production environment. For example, `https://192.0.2.20/api/v2`.

internalApiUrlV2

Default Value: None
Valid Values: A public schema-based URL ending with `/internal-api`.
Mandatory: Yes
Specifies the prefix used for internal resources. In a development environment, the host and port should be set to the host name or IP address of the Web Services node. In a production environment, the host and port should be set to the host name or IP address of the load balancer in a production environment. For example, `http://192.0.2.20/internal-api`.

Platform API Settings

excludeFields

Default Value: -
Valid Values: A list of strings.
Mandatory: No
Enables filtering of specified fields from the Platform Configuration API responses.

Sample

The following setting will filter out the **password** field from the response.

```
...
serverSettings:
  ...
  platformApiSettings:
    excludeFields:
      - password
```

Paths

pathPrefix

Default Value:
Valid Values: A valid prefix
Mandatory: No
Specifies a prefix that Web Services adds to the relative URIs it includes in responses. For example, if you set **pathPrefix** to `/api/v2` and make the following request:

```
GET http://localhost:8080/api/v2/devices
```

Web Services returns the following response:

```
{
   "statusCode":0,
   "paths":[
      "/api/v2/devices/971ed91d-82bf-490b-94d2-02d240165764",
      "/api/v2/devices/a3f9e854-54d8-4260-bea3-d6e450ee7df0"
   ],
   "uris":[
      "http://localhost:8080/api/v2/devices/7c7ab1f7-e596-41bc-9ff4-4a12c489865f",
      "http://localhost:8080/api/v2/devices/a3f9e854-54d8-4260-bea3-d6e450ee7df0"
   ]
}
```

Notice that `paths` includes relative URIs with the `/api/v2` prefix.

General

chatServerRejoinAttempts

Default Value: 5
Valid Values: `Any positive integer`
Mandatory: No
Specifies the number of join attempts to a chat session.

chatServerRejoinDelay

Default Value: `10000`
Valid Values: `Any positive integer`
Mandatory: No
Specifies the delay, in milliseconds, between join attempts to a chat session.

updateInteractionForSetContact

Default Value: `false`

Valid Values: true, false
Mandatory: No
Introduced: 8.5.202.96
Specifies whether Web Services RequestUpdateInteraction instead of RequestAssignInteractionToContact when the SetContact request is called. Using RequestUpdateInteraction prevents UCS from updating the ThreadId interaction attribute. In some scenarios, updates to this attribute can cause an interaction to become orphaned in the UCS database.

sendParticipantsUpdatedBeforeStatusChange

Default Value: false
Valid Values: true, false
Mandatory: No
Introduced: 8.5.202.84
Specifies the order in which GWS sends ParticipantsUpdated and StatusChange notifications. To make sure that after reconnecting to Chat Server GWS send the ParticipantsUpdated notification first and then the StatusChange notification, set the value of this option to true.

disableCreatorAppIdUpdates

Default Value: false
Valid Values: true, false
Mandatory: No
Introduced: 8.5.202.84
Preserves the original CreatorAppId of an interaction in UCS. Set the value of this option to true to make sure that the CreatorAppId is not changed after reconnecting to Chat Server.

enableAgentWorkbinStatisticsOptimization

Default Value: false
Valid Values: true, false
Mandatory: No
Turns on synchronization of the users_index_employee_id table, which improves the performance of the RequestStats operation. It is necessary to perform a force synchronization of Cassandra with Configuration Server after enabling this option.

> **Important**
>
> Applicable only for environments with a large number of agents and only when delay is observed on the GWS side.

enableIntermediateParticipantNicknameFix

Default Value: false
Valid Values: true, false
Mandatory: No
Specifies how the nickname of a participant in the chat message of the application CometD notification is displayed. By default (the false value of the option), the latest nickname of a participant is displayed for all messages in the chat transcript. When this option is set to true,

messages of the participant who changed the nickname are displayed with the actual nickname at the time when messages were sent.

enableNotificationOnPullChat

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Enables sending of a notification with interaction properties while pulling chat interaction from a workbin. By default, the same notification is sent to other media types.

Add the following notification to the application.yaml file:

```
...
serverSettings:
  ...
  enableNotificationOnPullChat: true
```

enableJoinOnPullChat

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Enables joining to the chat session of the chat interaction that is being pulled from a workbin. Following notification is a chat transcript similar to the one which is sent for the Accept operation.

Add the following notification to the application.yaml file:

```
...
serverSettings:
  ...
  enableJoinOnPullChat: true
```

enableSaveEmailReplyUserDataToUCS

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Introduced: 8.5.202.85
Enables GWS to save the UserData of the parent email interaction to the child email interaction for Reply and ReplyAll operations. When set to `true`, GWS sends the AttachedData filled with the UserData of the parent interaction to UCS and UCS can render the corresponding field codes in Standard Responses.

enableUcsOrphanedScrollCleanup

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
When enabled, Web Services cleans up the orphan "Scrolls" from Universal Contact Server (UCS) that consume memory. When Web Services gets a list of interactions from UCS that has more than 1000 results, it creates a "Scroll" that must be released by Web Services when it is no longer needed.

enableFindOrCreateCallSearchInMemory

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
When enabled, Web Services searches for a call in memory, if the call is not found in Cassandra.

enableSyncConnectionToIxnServer

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
When enabled, Web Services waits for a configured amount of time for the Interaction Server connection to open before trying to send requests. You can use the **syncConnectionToIxnServerTimeout** option to adjust the wait time.

syncConnectionToIxnServerTimeout

Default Value: `3000`
Valid Values: positive integer
Mandatory: No
Defines the wait time for opening the connection to Interaction Server when the **enableSyncConnectionToIxnServer** option is enabled.

enableNotReadyOnConferenceInviteExpiration

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
When enabled, Web Services changes an agent's state when the agent does not answer an invite to Consult or Conference by a queue (routing-based). You can configure the status, which will be set using the **statusNameOnConferenceInviteExpiration** option.

statusNameOnConferenceInviteExpiration

Default Value: `NotReady`
Valid Values: name of the agent's status
Mandatory: No
Defines the status which will be set when the agent does not answer an invite to Consult or Conference by a queue (routing-based), and the **enableNotReadyOnConferenceInviteExpiration** is enabled.

enableEmailCaseInsensitive

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Enables enforcing case-insensitive comparison of email addresses to remove the sender's address from recipients lists ("To", "Cc", "Bcc") while replying all in an email interaction.

enableInteractionRequestPull

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
When enabled, Web Services gets the current interaction from Interaction Server if an agent is disconnected from one Web Services node and is reconnected to another. This feature is only applicable if the agent reconnects using the same Workspace Web Edition instance. It does not apply to any active consultations or supervisors associated with the interaction. Previously, it was possible to have more than one agent in the same chat session when an agent switched connections to a new node while handling an interaction.

iwsDispositionCodeSync

Default Value: `10000`
Valid Values: positive integer
Mandatory: No
Enables Web Services to synchronize disposition codes with Workspace Web Edition. To turn on this feature, set **iwsDispositionCodeSync** to `true` on all nodes (this is done by default), and set the syncNode option to true on one node.

temporaryAuthenticationTokenTTL

Default Value: 300
Valid Values: An integer greater than 0
Mandatory: No
Specifies the time to live, in seconds, for the temporary authentication token.

enableCsrfProtection

Default Value:
Valid Values: `true`, `false`
Mandatory: No
Enables cross site request forgery protection. If you set the value to `true`, make sure you use the default values for **exposedHeaders** in the crossOriginSettings option. If you have already updated the **exposedHeaders**, just make sure the values include the defaults.

enableOpenIDConnect

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Specifies whether Web Services uses OAuth 2.0 authentication.

enableStaleSessionsMonitoring

Default Value: `true`
Valid Values: `true`, `false`
Mandatory: No
Specifies whether Web Services should run its StaleSessionsMonitor process to periodically poll Cassandra for expired CometD sessions. This process releases any devices that might not have been released as part of EndContactCenterSession if CometD session information was lost.

requestTakeSnapshotTimeout

Default Value: `10000`
Valid Values: positive integer
Mandatory: No
Defines the timeout for the **GetContent** operation. This API request leads to sending **RequestTakeSnapshot** to Interaction Server. In some cases, when there are lots of interactions in a queue, this request could take a long time.

staleSessionsMonitorSettings

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
monitoringInterval	No	60	Specifies, in seconds, how often Web Services scans for expired CometD sessions.
expiredSessionAge	No	180	Specifies the age, in seconds, at which Web Services considers the CometD session to be expired.

Mandatory: No
Specifies the configuration for monitoring expired CometD sessions. For example:

```
...
staleSessionsMonitorSettings:
    monitoringInterval: 60
    expiredSessionAge: 180
```

includeMessageType

Default Value: `false`
Valid Values: `true, false`
Mandatory: No
Specifies whether to include the original message type of a chat message in the MessageLogUpdated CometD notification when you make a SendMessage request with the Chat API.

enableInteractionPropertiesForStandardResponse

Default Value: `false`
Valid Values: `true, false`
Mandatory: No
Introduced: 8.5.202.90
Specifies whether Web Services uses the interaction properties from Interaction Server to render standard responses instead of using the interaction attributes from Universal Contact Server.

enableSpecificTwoStepTransferForAvayaSwitch

Default Value: false
Valid Values: true, false
Mandatory: No
Introduced: 8.5.202.94
When set to `true`, Web Services enables agents to complete a call transfer to a consultation target while the consultation call is on hold.

> Important
> This option is used for Avaya switch environments only.

enableStatusForOfflineChatOnRecovery

Default Value: false
Valid Values: true, false
Mandatory: No
Introduced: 8.5.202.97
When set to `true`, if an offline chat interaction is restored for an agent on a different node, the interaction has the 'LeftChat' status along with the list of corresponding capabilities.
To ensure that interactions are recovered as well as having the status set, Genesys recommends that when using this feature, you also set the value of the **enableSyncConnectionToIxnServer** to `enabled`.

enablePutOnHoldInWorkbin

Default Value: false
Valid Values: true, false
Mandatory: No
Introduced: 8.5.202.97
Specifies whether or not chat interactions can be put on hold in a workbin and then reopened later to continue the chat.

enableChatSynchronization

Default Value: false
Valid Values: true, false
Mandatory: No
Introduced: 8.5.202.97
Enable this option so that Web Services and Applications will display all chat messages for the current chat session. Previously, few messages could be lost if they were sent closely after an agent joined the chat session.

Timeouts

activationFailFastPeriod

Default Value: `10000`
Valid Values: Any integer greater than 0
Mandatory: Yes
Determines fast-fail period length after a failure. Any attempt to reconnect to a Genesys server within the time specified by the **activationFailFastPeriod** option results in an immediate failure. To understand how this option works, consider this scenario:

- Agent A logs in to a GWS node. Note that connection to tserver is not active yet. GWS attempts to connect to tserver and the connection attempt fails.
- Agent B attempts to login within the **activationFailFastPeriod**. GWS does not attempt to connect to tserver. The agent is authenticated, but is not logged into the voice channel.
- After **activationFailFastPeriod** expires, Agent C attempts to login. GWS connects to TServer if the authentication is successful. Note that if the connection attempt fails, the activationFailFastPeriod is restarted again.

activationTimeout

Default Value: `12000`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the timeout, in milliseconds, for connecting to any Genesys server (except Configuration Server). This may include several individual attempts if the initial attempt to connect is unsuccessful.

> **Important**
> The activation timeout for Configuration Server is specified with the `configServerActivationTimeout` option.

chatServerConnectionTimeout

Default Value: `7000`
Valid Values: Any positive integer
Mandatory: No
Specifies the timeout, in milliseconds, after which the attempt to connect to the Chat Server fails.

chatServerReconnectTimeout

Default Value: 10000
Valid Values: Any positive integer
Mandatory: No
Specifies the delay, in milliseconds, between attempts to connect to the Chat Server.

configServerActivationTimeout

Default Value: `35000`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the timeout, in milliseconds, for connecting to Configuration Server. This may include several individual attempts if the initial attempt to connect is unsuccessful.

configServerConnectionTimeout

Default Value: `15000`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the timeout, in milliseconds, for an individual connection attempt to Configuration Server.

connectionTimeout

Default Value: `4000`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the timeout, in milliseconds, for an individual connection attempt to any Genesys server (except Configuration Server).

> **Important**
>
> The connection timeout for Configuration Server is specified with the `configServerConnectionTimeout` option.

contactCenterSynchronizationTimeout

Default Value: 60000
Valid Values: An integer greater than 0
Mandatory: No
Specifies how often the syncNode looks for newly created contact centers.

inactiveUserTimeout

Default Value: `60`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the interval, in seconds, at which the inactive user cleanup process is run by the server. This process is run to invalidate HTTP sessions for users who have been deleted or whose user roles have changed.

reconnectAttempts

Default Value: 1
Valid Values: An integer greater than 0
Mandatory: Yes

Specifies the number of attempts Web Services makes to connect to any Genesys server before attempting to connect to the backup.

reconnectTimeout

Default Value: 10000
Valid Values: An integer greater than 0
Mandatory: Yes
Specifies the timeout, in milliseconds, between the reconnect attempts.

agentSessionCleanUpTimeout

Default Value: 60000 (ms)
Valid Values: Any positive integer.
Mandatory: No
Specify the timeout in milliseconds before the agent session cleanup procedure is initiated.

platformConfigurationReadTimeout

Default value: 10000
Valid values: Any positive integer.
Specify the timeout (in milliseconds) for platform configuration read requests. If an invalid or a negative value is provided, a warning message will be logged and the default value would be applied.

OPS account

opsUserName

Default Value: None
Valid Values: Any alphanumeric value that can include special characters
Mandatory: Yes
Specifies the name of the Web Services super user. Web Services creates this user at startup.

opsUserPassword

Default Value: None
Valid Values: Any alphanumeric value, including special characters
Mandatory: Yes
Specifies the password for the Web Services super user. Web Services creates this user at startup.

Configuration Server credentials

applicationName

Default Value: None
Valid Values: A valid application name
Mandatory: Yes
The name of the Web Services node application object in Configuration Server. For example, `WS_Node`.

applicationType

Default Value: None
Valid Values: A valid application type
Mandatory: Yes
The type of the Web Services node application object in Configuration Server. This value should be `CFGGenericClient`.

cmeUserName

Default Value: None
Valid Values: A valid Configuration Server user
Mandatory: Yes
The username that the Web Services server uses to connect to Configuration Server.

> **Important**
>
> Genesys recommends that you use the provided "default" account in Configuration Server. It is possible to use a different account, but you must take care in configuring the user's account permissions. Outside of a lab setting, this is best done in consultation with Genesys.

cmePassword

Default Value: None
Valid Values: A valid password
Mandatory: Yes
The password for the Configuration Server user Web Services uses to connect to Configuration Server.

syncNode

Default Value: None
Valid Values: `true, false`
Mandatory: No
Specifies whether the node is the synchronization node. This node is responsible for importing objects from Configuration Server into Cassandra, subscribing to changes notifications with Configuration Server, and processing updates.

> **Important**
>
> In each Web Services cluster, one node must be configured as the synchronization node: `syncNode = true`. All other nodes in the cluster must have `syncNode = false`.

synchronizationCmeEventsPrefilterEnabled

This option is deprecated in 8.5.2.

enableVirtualQueueSynchronization

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Specifies whether the syncNode imports virtual queues from the Configuration Layer.

Statistics

locationAwareMonitoringDistribution

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No

Enables you to configure additional connections to different StatServers. GWS chooses the one that is "visible" from the GWS node (based on the location specified in the connection).

Set this option to `true` on all nodes only when deploying multiple data centers.

> **Important**
>
> - This option applies only to multi-data center environments.
> - The statistics collected in a multi-data center environment will not be displayed properly without both this option and the **enableMultipleDataCenterMonitoring** option set to `true`.

enableMultipleDataCenterMonitoring

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No

Enables statistics collection for each data center in a multiple data center configuration.

Set this option to `true` on all nodes only when deploying multiple data centers.

> **Important**

- This option applies only to multi-data center environments.
- The statistics collected in a multi-data center environment will not be displayed properly without both this option and the **locationAwareMonitoringDistribution** option set to `true`.

statConnectionTimeout

Default Value: `5000`
Valid Values: A positive integer greater than 0
Mandatory: No
Specifies the connection timeout, in milliseconds, for connecting to Stat Server.

statReconnectAttempts

Default Value: `1`
Valid Values: A positive integer
Mandatory: No
Specifies the number of reconnect attempts before switching to the backup Stat Server, if the connection to the primary Stat Server is lost.

statReconnectTimeout

Default Value: `10000`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the timeout, in milliseconds, before reconnecting to Stat Server.

statOpenTimeout

Default Value: `60000`
Valid Values: An integer greater than 0
Mandatory: No
Specifies the timeout, in milliseconds, between when a request is sent to Stat Server to open a statistic and when Web Services server determines the statistic has not been opened. If the timeout expires, the Web Services server discards the request and sends a new one.

statisticsWritesCL

This option is no longer applicable as of release 8.5.201.09.

statisticsMonitorMultimediaChannelStates

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Specifies whether to monitor user states on non-voice channels.

reportingSyncInterval

Default Value: 30
Valid Values: An integer greater than 0
Mandatory: No
Specifies the interval, in seconds, for the reporting services to poll the database for information about the activities of other monitoring nodes and for the current state of the contact center configuration. If you set this option to a larger value, it decreases the load on the database, but also increases the timeout for detecting nodes that are down and objects that are added or updated in contact centers.

