



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.

## Agent Pacing Service Deployment Guide

# Table of Contents

<b>How Pacing Works</b>	
How Pacing works	4
<b>Architecture</b>	
Architecture	9
<b>Installation</b>	
Install the Pacing Service	12
<b>Provisioning</b>	
Provisioning	32
<b>Configuration</b>	
Configuration	37
<b>Troubleshooting</b>	
Troubleshooting	46

---

Learn the purpose of the Agent Pacing Service and the available documentation.

Agent Pacing Service combines statistical data from the contact center with statistics collected by Genesys Predictive Engagement to predict the availability of agent resources. It forwards the results to Genesys Predictive Engagement, so that Genesys Predictive Engagement can make an informed decision about whether to send a predictive engagement request to a specific visitor.

- How Pacing works
- Genesys Multicloud CX Prerequisites
- Provisioning
- Configuration
- Install the Pacing Service

# How Pacing works

## Contents

- 1 The pacing algorithm
  - 1.1 Virtual queue
  - 1.2 Agent Group
  - 1.3 Pacing target
- 2 Output parameter
- 3 High-level architecture

## How Pacing works

---

Understand the concept of pacing and learn how the Agent Pacing Service ensures agents are available to assist customers.

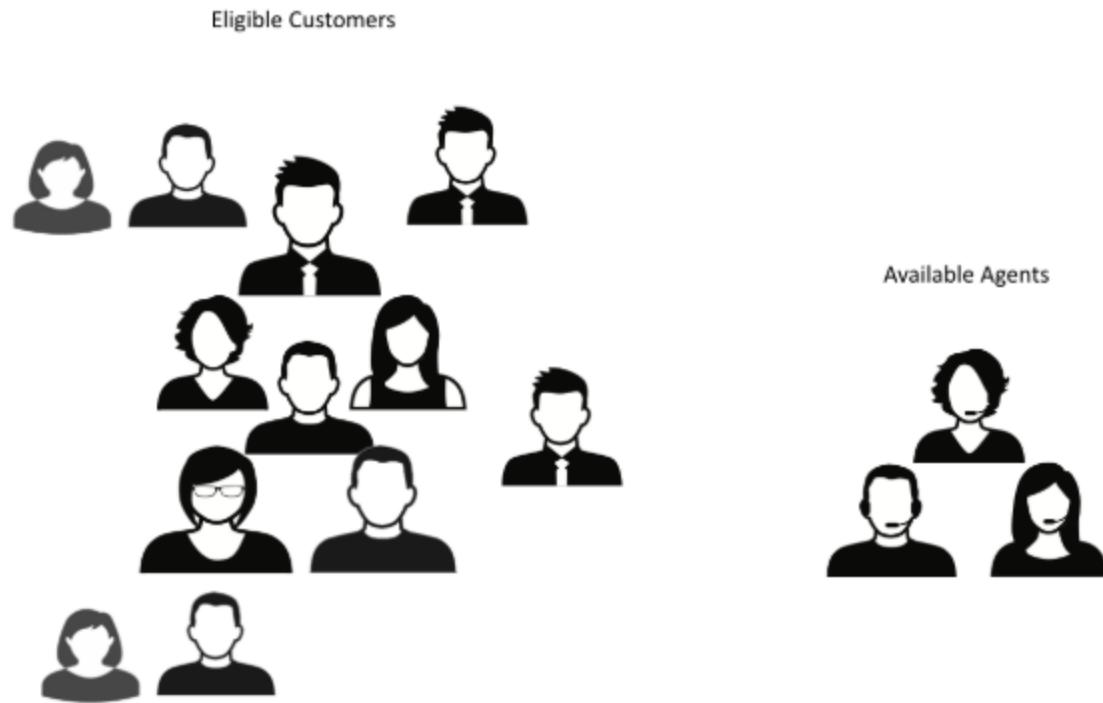
Many products allow you to focus on *reactive* engagement sessions, in which a customer initiates the connection with your contact center by responding to a *static* request, such as a **Chat Now** button. Genesys Predictive Engagement goes a step further, by predicting when it's a good time for your contact center to take the initiative—to reach out to a customer proactively. For example, Genesys Predictive Engagement might see that a customer is likely to abandon a purchase, so it proactively sends them a pop-up that says "Let's chat!" At that point, the customer can click an **Accept** button and your agents have an opportunity to help them complete their purchase.