> **Important**
>
> The value of this option specifies the rate of scheduling, not the delay between when the previous polling finishes and the new polling starts.

enableElasticSearchIndexing

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Specifies whether the configuration information and statistics should be indexed to Elasticsearch. If set to `true`, you must set the crClusterName option.

statisticsOpenRetryInterval

Default Value: `60` (1 hour)
Valid Values: An integer greater than 0
Mandatory: No
Specifies the interval, in minutes, for trying to reopen failed statistics. For example, if a statistic cannot be opened on Stat Server, it is marked as failed and then the server attempts to reopen the stat once every hour (the default value of **statisticsOpenRetryInterval**).

Multi regional supporting

nodePath

Default Value: None
Valid Values: A location and node ID, separated by a "/" — for example, `/US/node1`
Mandatory: Yes
Specifies the location and ID of the Web Services node within the deployment topology. This value must be unique across the deployment. For example, a value of `/US/node1` means that the node is located in the US region and has an ID of "node1". The node ID can be the hostname, the IP address, or any other unique identifier.

nodeId

Default Value: None
Valid Values: Any unique identifier, such as the node host name or IP

Mandatory: No
Specifies the unique identifier for the Web Services node. Each node in a cluster must have a unique `nodeId`.

SSL and CA

caCertificate

Default Value: None
Valid Values: Path to a signed certificate
Mandatory: No
Specifies the path to a certificate signed by a Certificate Authority. The file must be in the `.pem` or `.jks` format (if `.jks`, you can also set jksPassword). The certificate can be used if the WS_Cluster application uses Transport Layer Security (TLS) to connect to Genesys servers. This option is also mandatory to enable SAML authentication.

jksPassword

Default Value: None
Valid Values: Password for the key storage
Mandatory: No
Specifies the password for the key storage set in caCertificate, when the certificate is in `.jks` format. This option is mandatory to enable SAML authentication.

SAML

samlSettings

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
serviceProviderEntityId	No	If omitted, Web Services uses the value of the externalApiUrlV2 option.	Specifies the service provider entity ID to be used in the metadata.
encryptionKeyName	Yes	None	Specifies the Security Assertion Markup Language (SAML) encryption key name. This key has to be present in the JKS key storage specified in the caCertificate option.
encryptionKeyPassword	No	If omitted, Web Services uses the value of the jksPassword option.	Specifies the password used to extract the SAML encryption key from JKS storage.
signMetadata	No	true	Specifies whether generated metadata is signed with an XML

Name	Mandatory	Default Value	Description
			signature that uses the certificate with the alias of **signingKeyName**.
signingKeyName	Yes	None	Specifies the SAML signing key name. This key has to be present in the JKS key storage specified in the caCertificate option.
signingKeyPassword	No	If omitted, Web Services uses the value of the jksPassword option.	Specifies the password used to extract the SAML signing key from JKS storage.
tlsKeyName	No	None	Specifies the TLS key name. This key has to be present in the JKS key storage specified in the caCertificate option.
tlsKeyPassword	No	If omitted, Web Services uses the value of the jksPassword option.	Specifies the password used to extract the SAML TLS key from JKS storage.
responseSkewTime	No	60	Specifies the maximum difference, in seconds, between the local time and time of the assertion creation which still allows message to be processed. Determines the maximum difference between clocks of the IDP and SP servers.
defaultBinding	No	SSO_ARTIFACT	Specifies the default SAML binding Web Services uses. The valid values are: SSO_POST, SSO_PAOS, SSO_ARTIFACT, HOKSSO_POST, or HOKSSO_ARTIFACT.
requestSigned	No	true	Specifies whether this service signs authentication requests.
wantAssertionSigned	No	true	Specifies whether this service requires signed assertions.
identityProviderMetadata	Yes	None	Specifies the path or URL for the identity provider XML metadata file. You can set this

Name	Mandatory	Default Value	Description
			option to the path to a physical location or, if the metadata file is exposed by the remote server over HTTP, you can specify the URL (in this case, Web Services applies a 5-second default request timeout).
secureRelayState	No	false	If false, the RelayState attribute Web Services passes to the Identity Provider during SAML authentication contains the original URL (from the "referer" header) requested by the client. If true, Web Services instead saves the original RelayState in Cassandra and uses an alphanumeric token that can identify the RelayState.
secureRelayStateTTL	No	3600 (1 hour)	Specifies, in seconds, the time-to-live for the RelayState record in Cassandra. This setting is only used if **secureRelayState** is set to true.
useExternalUserId	No	false	Specifies whether Web Services should use the external user ID, a property of the CfgPerson object in the Configuration Database , for SAML authentication. If false, Web Services uses the username. If true, Web Service uses the external user ID.
maxAuthenticationAge	No	7200 (2 hours)	Specifies the maximum time, in seconds, between a user's authentication and when the authentication statement is processed.

Mandatory: No
Specifies the configuration for Security Assertion Markup Language (SAML) authentication for Web Services. For example:

```
...
samlSettings:
    serviceProviderEntityId: 10.10.15.60
    encryptionKeyName: client
    signingKeyName: client
    identityProviderMetadata: http://ipd.company.host/saml/metadata/idp-metadata.xml
    responseSkewTime: 120
    defaultBinding: SSO_POST
```

CORS

crossOriginSettings

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
allowedOrigins	No	None	Specifies a comma-separated list of allowed origins supported by this Web Services node. For example, http://*.genesys.com, http://*.genesyslab.com
allowedMethods	No	GET,POST,PUT,DELETE,OPTIONS	Specifies a comma-separated list of HTTP methods supported by the server.
allowedHeaders	No	X-Requested-With,Content-Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId	Specifies whether to include the Access-Control-Allow-Headers header as part of the response to a pre-flight request. This specifies which header field names can be used during the actual request.
allowCredentials	No	true	Specifies the value of the Access-Control-Allow-Credentials header. This should typically be left at the default value.
corsFilterCacheTimeToLive	No	120	Specifies the delay after the contact center allowDomain updating takes effect.
exposedHeaders	No	X-CSRF-HEADER,X-CSRF-TOKEN	Specifies which custom headers are allowed in cross-origin HTTP responses. This should typically be left at the

Name	Mandatory	Default Value	Description
			default value. If you do modify the value and you enable the enableCsrfProtection option, make sure the value for **exposedHeaders** includes `X-CSRF-HEADER,X-CSRF-TOKEN`.

Mandatory: No
Specifies the configuration for cross-origin resource sharing in Web Services. For example:

```
...
crossOriginSettings:
    corsFilterCacheTimeToLive: 120
    allowedOrigins: http://*.genesys.com, http://*.genesyslab.com
    allowedMethods: GET,POST,PUT,DELETE,OPTIONS
    allowedHeaders: "X-Requested-With,Content-Type,Accept,
Origin,Cookie,authorization,ssid,surl,ContactCenterId"
    allowCredentials: true
    exposedHeaders: "X-CSRF-HEADER,X-CSRF-TOKEN"
```

Elasticsearch

elasticSearchSettings

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
clientNode	No	false	Specifies whether the current Web Services node acts as an Elasticsearch client or server.
crClusterName	No	None	Specifies the name of the cluster to enable search functionality in Elasticsearch. If this option is not present, search functionality is not enabled.
indexPerContactCenter	No	false	Enables indexing on a per-contact center basis. If false, there is only one index for all current statistics.
enableScheduledIndexVerification	No	false	Enables scheduled index verification on the

Name	Mandatory	Default Value	Description
			Web Services node. This means that the node goes through all objects handled by Elasticsearch and makes sure they still exist in Cassandra and vice versa.
indexVerificationInterval	No	720 minutes (12 hours)	Specifies an interval, in minutes, between index verifications for this Web Services node.
enableIndexVerificationAtStartUp	No	true	Enables index verification at start-up on this node. This means that, at start-up, the node goes through all objects handled by Elasticsearch and makes sure they still exist in Cassandra and vice versa.
retriesOnConflict	No	3	Controls how many times to retry if there is a version conflict when updating a document.
useTransportClient	No	false	Specifies whether Web Services should use a transport client for Elasticsearch. If true, then Web Services ignores the **clientNode** setting.
transportClient	Yes, if **useTransportClient** is true.	Values specified in TransportClientSettings	TransportClientSettings]] in the next table.

TransportClientSettings

Name	Mandatory	Default Value	Description
nodes	Yes, if **useTransportClient** is true.	null	Specifies the list of Elasticsearch nodes the transport client should connect to.
useSniff	no	false	Specifies if the transport client should us sniffing functionality and perform auto-discovery of Elasticsearch nodes in the cluster.

Name	Mandatory	Default Value	Description
ignoreClusterName	no	false	Specifies if Web Services should ignore the name of the cluster when connecting to the cluster.
pingTimeout	no	5000	Specifies, in milliseconds, the ping timeout for Elasticsearch nodes.
nodesSamplerInterval	no	5000	Specifies, in milliseconds, how often Web Services should sample/ping the Elasticsearch nodes listed and connected.

Mandatory: No
Specifies the configuration for Elasticsearch in Web Services. For example:

```
...
elasticSearchSettings:
    clientNode: true
    indexPerContactCenter: false
    enableScheduledIndexVerification: true
    indexVerificationInterval: 60
    retriesOnConflict: 2
    useTransportClient: true
    transportClient:
        nodes:
             - {host: 127.0.0.1, port: 9300}
        useSniff: true
        ignoreClusterName: true
        pingTimeout: 10000
        nodesSamplerInterval: 10000
```

Screen Recording

screenRecordingSettings

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
screenRecordingVoiceEnabled	No	false	Specifies whether the current Web Services node supports screen recording for voice interactions. If false, the node rejects CometD requests from the ScreenRecording Client for agents with the voice channel.

Name	Mandatory	Default Value	Description
screenRecordingEServicesEnabled	No	false	Specifies whether the current Web Services node supports screen recording for non-voice interactions. If false, the node rejects CometD requests from the ScreenRecording Client for agents with the eServices channel.
recordingInteractionEventsTTL	No	172800	Specifies the time to live (TTL) for Cassandra to cache a screen recording interaction event.
clientSessionManagerCacheTTL	No	60	Specifies the TTL for the Web Services node to cache agent information (such as the agent's name) so that the node doesn't have to read the information from Web Services on each request.
contactCenterInfoManagerCacheTTL	No	90	Specifies the TTL for the Web Services node to cache contact center information so that the node doesn't have to read the information from Web Services on each request.

Mandatory: No
Specifies the screen recording configuration parameters. For example:

```
...
screenRecordingSettings:
   screenRecordingVoiceEnabled: false
   screenRecordingEServicesEnabled: false
   recordingInteractionEventsTTL: 172800
   clientSessionManagerCacheTTL: 60
   contactCenterInfoManagerCacheTTL: 90
```

Caching

cachingSettings

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
agentStatesTTL	No	30	The time to live (TTL), in seconds, for contact-center agent states in cache.
businessAttributesTTL	No	30	The time to live (TTL), in seconds, for contact-center business attributes in cache.
businessUnitsWithSubresourcesTTL	No	30	The time, in seconds, when cache is re-read and updated from Cassandra.
cleanupPeriod	No	1800	The interval, in seconds, between caches cleanup (eviction of expired elements).
contactCenterFeaturesTTL	No	30	The time to live (TTL), in seconds, for contact-center feature IDs in cache.
contactCenterSettingsTTL	No	30	The time to live (TTL), in seconds, for contact-center custom settings in cache.
enableBlockingStrategyForBusinessUnitsWithSubresourcesCache	No	false	Specifies whether business units with subresources cache should utilize blocking approach or not.
enableBlockingStrategyForContactCenterSettingsCache	No	false	Specifies whether ContactCenter settings cache should utilize blocking approach or not.
enableBusinessUnitsWithSubresourcesCaching	No	false	Specifies whether business units with subresources should be served via cache or via direct Cassandra reads.
enableSystemWideBlockingStrategy	No	false	Specifies whether all caches should utilize blocking approach or not.
enableSystemWideCaching	No	false	Specifies if caching is used system-wide (`true`) or only during statistics evaluation (`false`).
skillsTTL	No	30	The time to live (TTL), in seconds, for contact-

Name	Mandatory	Default Value	Description
			center skills in cache.
transactionsTTL	No	30	The time to live (TTL), in seconds, for contact-center transactions in cache.
virtualAgentGroupsTTL	No	30	The time to live (TTL), in seconds, for contact-center virtual agent groups in cache.
voiceContextCaching	No	false	Specifies whether to use in-memory cached context for processing voice events. Using caching reduces the processing time, but you can expect delays in configuration information propagation.
voiceContextRefreshInterval	No	30	The interval, in seconds, between refreshes of the context caches for voice event processing. The service reads configuration information from the database and then refreshes the corresponding caches.

Mandatory: No
Specifies how Web Services should handle various caching scenarios. For example:

```
...
cachingSettings:
   enableSystemWideCaching: true
   agentStatesTTL: 30
   contactCenterFeaturesTTL: 30
   contactCenterSettingsTTL: 30
   voiceContextCaching: true
   voiceContextRefreshInterval: 60
```

DoS Filter

enableDosFilter

Default Value: false
Valid Values: true, false
Mandatory: No
Enables the denial of service filter. If you set the value to true, you must also set values for the dosFilterSettings option.

dosFilterSettings

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
maxRequestsPerSec	No	25	Specifies the maximum number of requests from a connection per second. Requests that exceed this are first delayed, then throttled.
delayMs	No	100	Specifies the delay, in milliseconds, imposed on all requests over the rate limit, before they are considered at all. Valid values: • -1 = reject request • 0 = no delay • Any other number = delay in milliseconds
maxWaitMs	No	50	Specifies the length of time, in milliseconds, to blocking wait for the throttle semaphore.
throttledRequests	No	5	Specifies the number of requests over the rate limit that are able to be considered at once.
throttleMs	No	30000	Specifies the length of time, in milliseconds, to asynchronously wait for semaphore.
maxRequestMs	No	30000	Specifies the length of time, in milliseconds, to allow the request to run.
maxIdleTrackerMs	No	30000	Specifies the length of time, in milliseconds, to keep track of request rates for a connection, before deciding that the user has gone away, and discarding the connection.
insertHeaders	No	true	If true, DoSFilter headers are inserted into the response.
trackSessions	No	true	If true, the usage rate is

Name	Mandatory	Default Value	Description
			tracked by session if a session exists.
remotePort	No	false	If true and session tracking is not used, then the rate is tracked by IP + port (effectively connection).
ipWhitelist	No	""	A comma-separated list of IP addresses that is not rate limited.

Mandatory: No
Specifies how Web Services should handle denial of service. For example:

```
...
enableDosFilter: true
dosFilterSettings:
   maxRequestsPerSec: 30
   ipWhitelist: 192.168.0.1,192.168.0.2
```

These options only take effect if enableDosFilter is set to `true`.

Account management

accountManagement

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description

forgotPasswordEmailTemplate	No	None	The template to use for an email that is sent to a user who forgets his or her password.
			`forgotPasswordEmailTemplate:`
			`             from: <from address>`
			`subject: <Subject line>`
			`             body: <Email body>`

accountCreatedEmailTemplate	No	None	The template to use for a email that is sent to a user who creates a new account. `accountCreatedEmailTemplate:`

Name	Mandatory	Default Value	Description
			from: <from address> subject: <Subject line> body: <Email body>
smtpServer	No	None	The SMTP server configuration information. smtpServer: host: <smtp host name> port: <smtp port> userName: <user name for the account> password: <password in plain text> timeout: <optional SMTP timeout>

Mandatory: No
Specifies the configuration for the email notification and email server used when creating a new user. This option accepts the email server's login configuration, as well as email templates for resetting and creating user passwords. For example:

```
accountManagement:
     forgotPasswordEmailTemplate:
             from: <from address>
             subject: <Subject line>
             body: <Email body>
     accountCreatedEmailTemplate:
             from: <from address>
             subject: <Subject line>
             body: <Email body>
     smtpServer:
             host: <smtp host name>
             port: <smtp port>
             userName: <user name for the account>
             password: <password in plain text>
             timeout: <optional SMTP timeout>
```

CometD

cometDSettings

Default Value: None
Valid Values:

Name	Mandatory	Default Value	Description
cometdSessionExpirationTimeout	No	60	Specifies the timeout for the CometD session to expire on disconnect. It might take an additional minute for the session to be closed after it expires. If you set this option to -1, the session never expires. An agent can login again before the end of this timeout to disable session expiration.
closeHttpSessionOnCometDExpiration	No	true	Enables or disables HTTP session invalidation when CometD times out.
cookieHttpOnly	No	true	If true, it sets an HTTP-only flag for CometD's session cookies.
cookieSecure	No	false	If true, it sets the secure cookie flag for CometD's session cookies.
cookieSameSite	Yes	None	Specifies what should be returned as SameSite cookie attribute value in response for CometD's session cookie. Valid values are: None, Lax, or Strict.
maxSessionsPerBrowser	No	1	The maximum number of sessions (tabs/frames) allowed to long poll from the same browser; a negative value allows unlimited sessions.
multiSessionInterval	No	2000	Specifies the period of time, in milliseconds, for the client normal polling period, in case the server detects more sessions (tabs/frames) connected from the same browser than allowed by the `maxSessionsPerBrowser` parameter. A non-positive value means that additional sessions will be disconnected.

Mandatory: No
Specifies the configuration for the CometD-specific transport server embedded into the Web Services application. For example:

```
cometDSettings:
    cometdSessionExpirationTimeout: 60
    closeHttpSessionOnCometDExpiration: true
    maxSessionsPerBrowser: 2
    multiSessionInterval: 4000
```

And specifies how CometD's session cookie should be handled. For example:

```
cometDSettings:
    cookieHttpOnly: true
    cookieSecure: true
    cookieSameSite: None
```

Cookie options take effect only if **enableSsl** is set to true.

Log header

enableLogHeader

Default Value: `true`
Valid Values: `true`, `false`
Mandatory: No
Specifies whether Web Services includes a header in its main log file. This header contains key information about the Web Services installation, including the version, start time, libraries, and any applicable settings from the **applications.yaml** file.

excludeAppSettings

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Specifies if application setting should be excluded from the Log Header functionality.

Routing Point Monitoring

enableRPMonitoring

Default Value: `false`
Valid Values: `true`, `false`
Mandatory: No
Specifies whether Web Services will support call supervision for monitoring routing points.

Switch Prefixes

enableDialingPlanAvayaSwitchPrefixProcessing

Default Value: `false`
Valid Values: `true`, `false`

Mandatory: No
When enabled, Web Services will compare participant numbers with and without prefixes on outbound calls to avoid rendering redundant participants.

dialingPlanAvayaSwitchPrefix

Default Value: 9
Valid Values: Any positive integer
Mandatory: No
Specifies the outbound call prefix used on Avaya switches.

updateOnStartup

> ## Important
>
> In previous releases, a GWS server restart automatically updated global information. Now the updates are set to false by default. If you want to configure the GWS to update all global information, use the following configuration sample:
>
> ```
> updateOnStartup:
>         statistics: true
>         opsCredentials: true
>         features: true
> ```

opsCredentials

Default Value: false
Valid Values: true, false
Mandatory: No
Specifies whether to update the global ops credentials to the values specified in configuration file.

statistics

Default Value: false
Valid Values: true, false
Mandatory: No
Specifies whether to update statistic definitions to the values specified in corresponding configuration file. This option is taken into consideration only on StatNodes.

features

Default Value: false
Valid Values: true, false
Mandatory: No
Specifies whether to update feature definitions to the values specified in corresponding configuration file.

SIP Cluster Support

useEmployeeIdAsAgentLoginForSIPCluster

Default Value: false
Valid Values: true, false
Mandatory: No
Specifies if Web Services has to use the employee ID of the agent as the login ID.

> **Important**
> This option is used in SIP Cluster environments only.

onPremiseSettings

Settings in this section are listed under "onPremiseSettings".

cmeHost

Default Value: None
Valid Values: A valid IP address or host name
Mandatory: No
Specifies the Configuration Server host name (FQDN) or IP.

cmePort

Default Value: None
Valid Values: A valid port
Mandatory: No
Specifies the Configuration Server port.

backupCmeHost

Default Value: None
Valid Values: A valid IP address or host name
Mandatory: No
Specifies the backup Configuration Server host name (FQDN) or IP. You should only configure this option if there is a backup Configuration Server in the Genesys environment and you want high-availability support.

backupCmePort

Default Value: None
Valid Values: A valid port
Mandatory: No

Specifies the backup Configuration Server port. You should only configure this option if there is a backup Configuration Server in the Genesys environment and you want high-availability support.

countryCode

Default Value: None
Valid Values: A two-letter country code
Mandatory: No
The premise contact center's country code. For example, US.

tlsEnabled

Default Value: None
Valid Values: `true`, `false`
Mandatory: No
Specifies whether Web Services should create a secured connection to Configuration Server the first time it starts.

StandardResponse Caching

Overview

Default cache setting is one hour. However, the tree is not cached, only request to UCS is cached.

```
Default timeToLiveSeconds = 3600; maxEntriesLocalHeap = 10000.
```

Example

```
Request to GetRootCategories – it will be cached for 1 hour.
Request to GetCategory(x) – it will be cached for 1 hour.
You make any changes to Category x – you will not receive it for 1 hour.
You make a change to Category y.
You make a request to GetCategory(y) – you will receive the latest from the server and it
will be cached for 1 hour.
```

Cache Managing

You must provide the following settings in the **application.yaml** file to change default parameters for caching:

```
serverSettings:
 cachingSettings:
   dedicatedCacheSettings:
      - cacheName: ContactServerCategoriesCache
        timeToLiveSeconds: <TTL>
        maxEntriesLocalHeap: <Heap size>
      - cacheName: ContactServerStandardResponsesCache
        timeToLiveSeconds: <TTL>
        maxEntriesLocalHeap: <Heap size
```

Gplus Adapter for Salesforce

The Gplus Adapter for Salesforce is an integrated solution that enables Salesforce users to handle contact center interactions seamlessly within Salesforce. The adapter is part of the Genesys Gplus Adapters, which provide out-of-the-box, pre-packaged, and vendor-validated solutions that integrate Genesys’ Customer Experience Platform to the leading CRM solutions. It's included as part of the Web Services and Applications installation package.

The adapter provides a single integrated agent desktop experience that offers rich data integration for screen pops and dispositioning. It presents complete customer information at a glance to more effectively serve customers in Salesforce Classic or Salesforce Console. The adapter leverages the Web Services API and the Salesforce Open CTI API.

There are two options available for the adapter:

- Gplus Adapter for Salesforce provides a smaller interface designed for Salesforce Classic, although it also works in Salesforce Console.
- Gplus Adapter for Salesforce - WWE Option provides the full Workspace Web Edition interface and is available in Salesforce Console and Salesforce Lightning.

Deploying Gplus Adapter for Salesforce

Gplus Adapter for Salesforce is available in either Salesforce Classic and Salesforce Console. It provides voice and chat functionality, along with Salesforce-specific features such as updating activity history, screen pops, and click-to-dial.

Deployment Tasks

Complete the following tasks to install and configure Gplus Adapter for Salesforce in your Genesys environment and in Salesforce.

1. Install and configure Web Services.
    a. Make sure you set up SSL for Jetty — the adapter won't work without it. To set up SSL, configure the SSL section of the **application.yaml** file as follows:

    ```
    enableSsl: true
    ssl:
        port: 8043
        keyStorePath: bec.jks
        keyStorePassword: OBF:1g3p1kqt1xtl19q51ni31nlv19q91xtx1ku11fzt
        keyManagerPassword: OBF:1g3p1kqt1xtl19q51ni31nlv19q91xtx1ku11fzt
        trustStorePath: bec.jks
        trustStorePassword: OBF:1g3p1kqt1xtl19q51ni31nlv19q91xtx1ku11fzt
    ```

    For more information about configuring SSL, see Configure SSL.

    b. If you want the adapter to use single sign-on, make sure you Configure SAML in Web Services. (You'll also need to add a special parameter to the **CTI Adapter URL** field in Step 4 of Configuring the adapter in Salesforce.)
2. Install and configure the adapter in Salesforce.
3. Refer to the Web Services and Applications Configuration Guide for information about how to configure the adapter in your Genesys environment.
4. Refer to the Gplus Adapters User Guide for information about how to work with the adapter.

Installing and configuring the adapter in Salesforce

Complete the procedures on this page to install and configure the Gplus Adapter for Salesforce in your Salesforce environment.

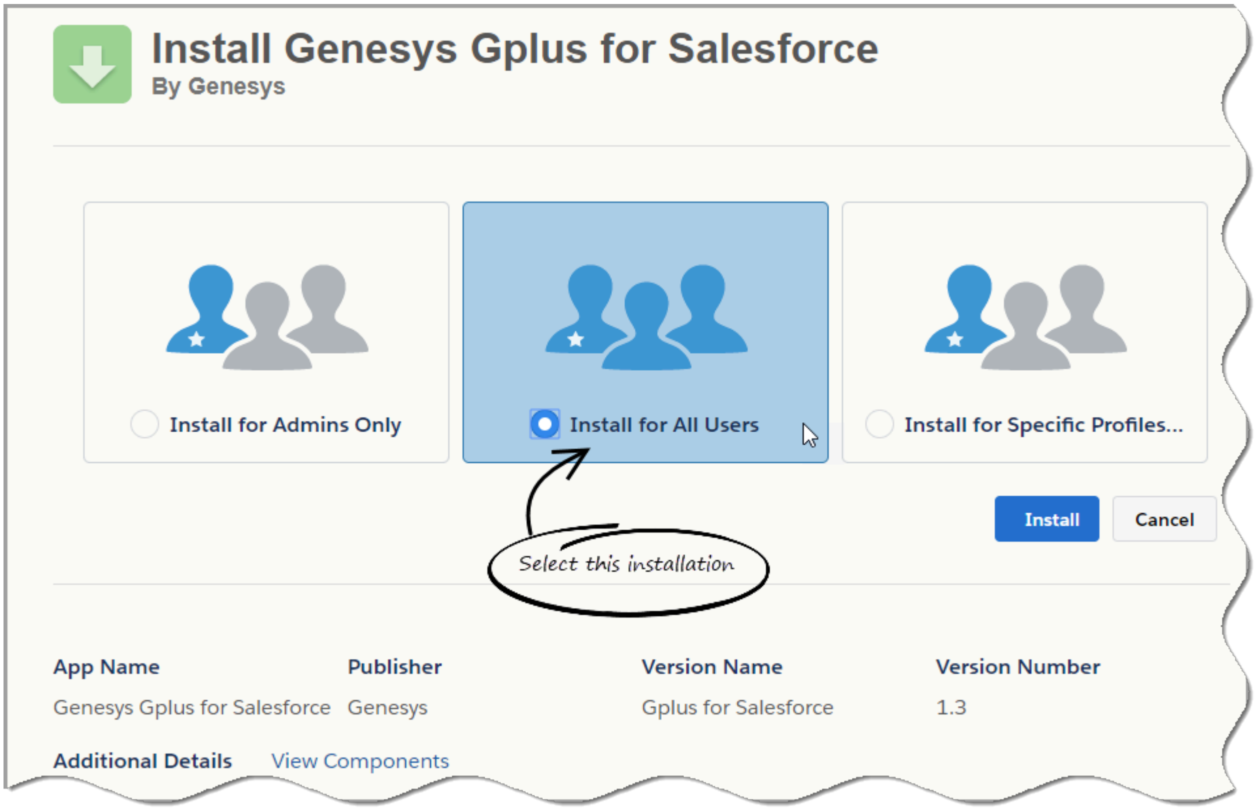
Installing the adapter in Salesforce

Prerequisites

- You have installed and configured Web Services.
- You have set up SSL for Jetty. For more information, see Configure SSL.

Start

1. Open the following URL to install the latest Gplus Adapter for Salesforce package in Salesforce:
   https://login.salesforce.com/packaging/installPackage.apexp?p0=04to0000000C3VD
   If you're not logged in, Salesforce prompts for your username and password.
2. Now you should see the **Install Genesys Gplus Adapter for Salesforce** page. Select an installation type. Generally, you should select **Grant access to all users**, but if you want to limit access to the adapter to specific profiles, then you can choose **Install for Specific Profiles ...**. Click **Install**.



Install Genesys Gplus for Salesforce
By Genesys

○ Install for Admins Only
● Install for All Users
○ Install for Specific Profiles...

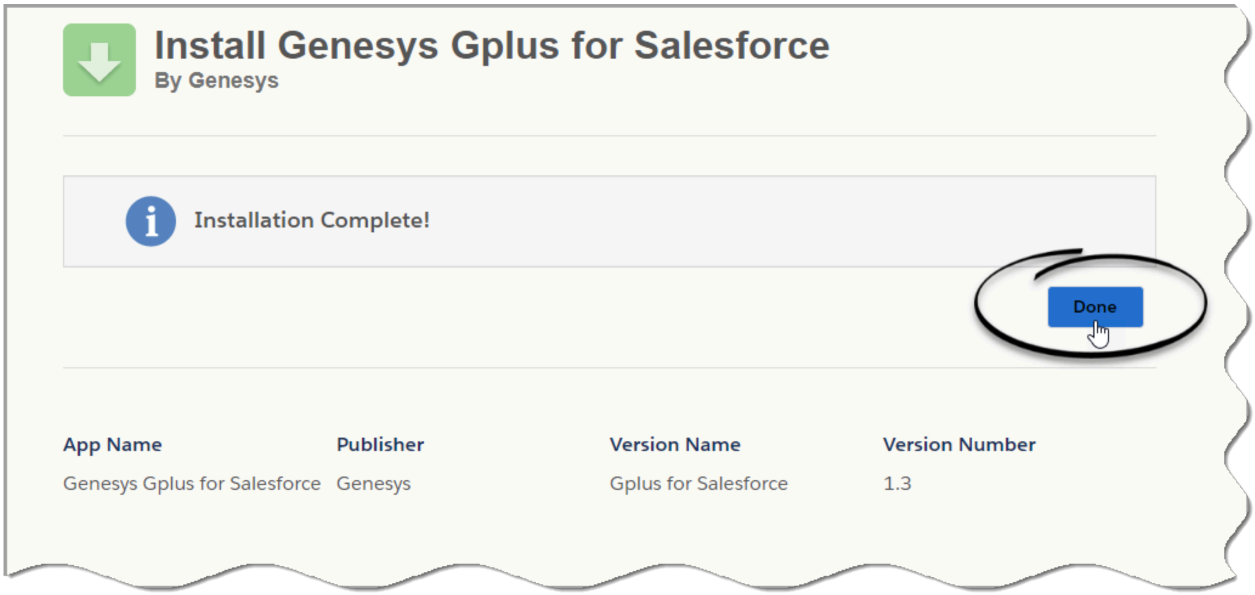
Install Cancel

Select this installation

App Name: Genesys Gplus for Salesforce
Publisher: Genesys
Version Name: Gplus for Salesforce
Version Number: 1.3

Additional Details View Components

3. When you see the "Installation Complete!" message, click **Done**.



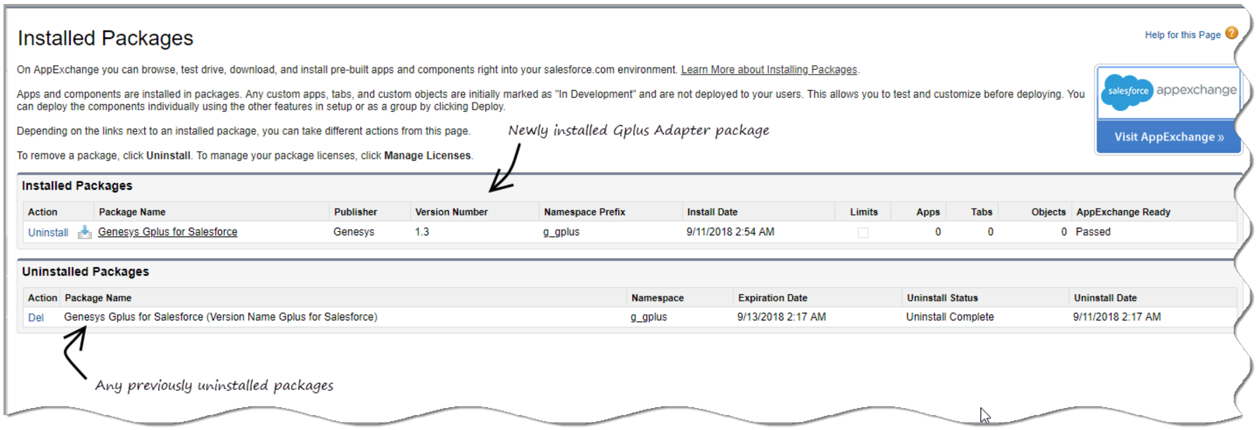
Install Genesys Gplus for Salesforce
By Genesys

Installation Complete!

Done

App Name: Genesys Gplus for Salesforce
Publisher: Genesys
Version Name: Gplus for Salesforce
Version Number: 1.3

You should be redirected to the **Installed Packages** page, with "Genesys Gplus for Salesforce" included in the list.



Installed Packages

Help for this Page

On AppExchange you can browse, test drive, download, and install pre-built apps and components right into your salesforce.com environment. Learn More about Installing Packages.

Apps and components are installed in packages. Any custom apps, tabs, and custom objects are initially marked as "In Development" and are not deployed to your users. This allows you to test and customize before deploying. You can deploy the components individually using the other features in setup or as a group by clicking Deploy.

Depending on the links next to an installed package, you can take different actions from this page.

To remove a package, click **Uninstall**. To manage your package licenses, click **Manage Licenses**.

Newly installed Gplus Adapter package

Installed Packages

Action	Package Name	Publisher	Version Number	Namespace Prefix	Install Date	Limits	Apps	Tabs	Objects	AppExchange Ready
Uninstall	Genesys Gplus for Salesforce	Genesys	1.3	g_gplus	9/11/2018 2:54 AM	□	0	0	0	Passed

Uninstalled Packages

Action	Package Name	Namespace	Expiration Date	Uninstall Status	Uninstall Date
Del	Genesys Gplus for Salesforce (Version Name Gplus for Salesforce)	g_gplus	9/13/2018 2:17 AM	Uninstall Complete	9/11/2018 2:17 AM

Any previously uninstalled packages

End

Configuring the adapter in Salesforce

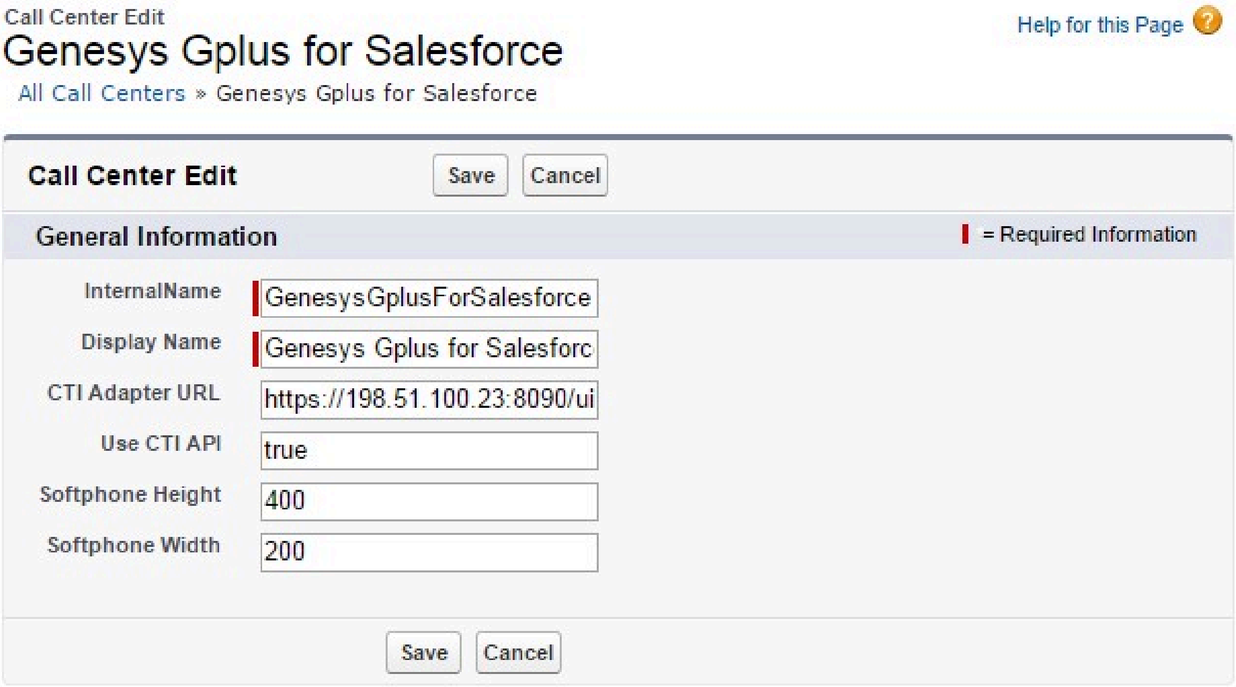
Complete this procedure to define your call center in Salesforce. The call center was created when you installed the Gplus Adapter for Salesforce package.