Most of the information that Genesys Predictive Engagement works with is focused on your customers. This data helps you pinpoint the ones who are most likely to respond well to a proactive invitation.

This works great when you have enough agents:



But what if a customer accepts an engagement offer—and your agents are too busy to answer?



That's what pacing is for. A *pacing algorithm* helps you manage your agent load by comparing it to your traffic and predicting how many proactive invitations you can send out—without making your customers wait too long for a response from your agents.

## The pacing algorithm

Agent Pacing Service uses a pacing algorithm to balance two things:

- The need for your agents to stay busy
- The need to minimize the abandonment rate for proactive interactions

The algorithm uses the following inputs as the basis of its calculations.

## Virtual queue

Contact centers use queues of various kinds to hold interactions that are waiting to be processed. These queues can take the form of an automatic call distributor (ACD) on a hardware switch, a queue of digital media interactions supported by Genesys Interaction Server, or a router-based virtual queue. Virtual queues are great because you can easily use them with just about any media type—including voice, chat and email. Agent Pacing Service requires that you pass your interactions through virtual queues, and that you dedicate each queue to a single media type. It's important to keep this in mind when creating your routing strategies.

---

## Agent Group

The Agent Group defines which agents the pacing algorithm should include in its calculations when it predicts whether it's safe for Genesys Predictive Engagement to generate a proactive request. You can use a regular Agent Group, which consists of a pre-configured list of agents that you can only change explicitly, or a virtual Agent Group, whose agent list is defined dynamically based on a skill expression, such as Spanish > 5.

## Pacing target

The pacing target uses the output from the pacing algorithm (the number of proactive invitations that can be sent during a specific time interval) to decide whether to send an invitation to a specific visitor who has triggered a hot-lead event.

### Important

Genesys Predictive Engagement automatically propagates Agent Group and Virtual Queue information to Genesys Widgets, making it available as UserData to the Routing Logic for the selected engagement channel.

## Output parameter

The pacing algorithm produces only one output parameter: the number of proactive invites that can be sent during a specific time interval.