Start

1. If you haven't already, login to Salesforce and go to **Setup** > **Build** > **Customize** > **Call Center** > **Call Centers**. Or, you can search for "Call Centers" in the **Search All Setup** field and select the "Call Centers" result. You should see the **Introducing Salesforce CRM Call Center** page. **Note:** You must have administrator privileges.
2. You can select **Don't show me this page again** if you want to hide the page in the future, and click **Continue**.
3. On the **All Call Centers** page, click **Edit** next to the Genesys Gplus for Salesforce entry.
4. In the **CTI Adapter URL** field, replace `GWS_HOST:GWS_PORT` with the correct host and port for your installation of Web Services. For example: `https://198.51.100.23:8090/ui/crm-adapter/index.html?crm=salesforce`

   If you're enabling single sign-on in the adapter, add the `authType=saml` parameter to the **CTI Adapter URL**. For example: `https://198.51.100.23:8090/ui/crm-adapter/index.html?crm=salesforce&authType=saml`

   You should leave the other options at their default values so the adapter works correctly in Salesforce.



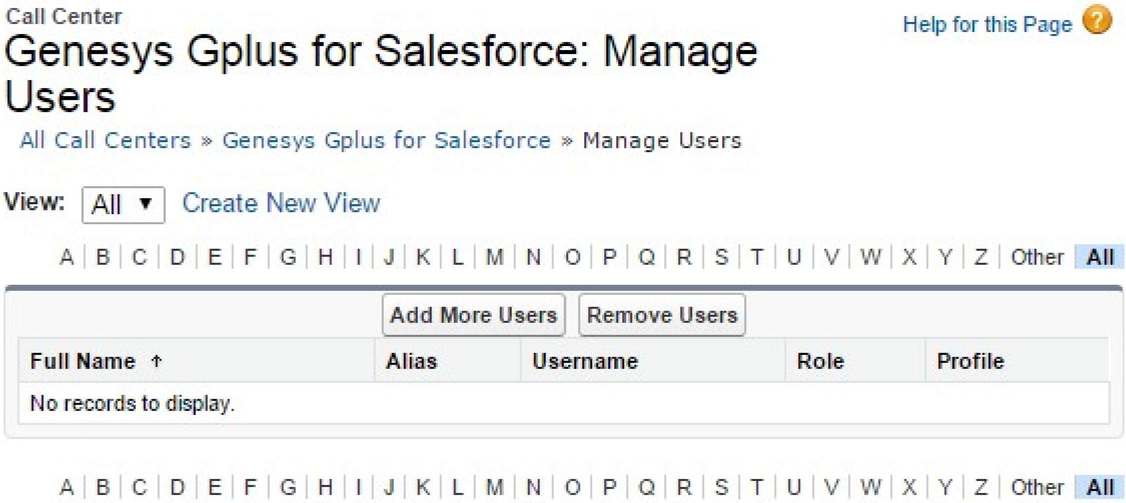
Call Center Edit

Genesys Gplus for Salesforce

All Call Centers » Genesys Gplus for Salesforce

InternalName	GenesysGplusForSalesforce
Display Name	Genesys Gplus for Salesforc
CTI Adapter URL	https://198.51.100.23:8090/ui
Use CTI API	true
Softphone Height	400
Softphone Width	200

5. Click **Save**.
6. Click **Manage Call Center Users** and then click **Add users**.



Call Center

Genesys Gplus for Salesforce: Manage Users

All Call Centers » Genesys Gplus for Salesforce » Manage Users

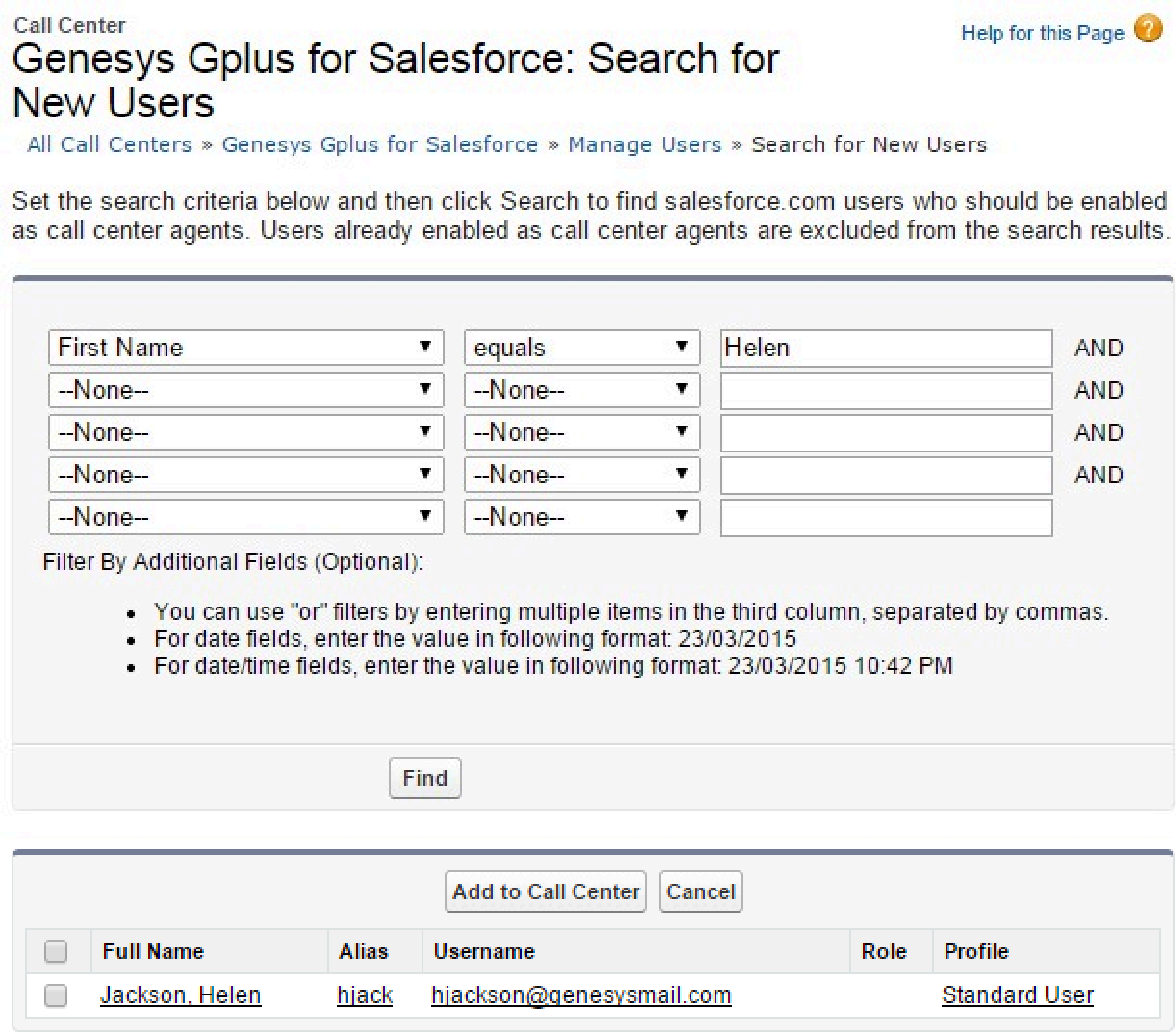
View: All

Help for this Page
Create New View
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All
Add More Users
Remove Users

Full Name ↑	Alias	Username	Role	Profile
No records to display.				

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

7. On the **Search for New Users** page, you can enter search criteria to find users. Select the ones you want to be able to use the adapter and click **Add to Call Center**.



Call Center

Genesys Gplus for Salesforce: Search for New Users

All Call Centers » Genesys Gplus for Salesforce » Manage Users » Search for New Users

Set the search criteria below and then click Search to find salesforce.com users who should be enabled as call center agents. Users already enabled as call center agents are excluded from the search results.

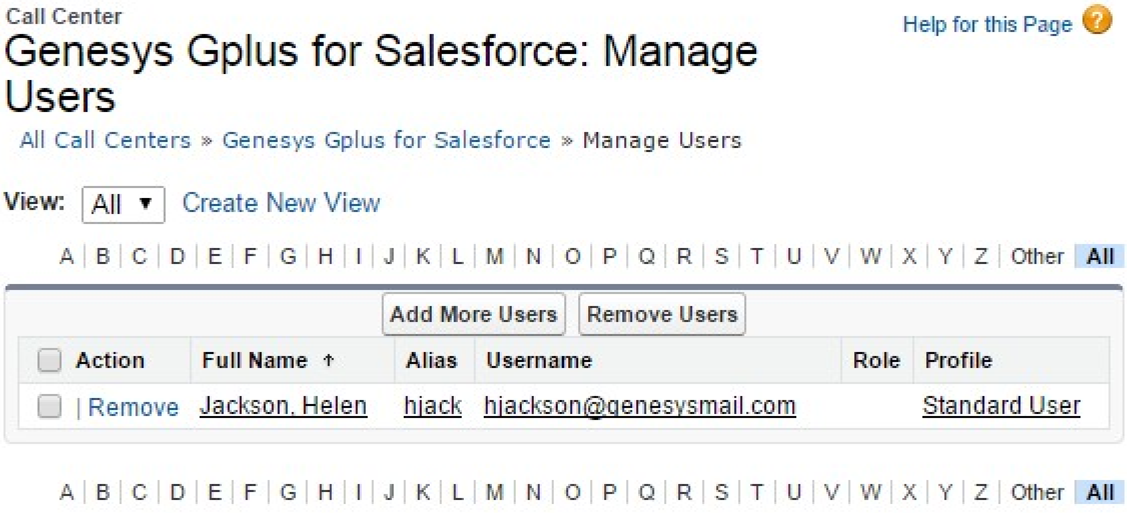
First Name	equals	Helen	AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		

Filter By Additional Fields (Optional):

- You can use "or" filters by entering multiple items in the third column, separated by commas.
- For date fields, enter the value in following format: 23/03/2015
- For date/time fields, enter the value in following format: 23/03/2015 10:42 PM

□	Full Name	Alias	Username	Role	Profile
□	Jackson, Helen	hjack	hjackson@genesysmail.com		Standard User

Your selected users are added to the list. You can remove a user on this page at any time.



Call Center

Genesys Gplus for Salesforce: Manage Users

Help for this Page

All Call Centers » Genesys Gplus for Salesforce » Manage Users

View: All ▾ Create New View

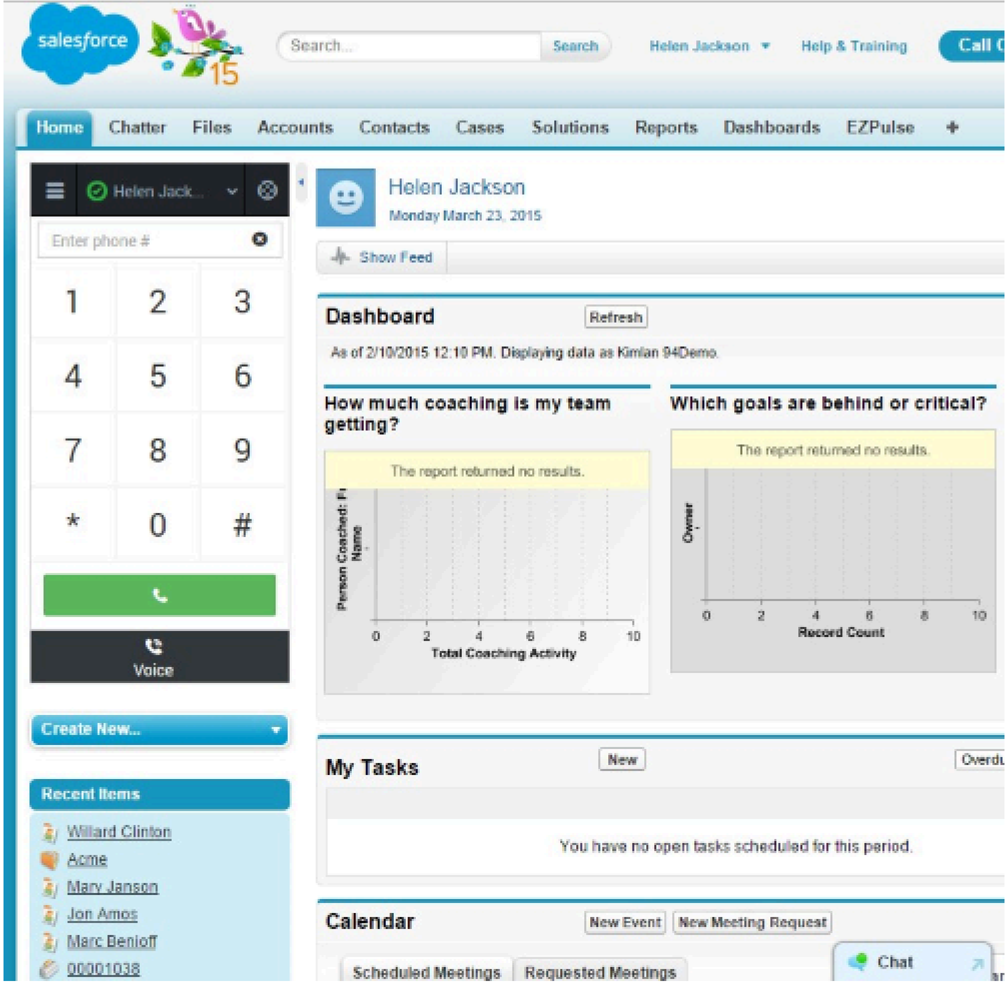
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

Add More Users | Remove Users

□ Action	Full Name ↑	Alias	Username	Role	Profile
□ | Remove	Jackson, Helen	hjack	hjackson@genesysmail.com		Standard User

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

8. To access the adapter in Salesforce Classic, look for it in the left pane of your browser; in Salesforce Console, click the **Phone** button in the bottom right corner.



Helen Jackson

Monday March 23, 2015

Dashboard

As of 2/10/2015 12:10 PM. Displaying data as Kimlan 94Demo.

How much coaching is my team getting?

The report returned no results.

Which goals are behind or critical?

The report returned no results.

My Tasks

You have no open tasks scheduled for this period.

Calendar

Recent Items

Willard Clinton

Acme

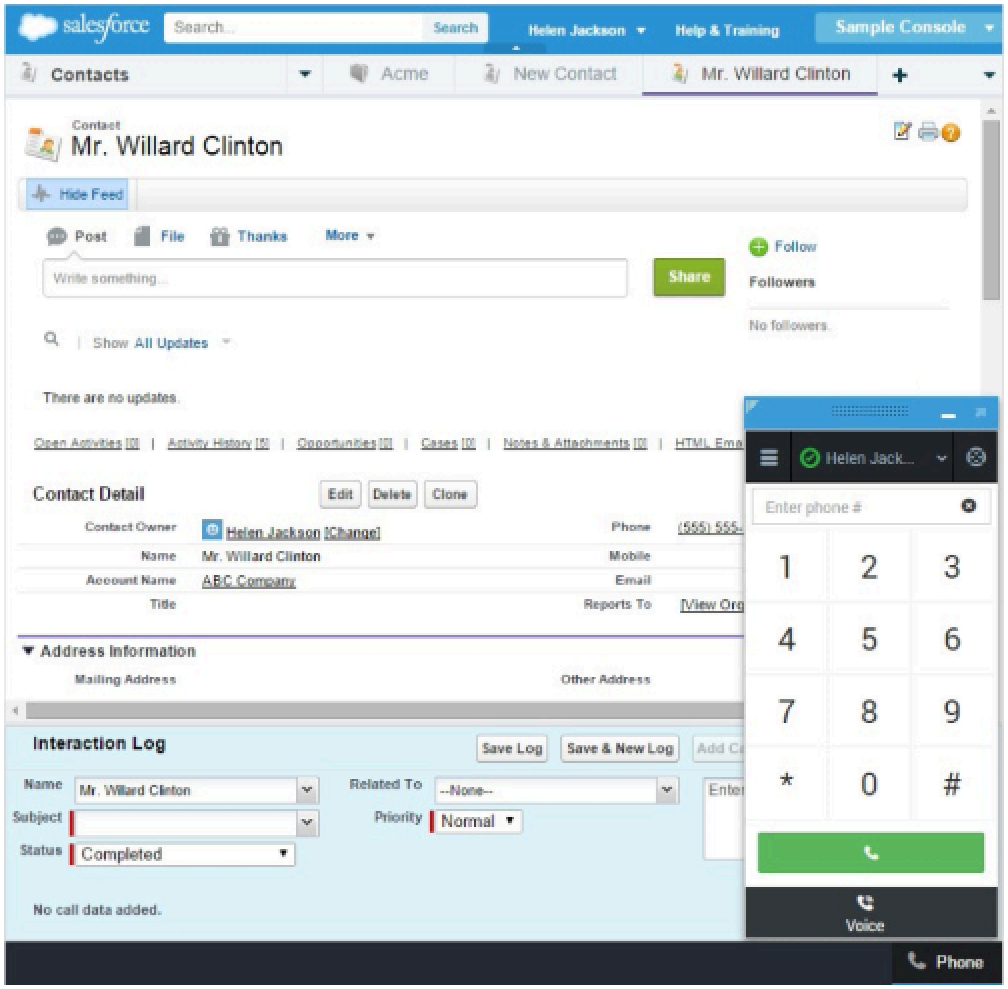
Mary Janson

Jon Amos

Marc Benioff

00001038

The adapter in Salesforce Classic.



Contact
Mr. Willard Clinton

Contact Detail

Contact Owner	Helen Jackson [Change]	Phone	(555) 555-
Name	Mr. Willard Clinton	Mobile	
Account Name	ABC Company	Email	
Title		Reports To	[View Org

Interaction Log

Name	Mr. Willard Clinton	Related To	--None--
Subject		Priority	Normal
Status	Completed		

No call data added.

The adapter in Salesforce Console.

End

Configuring screen pops in Salesforce

When an agent receives an external call, the adapter can initiate a screen pop that causes Salesforce to show an appropriate record for the caller. To set up this functionality in Salesforce, login and go to **Setup > Customize > Call Center > SoftPhone Layouts** to create a SoftPhone Layout. Check out the Salesforce documentation for details about configuration.

In general, there are a couple of things to consider when you set up a SoftPhone Layout for the adapter:

- The Gplus Adapter for Salesforce ignores the SoftPhone Layout settings that control call-related fields. Instead, the adapter gets this information from Toast and case data you configure in the Genesys environment.

- Make sure you configure the **Screen Pop Settings** in the "CTI 2.0 or Higher Settings" section. These settings control whether the screen pop opens in a new window, tab, or Visualforce page.

See Screen pop for more information about configuring screen pops in your Genesys environment.

Deploying Gplus Adapter for Salesforce - WWE Option

The WWE option of Gplus Adapter for Salesforce is available in Salesforce Console and Salesforce Lightning. It provides Salesforce-specific features such as updating activity history, screen pop, and click-to-dial, along with the full Workspace Web Edition user interface and the following features:

- Voice
- Chat
- Email
- Outbound Preview
- Voice and Chat Supervision (monitoring, coaching, barge-in)

> **Important**
>
> The Gplus Adapter URL in Salesforce Call Center follows this format: `https://<your company name>.genesyscloud.com/ui/crm-workspace/index.html`

Deployment tasks

Complete the following tasks to install and configure the adapter for the WWE option in your Genesys environment and in Salesforce.

1. Install and configure Web Services.
   a. Make sure you set up SSL for Jetty — the adapter won't work without it. For more information about configuring SSL, see Configure SSL.
   b. If you want the adapter to use single sign-on, make sure you Configure SAML in Web Services. (You'll also need to add a special parameter to the **CTI Adapter URL** field in Step 4 of Configuring the adapter in Salesforce.)

   > **Important**
   >
   > The Gplus Adapter URL in Salesforce Call Center follows this format: `https://<your company name>.genesyscloud.com/ui/crm-workspace/index.html`

2. Install and configure the adapter in Salesforce.
3. Refer to the Web Services and Applications Configuration Guide for information about how to configure

Workspace Web Edition and configure the adapter in your Genesys environment. The following Workspace Web Edition features are supported if you're using the WWE option:

- Voice
- Chat
- Email
- Outbound Preview
- Voice and Chat Supervision (monitoring, coaching, barge-in)

Don't forget to test and confirm that your Workspace Web Edition configuration is valid and your required features are enabled.

4. Refer to the Gplus Adapters User Guide for information about how to work with the adapter.

Installing and configuring the adapter in Salesforce - WWE option

If you're using the WWE option, complete the procedures on this page to install and configure the adapter in your Salesforce environment.

If you want to enable Gplus Adapter in Salesforce Lightning after you install and configure the adapter in Salesforce, go here.

Installing the adapter in Salesforce

Prerequisites

- You have installed and configured Web Services.
- You have set up SSL for Jetty. For more information, see Configure SSL.

The adapter installation procedure involves the following steps:

1. Creating a Gplus Adapter URL.
2. Configuring Gplus Adapter for Salesforce - Workspace Web Edition (WWE).
3. Adding users to Call Center.
4. Configuring the Utility bar.
5. Configuring the whitelist domain for your Salesforce Console.
6. Configuring screen pops in Salesforce.
7. Accessing Gplus Adapter for Salesforce - WWE Option.

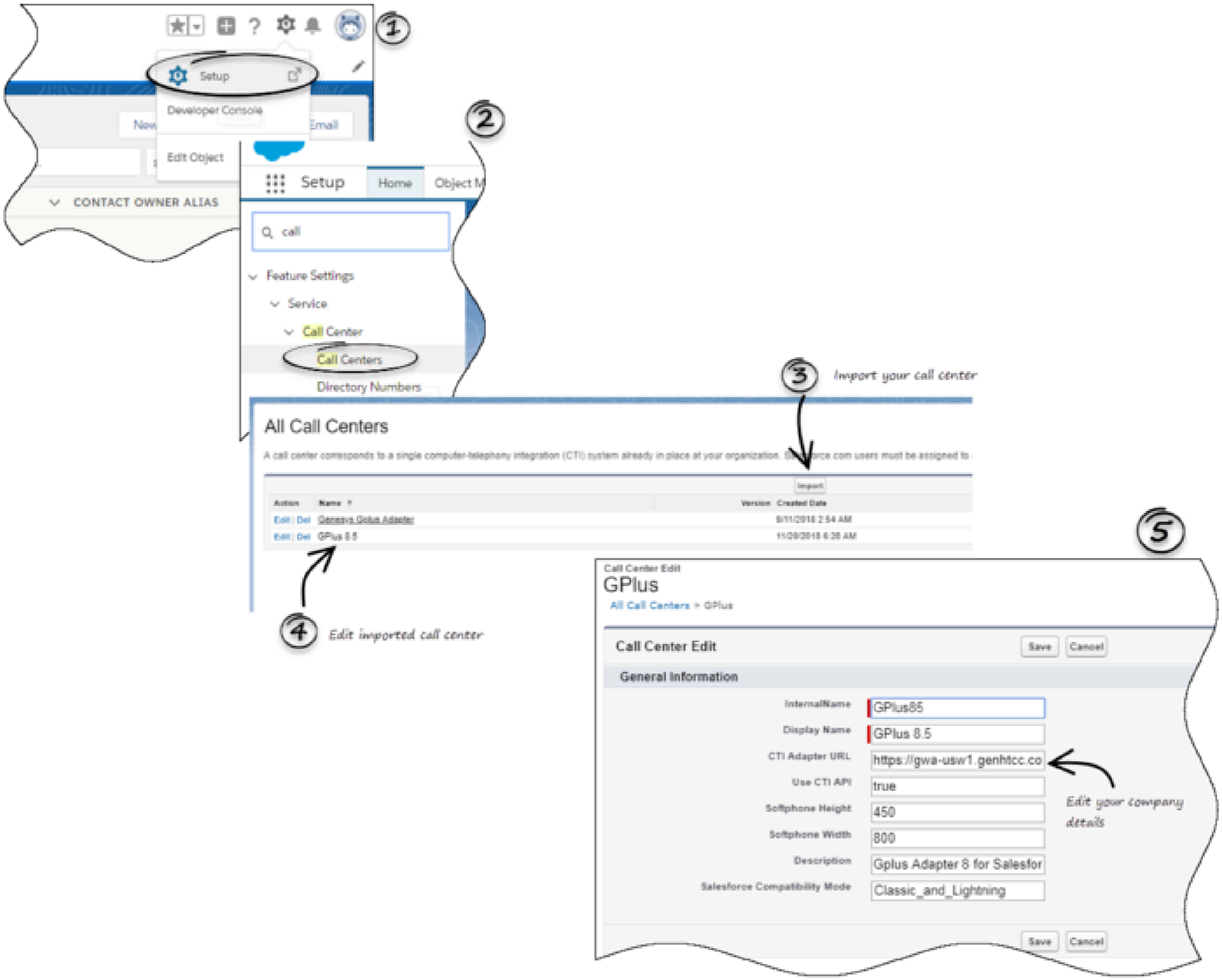
Creating a Gplus Adapter URL

From the Agent Workspace URL, copy the domain name and substitute it in the following URL:

```
https://<domain-name>/ui/crm-workspace/index.html?crm=salesforce
```

Configuring Gplus Adapter for Salesforce - Workspace Web Edition (WWE)

Follow these steps:



All Call Centers

A call center corresponds to a single computer-telephony integration (CTI) system already in place at your organization. Salesforce.com users must be assigned to

Import

Action	Name ↑	Version	Created Date
Edit | Del	Genesys Gplus Adapter		9/11/2018 2:54 AM
Edit | Del	GPlus 8.5		11/29/2018 6:28 AM

Call Center Edit
GPlus
All Call Centers » GPlus

Call Center Edit Save Cancel

General Information

InternalName	GPlus85
Display Name	GPlus 8.5
CTI Adapter URL	https://gwa-usw1.genhtcc.co
Use CTI API	true
Softphone Height	450
Softphone Width	800
Description	Gplus Adapter 8 for Salesfor
Salesforce Compatibility Mode	Classic_and_Lightning

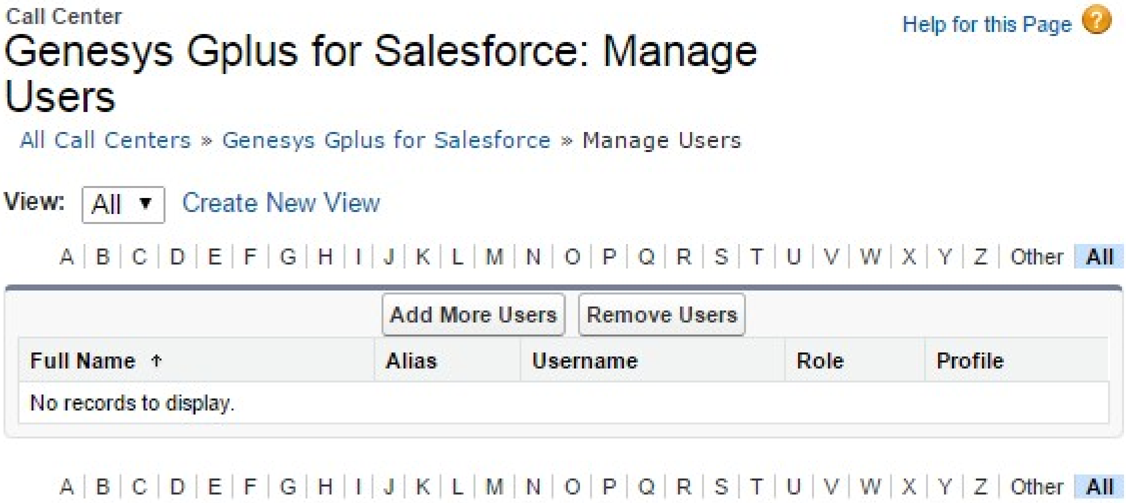
Save Cancel

1. Click the gear icon in the top right corner and then click **Setup**.
2. Using the **Quick Find** field, search for and access the **Call Centers setup** page.
3. From the **Call Center** settings page, using the **Import** functionality, import the crm-workspace-callcenter.xml file from your computer. If you have not already downloaded the file, download it from here.
4. From the **All Call Centers** list, click the **Call Center** you just imported. For example, Gplus85.
5. From the **Call Center Edit** page, in the **CTI Adapter URL** field, specify the Adapter URL that you constructed in the **Creating a Gplus Adapter URL** section. For example, https://www.genesysgplustest.com/ui/crm-workspace/index.html?crm=salesforce
6. Set **Salesforce Compatibility mode** to **Classic_and_Lightning**.
7. Save the changes.

Adding users to Call Center

The next step after you setup your Adapter is to add users to your call center. You must add at least one user to your call center.

1. In the **Call Centers setup** page, click **Manage Call Center Users**.
2. Click **Add More Users**.



Call Center

Genesys Gplus for Salesforce: Manage Users

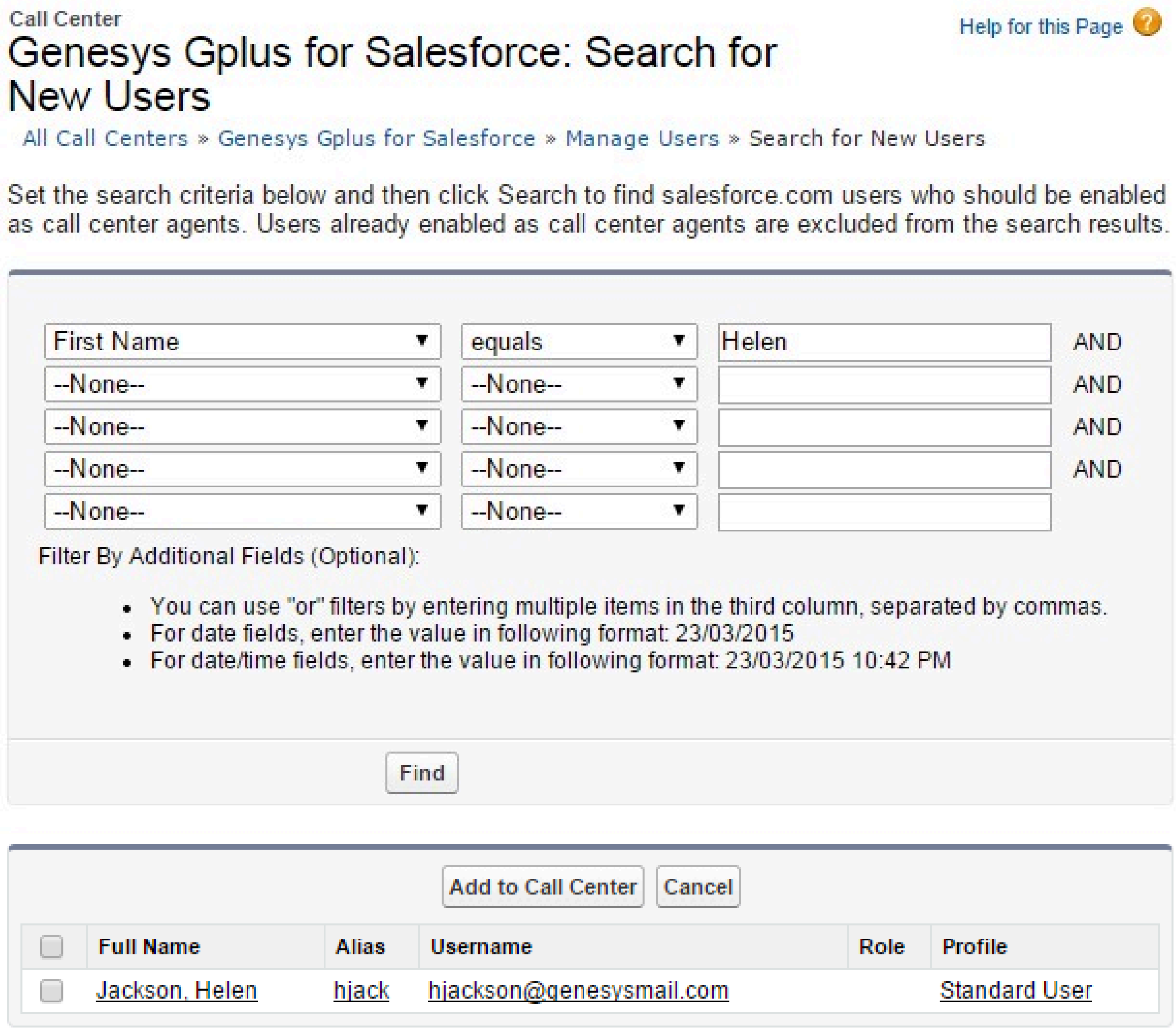
All Call Centers » Genesys Gplus for Salesforce » Manage Users

View: All

Full Name ↑	Alias	Username	Role	Profile
No records to display.				

Help for this Page
Create New View
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other All
Add More Users
Remove Users
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other All

3. On the **Search for New Users** page, you can enter search criteria to find users. Select the ones you want to add to the Gplus Adapter for Salesforce - Workspace Web Edition (WWE) option.
4. Click **Add to Call Center**.



Call Center

Genesys Gplus for Salesforce: Search for New Users

All Call Centers » Genesys Gplus for Salesforce » Manage Users » Search for New Users

Set the search criteria below and then click Search to find salesforce.com users who should be enabled as call center agents. Users already enabled as call center agents are excluded from the search results.