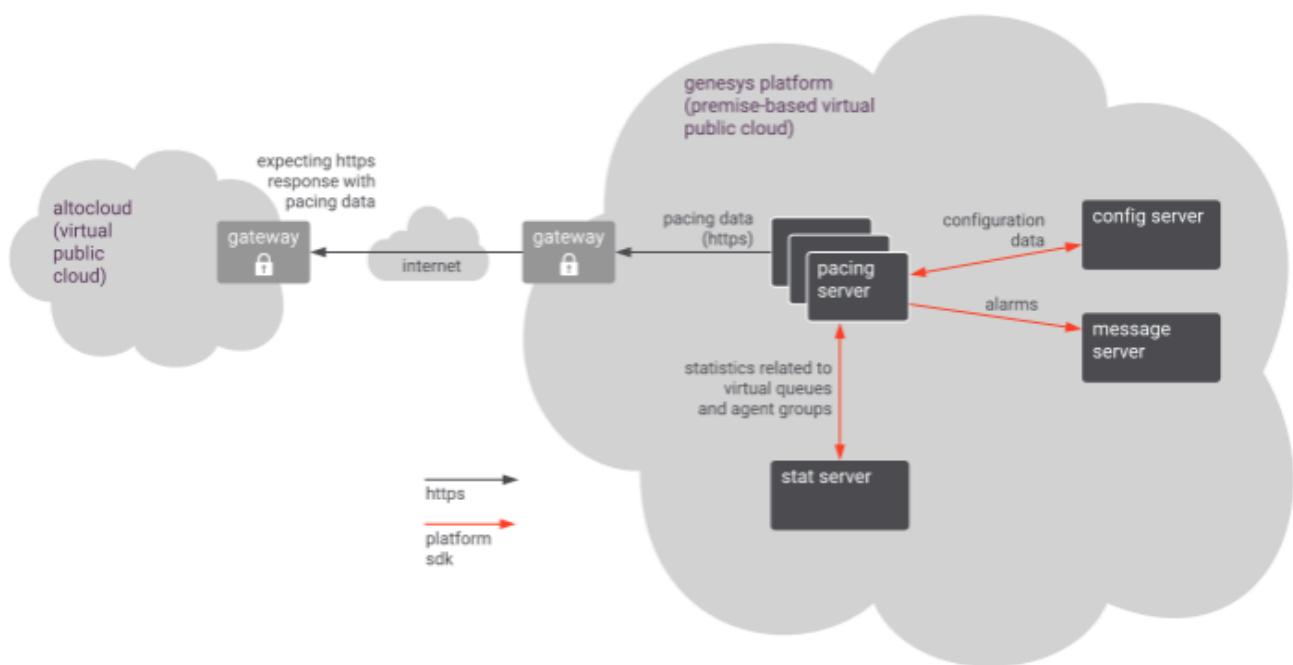
## High-level architecture

1. When the Pacing Server starts, it opens a connection to Config Server and discovers its pacing targets.
2. Pacing Server opens a connection to Stat Server.
3. Pacing Server starts performing its pacing calculations, using Stat Server as a primary source of information. For the initial calculations, the count of pending invitations is considered to be **0**.
4. Pacing Server uses the currently available data from Stat Server and Genesys Predictive Engagement to calculate the number of proactive invites that can be sent during the specified time interval.
5. Pacing Server sends the results of its calculations to Genesys Predictive Engagement.
6. Genesys Predictive Engagement processes the pacing information and returns the count of pending invitations.
7. A new pacing cycle is started from step 4, using the information sent by Genesys Predictive Engagement.

If something goes wrong—for example, if Pacing Server starts using too much memory, or if Genesys Predictive Engagement becomes unreachable—Pacing Server sends an alarm to Message Server.