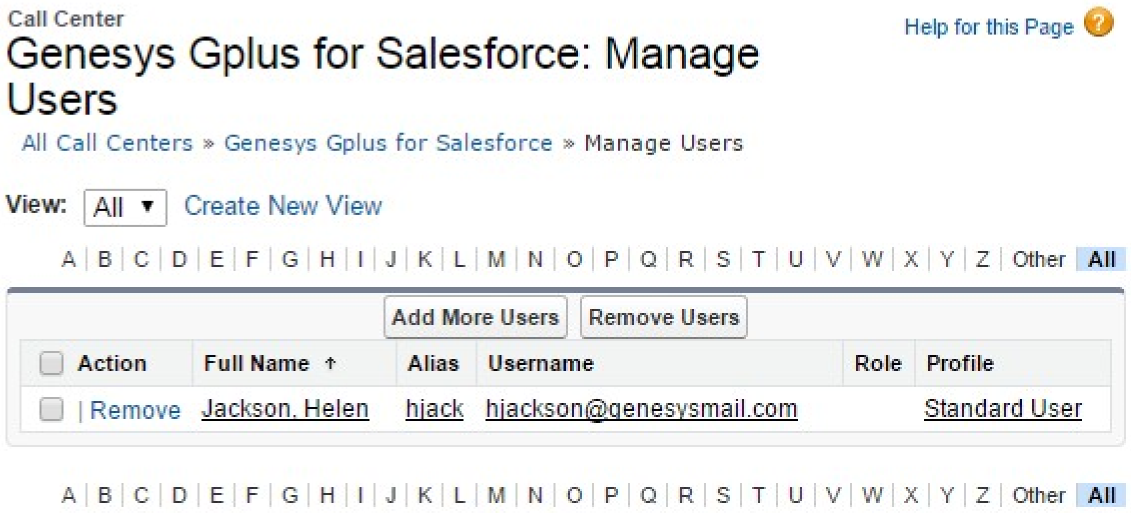
First Name	equals	Helen	AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		

Filter By Additional Fields (Optional):

- You can use "or" filters by entering multiple items in the third column, separated by commas.
- For date fields, enter the value in following format: 23/03/2015
- For date/time fields, enter the value in following format: 23/03/2015 10:42 PM

□	Full Name	Alias	Username	Role	Profile
□	Jackson, Helen	hjack	hjackson@genesysmail.com		Standard User

5. Your selected users are added to the list. You can remove a user on this page at any time.

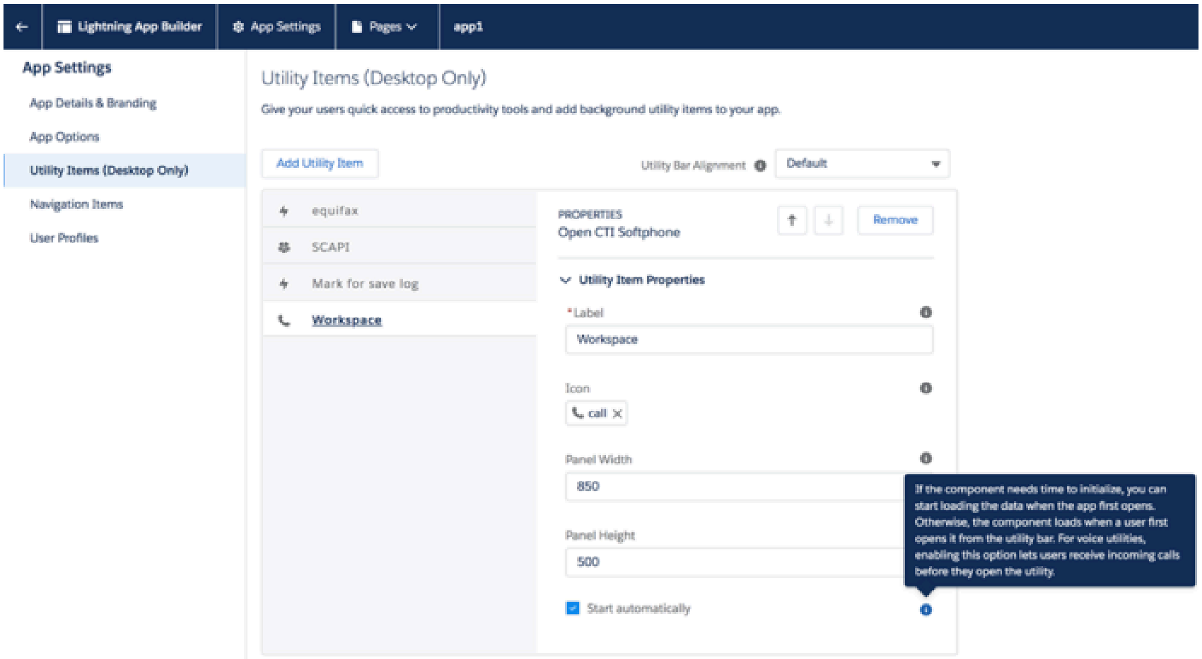

Call Center
Genesys Gplus for Salesforce: Manage Users
Help for this Page
All Call Centers » Genesys Gplus for Salesforce » Manage Users
View: All
Create New View
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other All
Add More Users
Remove Users

□ Action	Full Name ↑	Alias	Username	Role	Profile
□ | Remove	Jackson, Helen	hjack	hjackson@genesysmail.com		Standard User

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other All

Configuring the Utility bar

1. In the setup page, using the **Quick Find** field, search for and access the **App Manager settings** page.
2. Create a new application by clicking **New Lightning App**. Follow the steps in the Lightning App creation procedure in Salesforce documentation.
3. When creating a new Lightning app, follow the additional steps below to configure the utility items for your Gplus Adapter for Salesforce - Workspace Web Edition (WWE).
    - In the **App Options** step, ensure that you select **Console Navigation**. Gplus Adapter for Salesforce - Workspace Web Edition (WWE) does not support Standard Navigation.
    - In the **Utility Items** step, add **Open CTI Softphone** by clicking **Add Utility Item**.
    - For the soft phone, you can modify the display properties such as **Label**, **Panel Width**, and **Panel Height**.
    - Select **Start automatically** to start loading the data when the Gplus Adapter first opens to ensure that it initializes immediately; otherwise, Gplus Adapter does not load until an agent first opens it from the **Utility Bar**. If Gplus Adapter for Salesforce - Workspace Web Edition (WWE) is not initialized immediately, then agents do not receive voice calls until they first open the application.



Utility Items (Desktop Only)

Give your users quick access to productivity tools and add background utility items to your app.

Utility Bar Alignment: Default

PROPERTIES
Open CTI Softphone

Utility Item Properties

*Label: Workspace

Icon: call

Panel Width: 850

Panel Height: 500

☑ Start automatically

If the component needs time to initialize, you can start loading the data when the app first opens. Otherwise, the component loads when a user first opens it from the utility bar. For voice utilities, enabling this option lets users receive incoming calls before they open the utility.

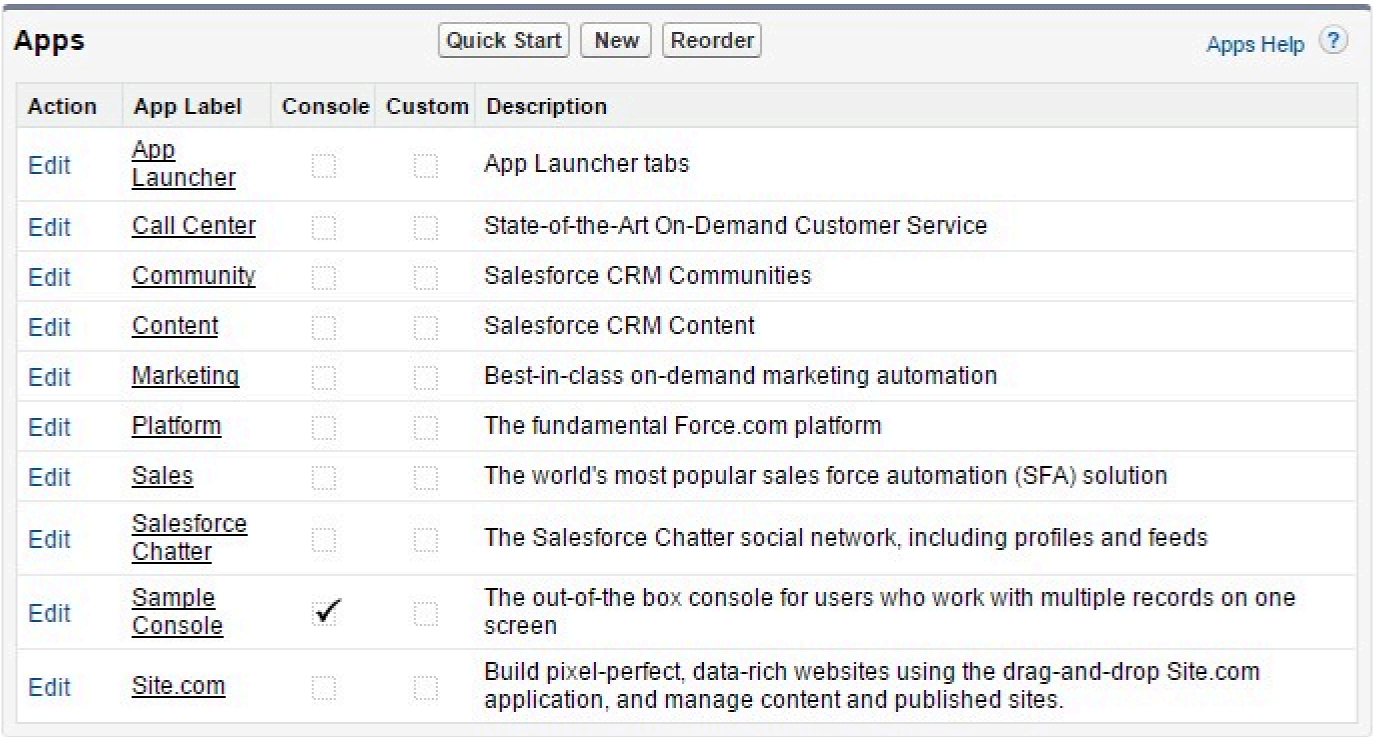
4. Specify **Navigation Items** and **User Profiles** for the new Lightning app by referring to Salesforce documentation.
5. Save the application.

Configuring the whitelist domain for your Salesforce Console

Complete this procedure to add the Genesys domain to the whitelist domains for your Salesforce Console. You need to complete this procedure to allow your users to access the adapter in Salesforce Console in a separate browser window.

Start

1. If you haven't already, login to Salesforce and go to **App Setup > Create > Apps** and select your console app — "Sample Console" in the image below:



Action	App Label	Console	Custom	Description
Edit	App Launcher	□	□	App Launcher tabs
Edit	Call Center	□	□	State-of-the-Art On-Demand Customer Service
Edit	Community	□	□	Salesforce CRM Communities
Edit	Content	□	□	Salesforce CRM Content
Edit	Marketing	□	□	Best-in-class on-demand marketing automation
Edit	Platform	□	□	The fundamental Force.com platform
Edit	Sales	□	□	The world's most popular sales force automation (SFA) solution
Edit	Salesforce Chatter	□	□	The Salesforce Chatter social network, including profiles and feeds
Edit	Sample Console	✓	□	The out-of-the box console for users who work with multiple records on one screen
Edit	Site.com	□	□	Build pixel-perfect, data-rich websites using the drag-and-drop Site.com application, and manage content and published sites.

2. Click **Edit**. In **Whitelist Domains**, add the host and port for your installation of Web Services. For example: `198.51.100.23:8090`
3. Click **Save**.

End

Configuring screen pops in Salesforce

When an agent receives an external call, the adapter can initiate a screen pop that causes Salesforce to show an appropriate record for the caller. To set up this functionality in Salesforce, login and go to **Setup > Customize > Call Center > SoftPhone Layouts** to create a SoftPhone Layout. Check out the Salesforce documentation for details about configuration.

In general, there are a couple of things to consider when you set up a SoftPhone Layout for the adapter:

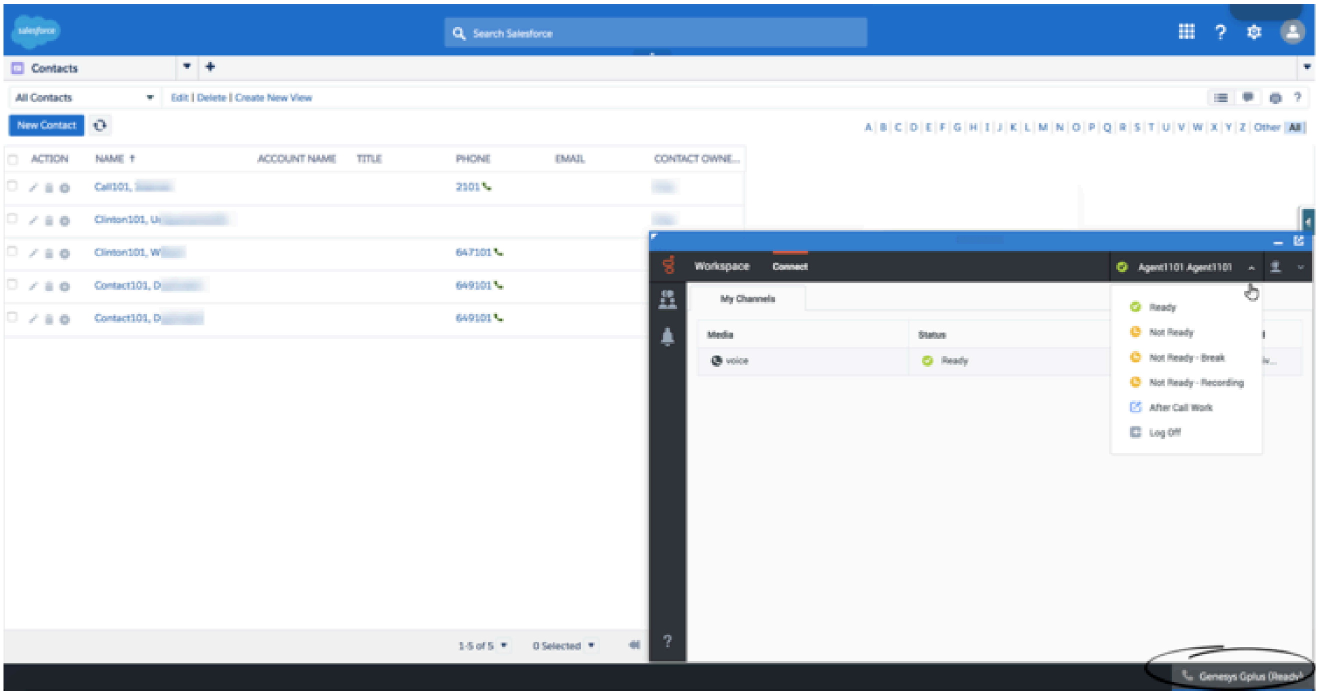
- The Gplus Adapter for Salesforce ignores the SoftPhone Layout settings that control call-related fields. Instead, the adapter gets this information from toast and case data you configure in the Genesys environment.
- Make sure you configure the **Screen Pop Settings** in the "CTI 2.0 or Higher Settings" section. These settings control whether the screen pop opens in a new window, tab, or Visualforce page.

See Screen pop for more information about configuring screen pops in your Genesys environment.

Accessing Gplus Adapter for Salesforce - WWE Option

Classic Mode

You can access Gplus Adapter in Salesforce Console by clicking the phone icon in the bottom-right corner.

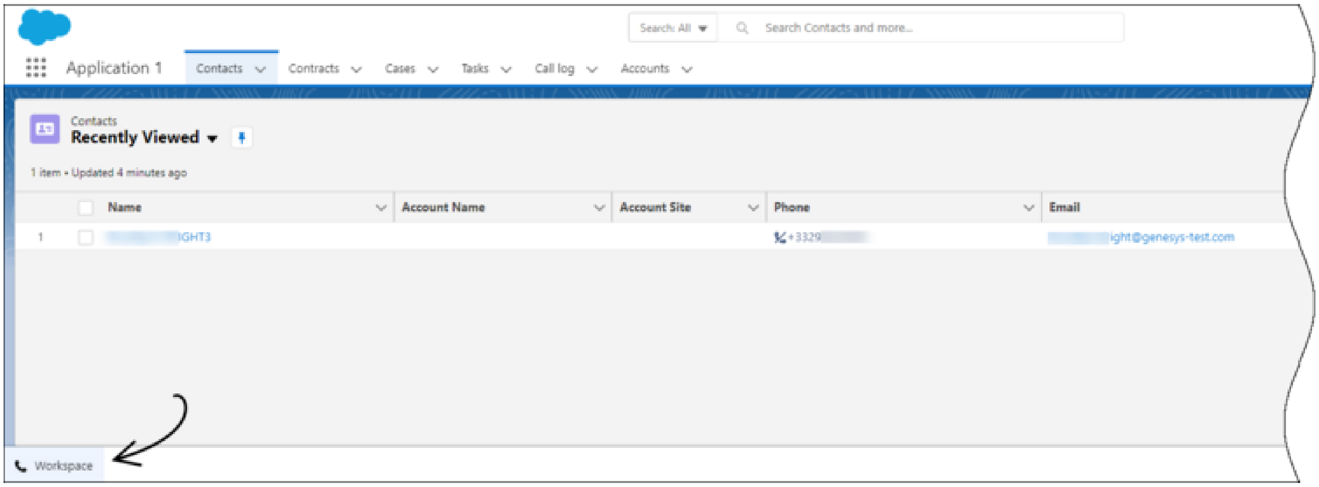


ACTION	NAME ↑	ACCOUNT NAME	TITLE	PHONE	EMAIL	CONTACT OWNE...
	Call101, [redacted]			2101		[redacted]
	Clinton101, U[redacted]					[redacted]
	Clinton101, W[redacted]			647101		
	Contact101, D[redacted]			649101		
	Contact101, D[redacted]			649101		

Media	Status
voice	Ready

Lightning Mode

You can access Gplus Adapter in Salesforce Lightning by clicking the phone icon in the bottom-left corner.



	Name	Account Name	Account Site	Phone	Email
1	[redacted]GHT3			+3329[redacted]	[redacted]ight@genesys-test.com

Enabling Lightning Experience

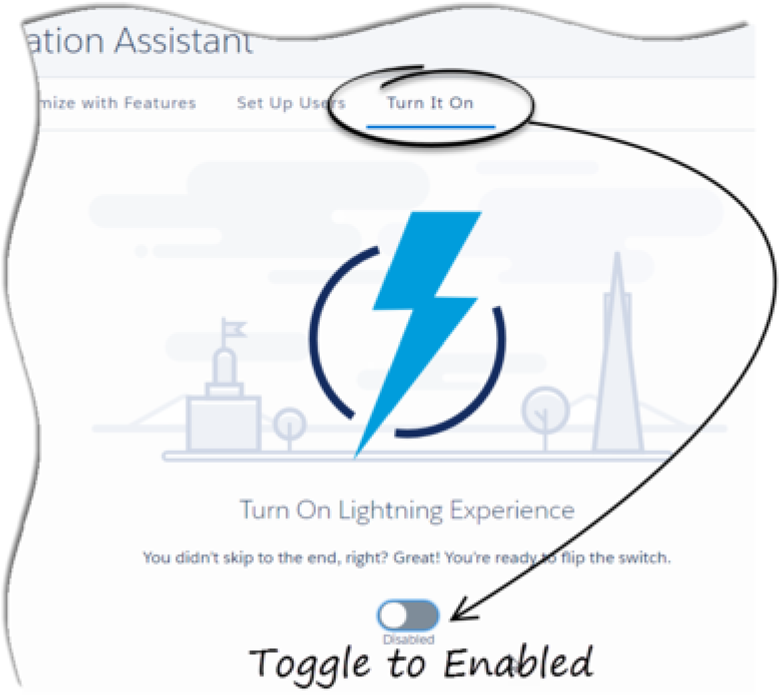
If you're using the WWE option, complete the procedures on this page to enable, set up, and access Lightning in your Salesforce environment.

> **Important**
>
> The Gplus Adapter URL in Salesforce Call Center follows this format: `https://<your company name>.genesyscloud.com/ui/crm-workspace/index.html?crm=lightning`

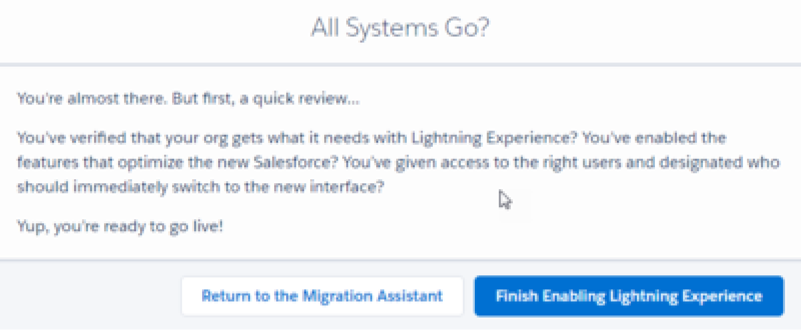
Enabling Lightning in Salesforce

Start

1. Log into the Salesforce environment.
2. From the **Setup** page, select **Lightning Experience** in the left-hand navigation bar. **Note:** If in Salesforce Classic mode, click the **Setup** menu and then the **Get Started** button found in the left-hand navigation bar.
3. In the **Lighting Experience** window, select **Turn It On**.
4. Move the toggle to the **Enabled** state.


ation Assistant
mize with Features
Set Up Users
Turn It On
Turn On Lightning Experience
You didn't skip to the end, right? Great! You're ready to flip the switch.
Disabled
Toggle to Enabled

5. A modal will pop up; click the **Finish Enabling Lightning Experience** button in the modal.



All Systems Go?

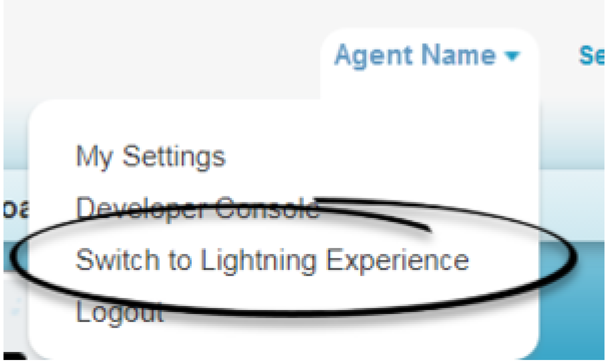
You're almost there. But first, a quick review...

You've verified that your org gets what it needs with Lightning Experience? You've enabled the features that optimize the new Salesforce? You've given access to the right users and designated who should immediately switch to the new interface?

Yup, you're ready to go live!

Return to the Migration Assistant
Finish Enabling Lightning Experience

6. In the dropdown labeled with the agent’s name at the top of the Salesforce classic view, click **Switch to Lightning Experience**.



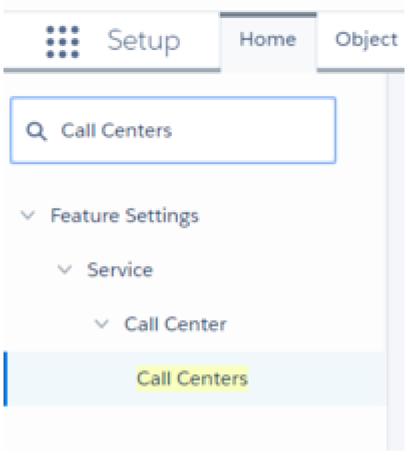
End

Setting Up The Adapter In Lightning

Prerequisite

Download the **lightning-callcenter.xml** file on your computer by right-clicking the link lightning-callcenter.xml and selecting the **Save link as...** option on the popup menu.

1. Go to the **Setup** page by clicking on the gear icon in the top right corner and clicking **Setup**.
2. Using the quick find field, search for and access the **Call Centers** settings page.

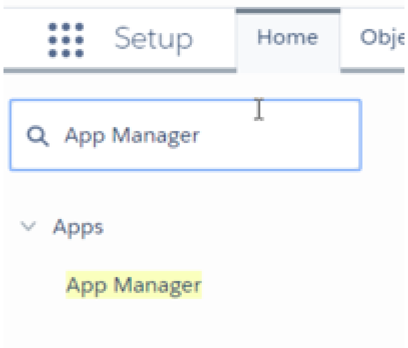


3. Click **Import**.
4. Click **Choose File**.
5. Select the **lightning-callcenter.xml** file downloaded on your computer. If you have not already downloaded the file, right-click the link lightning-callcenter.xml and select the **Save link as...** option to download.
6. Click **Import**.
7. From **All Call Centers** list, click the call center you just imported. For example, **GPlusLightning**.
8. Click **Edit**.
9. In the **CTI Adapter URL** field, replace 'GWSHOST' and 'GWSPORT' with the host and port details of the adapter in your environment. For example, an updated URL will look like this: `https://bec135-gws.live.genesys.com/ui/crm-workspace/index.html?crm=lightning`
   Note: If you are deploying the adapter with Single-Sign-On (SSO) capability, ensure that you add the `&authType=saml` parameter at the end of the CTI Adapter URL. For example, an updated URL with SSO capability will look like this: `https://bec135-gws.live.genesys.com/ui/crm-workspace/index.html?crm=lightning&authType=saml`
10. Click **Save**.
11. Click **Manage Call Center Users**.
12. Click **Add More Users**.
13. Search the interface to find the users you want to add to the Lightning adapter.

> ## Important
> A user cannot be added to both the Lightning and non-Lightning adapters

.

14. Select the users you want to add and click **Add to Call Center**.
15. Using the quick find field, access the **App Manager** settings page.



16. In the apps list, click the **Show more actions** drop down on the far right side of the adapter app you wish to use.
    Note: If you do not see any apps in the list, you can create one by clicking **New Lightning App**.
17. Click **Edit**.
18. Click **Utility Bar**.
19. From the Utility Bar window, click **Add** and select "Open CTI Softphone".
20. Change the **Label** field to "Workspace".
21. Click **Done**.

Accessing the Adapter

Start

1. In the top-left corner, click the **App Launcher** icon:
2. Select the app that you created when setting up the adapter.
3. Click **Workspace** from the bar at the bottom-left to open the adapter.
4. Log in to the Adapter.

End

Troubleshooting

This page provides solutions to common problems in Web Services.

The following log for Web Services is saved to the `/var/log/jetty` directory on the Web Services node:

- `cloud.log` — Stores WARN level messages about Web Services.

To modify the log message levels, you can edit the `$JETTY_HOME/resources/logback.xml` file and change the level to DEBUG or TRACE (instead of WARN):

```
<logger name="com.genesyslab" level="DEBUG" />
```

404 Error

Problem

You receive a 404 Error on a diagnostic API request. For example, `http://192.0.2.20:8080/api/v2/diagnostics/version`.

Resolution

The Web Services web application uses the Jetty root context. If other web applications served by the same instance of Jetty also use the root context, this can prevent the Web Services web application from getting routed requests. If you are working with a fresh install of Jetty, you should remove the default Jetty files from the `$JETTY_HOME/webapps` and `$JETTY_HOME/contexts` folders.

Retrieve Metrics

GWS offers a method to retrieve system data including Java Virtual Machine (JVM) information and other metrics using the API. This information is useful in troubleshooting and running diagnostics.

To get runtime information, you should be either a Contact Center administrator or an OPS user. Runtime information can be retrieved for a single node.

Request

```
GET ../api/v2/service/runtime
```

HTTP Response

```
{
    "information": {
        "jvmMetrics": {
            "gc": {
                "jvm.gc.PS-MarkSweep.count": {
                    "value": <int>
                },
                "jvm.gc.PS-MarkSweep.time": {
                    "value": <int>
                },
                "jvm.gc.PS-Scavenge.count": {
                    "value": <int>
                },
                "jvm.gc.PS-Scavenge.time": {
                    "value": <int>
                }
            },
            "memory": {
                "jvm.memory.heap.committed": {
                    "value": <int>
                },
                "jvm.memory.heap.init": {
                    "value": <int>
                },
                "jvm.memory.heap.max": {
                    "value": <int>
                },
                "jvm.memory.heap.usage": {
                    "value": <double>
                },
                "jvm.memory.heap.used": {
                    "value": <int>
                },
                "jvm.memory.non-heap.committed": {
                    "value": <int>
                },
                "jvm.memory.non-heap.init": {
                    "value": <int>
                },
                "jvm.memory.non-heap.max": {
                    "value": <int>
                },
                "jvm.memory.non-heap.usage": {
                    "value": <double>
                },
                "jvm.memory.non-heap.used": {
                    "value": <int>
                },
                "jvm.memory.pools.Code-Cache.usage": {
                    "value": <double>
                },
                "jvm.memory.pools.Compressed-Class-Space.usage": {
                    "value": <double>
                },
                "jvm.memory.pools.Metaspace.usage": {
                    "value": <double>
                },
                "jvm.memory.pools.PS-Eden-Space.usage": {
                    "value": <double>
                },
                "jvm.memory.pools.PS-Old-Gen.usage": {
                    "value": <double>
                },
                "jvm.memory.pools.PS-Survivor-Space.usage": {
                    "value": <double>
                },
                "jvm.memory.total.committed": {
                    "value": <int>
                },
                "jvm.memory.total.init": {
                    "value": <int>
                },
                "jvm.memory.total.max": {
                    "value": <int>
                },
                "jvm.memory.total.used": {
                    "value": <int>
                }
            },
            "threads": {
                "jvm.threads.blocked.count": {
                    "value": <int>
                },
                "jvm.threads.count": {
                    "value": <int>
                },
                "jvm.threads.daemon.count": {
                    "value": <int>
                },
                "jvm.threads.deadlock.count": {
                    "value": <int>
                },
                "jvm.threads.deadlocks": {
                    "value": []
                },
                "jvm.threads.new.count": {
                    "value": <int>
                },
                "jvm.threads.runnable.count": {
                    "value": <int>
                },
                "jvm.threads.terminated.count": {
                    "value": <int>
                },
                "jvm.threads.timed_waiting.count": {
                    "value": <int>
                },
                "jvm.threads.waiting.count": {
                    "value": <int>
                }
            }
        },
        "jvmParameters": {
            "inputArguments": [
                "-Xms4G",
                "-Xmx4G",
                ...
            ],
            "name": "...",
            "specName": "...",
            "specVendor": "...",
            "specVersion": "...",
            "startTime": <long>,
            "systemProperties": {
```

```
            "LOG_FILE": "...",
            "LOG_PATH": "...",
            "PID": "...",
            "archaius.configurationSource.additionalUrls": "...",
            "awt.toolkit": "...",
            "file.encoding": "...",
            "file.encoding.pkg": "...",
            "file.separator": "...",
            "gopherProxySet": "...",
            "java.awt.graphicsenv": "...",
            "java.awt.headless": "...",
            "java.awt.printerjob": "...",
            "java.class.version": "...",
            "java.endorsed.dirs": "...",
            "java.ext.dirs": "...",
            "java.home": "...",
            "java.io.tmpdir": "...",
            "java.library.path": "...",
            "java.runtime.name": "...",
            "java.runtime.version": "...",
            "java.specification.name": "...",
            "java.specification.vendor": "...",
            "java.specification.version": "...",
            "java.vendor": "...",
            "java.vendor.url": "...",
            "java.vendor.url.bug": "...",
            "java.version": "...",
            "java.vm.info": "...",
            "java.vm.name": "...",
            "java.vm.specification.name": "...",
            "java.vm.specification.vendor": "Oracle Corporation",
            "java.vm.specification.version": "...",
            "java.vm.vendor": "...",
            "java.vm.version": "...",
            "line.separator": "...",
            "org.apache.xml.security.ignoreLineBreaks": "...",
            "org.jboss.logging.provider": "...",
            "org.owasp.esapi.SecurityConfiguration": "...",
            "os.arch": "...",
            "os.name": "...",
            "os.version": "...",
            "path.separator": ":",
            "spring.beaninfo.ignore": "...",
            "sun.arch.data.model": "...",
            "sun.boot.library.path": "...",
            "sun.cpu.endian": "...",
            "sun.cpu.isalist": "...",
            "sun.io.unicode.encoding": "....",
            "sun.java.command": "...",
            "sun.java.launcher": "...",
            "sun.jnu.encoding": "...",
            "sun.management.compiler": "...",
            "sun.nio.ch.bugLevel": "...",
            "sun.os.patch.level": "...",
            "user.country": "...",
            "user.country.format": "...",
            "user.dir": "...",
            "user.home": "...",
            "user.language": "...",
            "user.name": "...",
            "user.timezone": "..."
        },
        "uptime": <long>,
            "vmName": "...",
            "vmVendor": "...",
            "vmVersion": "..."
        }
    },
    "statusCode": 0
}
```

Appendix: How-to Create SSL Certificate

Prerequisites

- Create the root pair (rootCA key & rootCA cert).
- Prepare the `mkdir /root/ca` directory.
- Create the directory structure:

  ```
  # cd /root/ca
   # mkdir certs crl newcerts private
   # chmod 700 private
   # touch index.txt
   # echo 1000 > serial
  ```

- Copy the root CA configuration (openssl.cnf) to /root/ca/openssl.cnf
- Create the root key:

  ```
  # cd /root/ca
   # openssl genrsa -aes256 -out private/<rootCA>.key.pem 4096
  ```

- Enter pass phrase for `<rootCA>.key.pem`: <Enter password>
- Verifying - Enter pass phrase for `<rootCA>.key.pem`: <Enter password>

```
# chmod 400 private/<rootCA>.key.pem
```

Create Root Certificate

- Use the `<rootCA>.key.pem` root key to create the `<rootCA>.cert.pem` root certificate.

  ```
  # cd /root/ca
   # openssl req -config openssl.cnf -key private/<rootCA>.key.pem -new -x509 -days 7300
   -sha256 -extensions v3_ca -out certs/<rootCA>.cert.pem
  ```

- Enter the pass phrase for `<rootCA>.key.pem`: <password for "rootCA.key.pem">

  ```
  You are about to be asked to enter information that will be incorporated
   into your certificate request.
   -----
   Country Name (2 letter code) [XX]: <Enter country code>
   State or Province Name []: <Enter state or province>
   Locality Name []: <Enter city>
   Organization Name []: <Enter company name>
   Organizational Unit Name []: <Enter company OU>
   Common Name []: <Enter some value>
   Email Address []: <Enter admin mail account>
  ```

```
# chmod 444 certs/<rootCA>.cert.pem
```

Verify Root Certificate

```
# cd /<rootCA>.cert.pem
```

The output shows:

- The `Signature Algorithm` used
- The dates of certificate `Validity`
- The `Public-Key` bit length
- The `Issuer`, which is the entity that signed the certificate
- The `Subject`, which refers to the certificate itself

The `Issuer` and `Subject` are identical as the certificate is self-signed. Note that all root certificates are self-signed.

```
Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=GB, ST=England,
            O=Alice Ltd, OU=Alice Ltd Certificate Authority,
            CN=Alice Ltd Root CA
    Validity
        Not Before: Apr 11 12:22:58 2015 GMT
        Not After : Apr  6 12:22:58 2035 GMT
    Subject: C=GB, ST=England,
             O=Alice Ltd, OU=Alice Ltd Certificate Authority,
             CN=Alice Ltd Root CA
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
            Public-Key: (4096 bit)
```

The output also shows the `X509v3 extensions`. We applied the v3_ca extension, so the options from `[ v3_ca ]` should be reflected in the output.

```
X509v3 extensions:
    X509v3 Subject Key Identifier:
        38:58:29:2F:6B:57:79:4F:39:FD:32:35:60:74:92:60:6E:E8:2A:31
    X509v3 Authority Key Identifier:
        keyid:38:58:29:2F:6B:57:79:4F:39:FD:32:35:60:74:92:60:6E:E8:2A:31

    X509v3 Basic Constraints: critical
        CA:TRUE
    X509v3 Key Usage: critical
        Digital Signature, Certificate Sign, CRL Sign
```

GWS Key and Certificate Generation

- Make a directory for GWS files:

  ```
  # cd /root/ca # mkdir gwsCerts
  ```

- Create a key:

  ```
  # cd /root/ca
  ```

```
# openssl genrsa -aes256 -out gwsCerts/<gwsKey>.key.pem 2048
# chmod 400 gwsCerts/<gwsKey>.key.pem
```

- Create a certificate (CSR):
  Requirement
  the Common Name must be a fully qualified domain name.

  Copy `san.cnf` and `v3.ext` to `/root/ca` and modify the following parameters in these files

  ```
  commonName = <Enter FQDN of your GWS host>
   DNS.1   = commonName
   DNS.2   = *.<part of FQDN>
  ```

```
# cd /root/ca
# openssl req -out gwsCerts/<gwsCSR>.csr -newkey rsa:2048 -nodes -keyout
gwsCerts/<gwsKey>.key.pem -config san.cnf
```

- **Enter pass phrase for <gwsKey>.key.pem**
  <password for gws Key>

  You are about to be asked to enter information that will be incorporated into your certificate request.

  ```
  Country Name (2 letter code) [XX]: <Enter country code>
   State or Province Name []: <Enter state>
   Locality Name []: <Enter city>
   Organization Name []: <Enter company>
   Organizational Unit Name []: <Enter company OU>
   Common Name []: <Enter FQDN of your GWS host>
   Email Address []: <Enter email address>
  ```

- Sign the GWS CSR file:
  Use rootCA authority to sign up GWS csr file.

  ```
  # cd /root/ca
   # openssl x509 -req -sha256 -days 367 -in gwsCerts/<gwsCSR>.csr -CA <full path to
   ‘rootCA.cert.pem’> -CAkey <full path to ‘rootCA.key.pem’> -CAcreateserial -out
   gwsCerts/<gwsSignedCert>.pem -extfile v3.ext -extensions v3_req
   # chmod 444 gwsCerts/<gwsSignedCert>.cert.pem
  ```

  Example:

  ```
  openssl x509 -req -sha256 -days 367 -in gwsCerts/gwsCSR.csr -CA /root/ca/certs/
   rootCA.cert.pem -CAkey /root/ca/private/rootCA.key.pem -CAcreateserial -out gwsCerts/
   gwsSignedCert.pem -extfile v3.ext -extensions v3_req
   # chmod 444 gwsCerts/gwsSignedCert.pem
  ```

- Verify the certificate:

  ```
  # openssl x509 -noout -text -in gwsCerts/<gwsSignedCert>.pem
   Check for x509v3 extensions (SAN & v3 extensions).
  ```

Converting Procedures

- Convert the existing cert to a PKCS12 using OpenSSL.

> **Important**
>
> A password is required.

```
# cd /root/ca
 # openssl pkcs12 -export -in <gwsSignedCert>.pem -inkey <gwsKey>.key.pem -out
 <keystore.p12> -name <certAlias> -CAfile <full path to ‘rootCA.cert.pem’> -caname rootCA
```

Example:

```
# openssl pkcs12 -export -in /root/ca/gwsCerts/gwsSignedCert.pem -inkey /root/ca/gwsCerts/
 gwsKey.key.pem -out keystore.p12 -name firstcert -CAfile /root/ca/certs/rootCA.cert.pem
 -caname rootCA
```

- Convert the PKCS12 to a Java Keystore File.

```
# cd /root/ca
 # keytool -importkeystore -deststorepass <new_keystore_pass> -destkeypass <new_key_pass>
 -destkeystore <gwsKeystore.jks> -srckeystore <keystore.p12> -srcstoretype PKCS12
 -srcstorepass <pass_used_in_p12_keystore> -alias <alias_used_in_p12_keystore>
```

Example:

```
keytool -importkeystore -deststorepass password -destkeypass password -destkeystore
 gwsKeystore.jks -srckeystore keystore.p12 -srcstoretype PKCS12 -srcstorepass password
 -alias firstcert
 * System will automatically tell to change the format:<source lang="text">
 keytool -importkeystore -srckeystore gwsKeystore.jks -destkeystore gwsKeystore.jks
 -deststoretype pkcs12
```

Import Missing Certs and Create Truststore

- Import rootCa certificate to `gwsKeystore.jks`:
- Use keytool -importcert to import the rootCa certificate into each node keystore:

```
# cd /root/ca
 # keytool -importcert -keystore <gwsKeystore>.jks -alias rootCA -file <path to
 ‘rootCA.cert.pem’> -noprompt -keypass <keystore password> -storepass <password>
```

Example:

```
# keytool -importcert -keystore gwsKeystore.jks -alias rootCA -file /root/ca/certs/
 rootCA.cert.pem -noprompt -keypass password -storepass password
```

- Create a server truststore:

```
# cd /root/ca
 # keytool -importcert -keystore <gwsTruststore>.jks -alias rootCA -file
 <rootCA>.cert.pem -noprompt -keypass <key password> -storepass <password>
```

Example:

```
# keytool -importcert -keystore gwsTruststore.jks -alias rootCA -file /root/ca/certs/
 rootCA.cert.pem -noprompt -keypass password -storepass password
```

GWS Configuration (application.yaml)

- Configuration example (**jetty** section):

```
enableSsl: true
  ssl:
    port: 8443
    securePort: 443
    idleTimeout: 30000
    soLingerTime: -1
    trustAll: true
    keyStorePath: /root/ca/<gwsKeystore>.jks
    keyStorePassword: <keystore password>
    keyStoreType: JKS
    trustStorePath: /root/ca/<gwsTruststore>.jks
    trustStorePassword: <truststore password>
```

Example:

```
# caCertificate: /root/ca/myKeystore.jks
 # jksPassword: Manila@1234
 port: 8443
 securePort: 443
 idleTimeout: 30000
 soLingerTime: -1
 trustAll: true
 keyStorePath: /root/ca/gwsKeystore.jks
 keyStorePassword: password
 keyStoreType: JKS
 trustStorePath: /root/ca/gwsTruststore.jks
 trustStorePassword: password
```

On Client Desktop

- Add your host (hostname should be specified as FQDN) in `<%system_drive%>\Windows\System32\drivers\etc\hosts`.
Example in the file

```
192.168.100.26      gws-centos7.genesys.com
```

- Convert `<rootCA>.cert.pem` to PFX format:

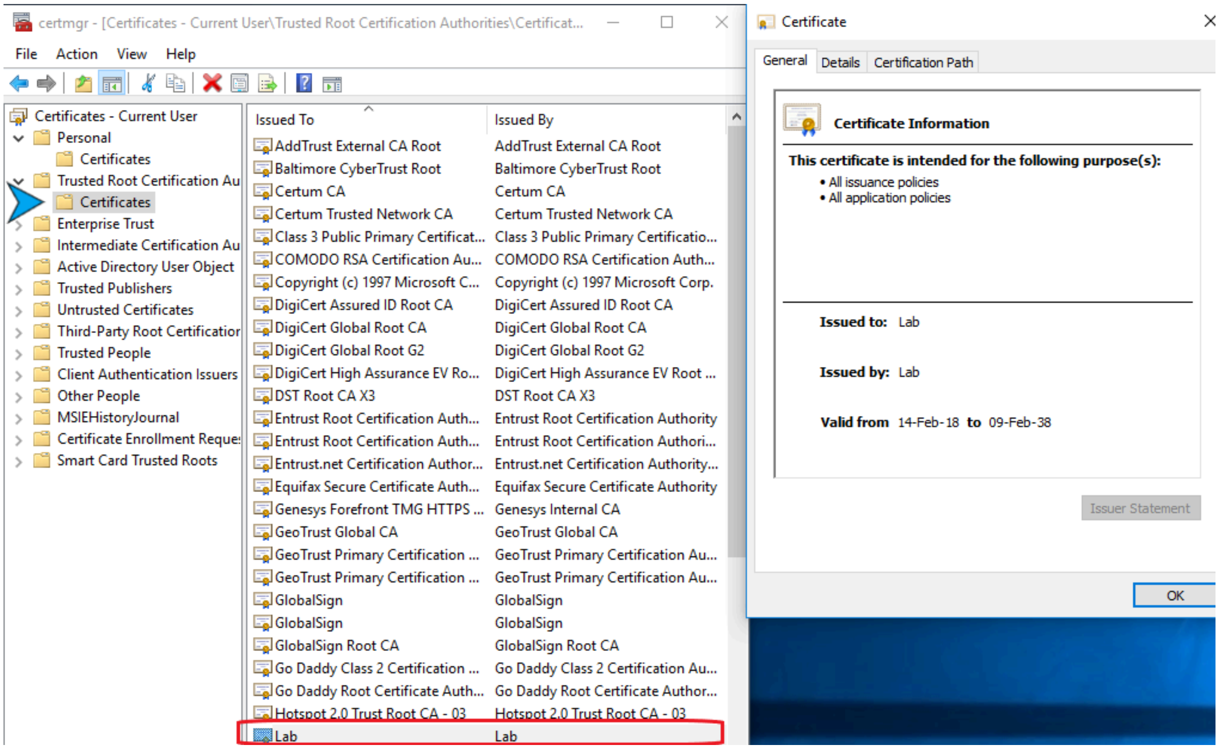
```
# cd /root/ca
 # openssl pkcs12 -inkey <rootCA>.key.pem -in <rootCA>.cert.pem -export -out <rootCA>.pfx
```

Example:

```
openssl pkcs12 -inkey /root/ca/private/rootCA.key.pem -in /root/ca/certs/rootCA.cert.pem
 -export -out rootCA.pfx
```

- Copy <rootCA>.pfx and <keystore>.p12 to Windows host.
- Import the `<keystore>.p12` file by double-clicking on it. Use default configuration and specify the password.
- Import the `<rootCA>.pfx` file, make sure to select “Place all certificates in the following store”. Browse “Trusted Root Certification Authorities”

- Verify that certificates are present using `certmgr.msc`.
  Example:
  - For rootCA certificate:


certmgr - [Certificates - Current User\Trusted Root Certification Authorities\Certificat...
File
Action
View
Help
Certificates - Current User
Personal
Certificates
Trusted Root Certification Au
Certificates
Enterprise Trust
Intermediate Certification Au
Active Directory User Object
Trusted Publishers
Untrusted Certificates
Third-Party Root Certificatior
Trusted People
Client Authentication Issuers
Other People
MSIEHistoryJournal
Certificate Enrollment Reque:
Smart Card Trusted Roots
Certificate
General
Details
Certification Path
Issuer Statement
OK

Issued To	Issued By
AddTrust External CA Root	AddTrust External CA Root
Baltimore CyberTrust Root	Baltimore CyberTrust Root
Certum CA	Certum CA
Certum Trusted Network CA	Certum Trusted Network CA
Class 3 Public Primary Certificat...	Class 3 Public Primary Certificatio...
COMODO RSA Certification Au...	COMODO RSA Certification Auth...
Copyright (c) 1997 Microsoft C...	Copyright (c) 1997 Microsoft Corp.
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA
DigiCert Global Root CA	DigiCert Global Root CA
DigiCert Global Root G2	DigiCert Global Root G2
DigiCert High Assurance EV Ro...	DigiCert High Assurance EV Root ...
DST Root CA X3	DST Root CA X3
Entrust Root Certification Auth...	Entrust Root Certification Authority
Entrust Root Certification Auth...	Entrust Root Certification Authori...
Entrust.net Certification Author...	Entrust.net Certification Authority...
Equifax Secure Certificate Auth...	Equifax Secure Certificate Authority
Genesys Forefront TMG HTTPS ...	Genesys Internal CA
GeoTrust Global CA	GeoTrust Global CA
GeoTrust Primary Certification ...	GeoTrust Primary Certification Au...
GeoTrust Primary Certification ...	GeoTrust Primary Certification Au...
GlobalSign	GlobalSign
GlobalSign	GlobalSign
GlobalSign Root CA	GlobalSign Root CA
Go Daddy Class 2 Certification ...	Go Daddy Class 2 Certification Au...
Go Daddy Root Certificate Auth...	Go Daddy Root Certificate Author...
Hotspot 2.0 Trust Root CA - 03	Hotspot 2.0 Trust Root CA - 03
Lab	Lab

Certificate Information

This certificate is intended for the following purpose(s):

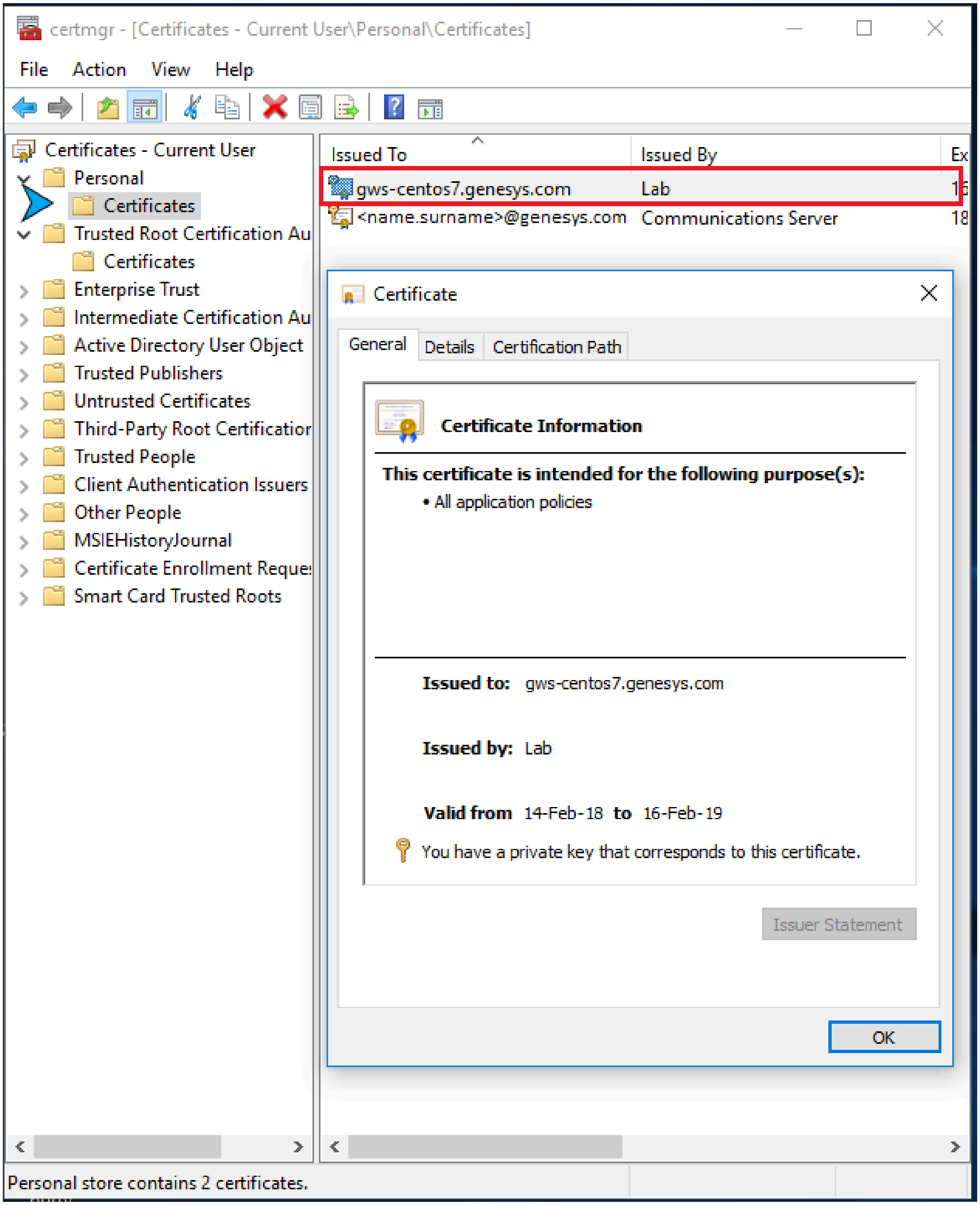
- All issuance policies
- All application policies

Issued to: Lab

Issued by: Lab

Valid from 14-Feb-18 **to** 09-Feb-38

  - For GWS certificate:


certmgr - [Certificates - Current User\Personal\Certificates]
File
Action
View
Help
Certificates - Current User
Personal
Certificates
Trusted Root Certification Au
Certificates
Enterprise Trust
Intermediate Certification Au
Active Directory User Object
Trusted Publishers
Untrusted Certificates
Third-Party Root Certificatior
Trusted People
Client Authentication Issuers
Other People
MSIEHistoryJournal
Certificate Enrollment Reque:
Smart Card Trusted Roots
Certificate
General
Details
Certification Path
Certificate Information
Issuer Statement
OK

Issued To	Issued By	Ex
gws-centos7.genesys.com	Lab	16
<name.surname>@genesys.com	Communications Server	18

This certificate is intended for the following purpose(s):

- All application policies

Issued to: gws-centos7.genesys.com

Issued by: Lab

Valid from 14-Feb-18 **to** 16-Feb-19

You have a private key that corresponds to this certificate.

Personal store contains 2 certificates.