## How Pacing works

---



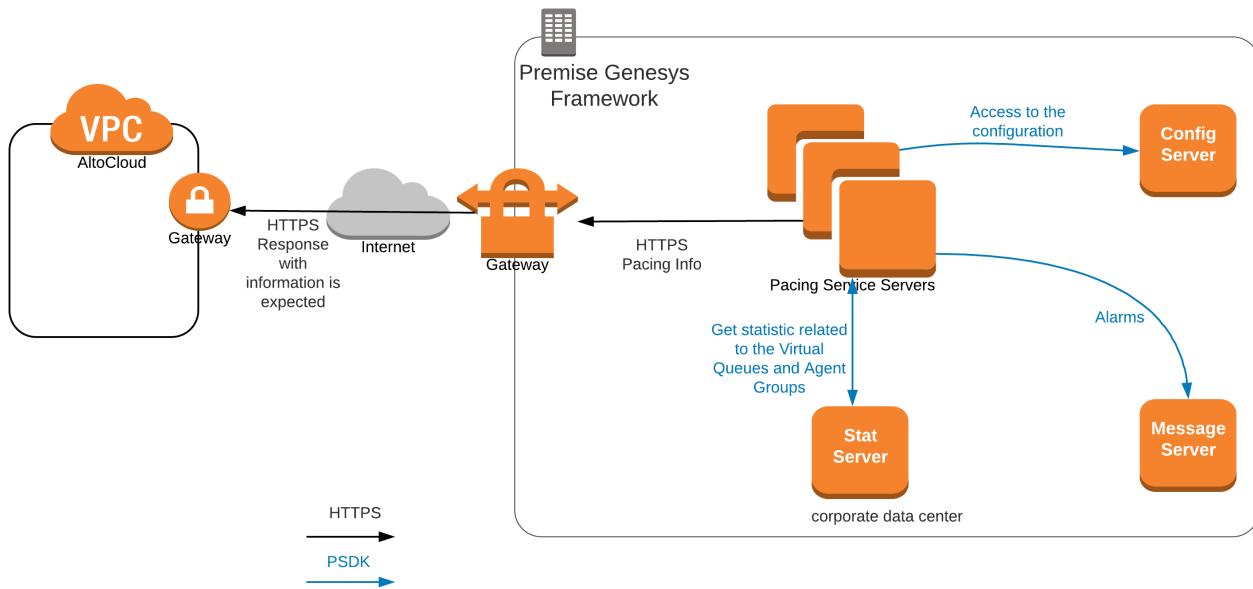
# Architecture

## Contents

- 1 Architecture diagram
- 2 Performance
- 3 Scalability
- 4 High availability

Read about performance and high availability considerations for Agent Pacing Service.

## Architecture diagram



The Agent Pacing Service connects to only one Stat Server (high-availability pair).

### Important

The Agent Pacing Service requires that the Stat Server has visibility of all potential pacing targets.

Multiple actively running pacing nodes can belong to the same pacing cluster. Under normal conditions, there is only one master pacing node that provides pacing target information to Genesys Predictive Engagement. All pacing nodes are interconnected, and they self-determine which is the master pacing node based on an internal election algorithm.

## Performance

Each pacing node is designed to support up to 10,000 pacing targets. The execution of the pacing algorithm does not put significant pressure on either the memory or the CPU. However, as the number of pacing targets increases, so does the size of the HTTP request and the corresponding response between Pacing Service and Genesys Predictive Engagement. For this reason, network bandwidth becomes a concern in terms of system stability.

Problems may occur if the roundtrip of the HTTP package reaches the timeout limit or the pacing refresh period. If you experience performance problems, try tuning your network bandwidth or reducing the number of pacing targets.

### Important

The performance of a single pacing server node has been verified for the following configuration:

- 2 CPUs (Core i7)
- 6 Gb RAM
- 2 Gb of allocated Java Heap memory

## Scalability

For nodes belonging to same pacing cluster, the scalability model is N+1. Based on performance testing, two pacing nodes should be sufficient to sustain most deployments.

## High availability

This release of Agent Pacing Service works with a single Stat Server HA pair. To ensure high availability in a premises-based environment, install two (2) Pacing Server nodes.

# Install the Pacing Service

## Contents

- [1 About the Pacing Service Cluster](#)
- [2 Deploy Pacing Server](#)
- [3 Import the Pacing Service cluster application template](#)
- [4 Create the cluster application](#)
- [5 Configure the cluster application](#)
- [6 Import the Pacing Server application template](#)
- [7 Create and configure a node application](#)
- [8 Add a node application to the cluster](#)
- [9 Install Pacing Server](#)
- [10 Configure alarms](#)
- [11 Configure a forward proxy \(optional\)](#)

Complete the steps to install the Pacing Server.

The 9.0 release of Pacing Server can be used on premises.

## About the Pacing Service Cluster

Agent Pacing Service uses an N+1 architecture. This means that almost all of the configuration is handled at the level of the cluster, rather than at the level of individual nodes. In particular,

- The cluster combines one or more nodes. You must install and configure at least one Pacing Server node in order to use Pacing Service. Every time you add a node to the cluster, you must create and configure it using the same steps you used for the first node.
- All nodes are treated as equivalent to each other. Because of this, the nodes only contain a connection to the cluster application.
- Connections to other Genesys servers—such as Stat Server and Message Server—are defined for the cluster.

### Important

Pacing Service is a single-tenant application—it only has access to the tenant, objects, and transaction list of the tenant it is configured for. In a multi-tenant environment, you can configure a Pacing Service for each individual tenant.

## Deploy Pacing Server

To deploy Pacing Server, follow these steps:

1. Import the Pacing Service Cluster Application template
2. Create the cluster application
3. Configure the cluster application
4. Import the Pacing Server Application template
5. Create and configure a node application
6. Add the node application to the cluster

## Install the Pacing Service

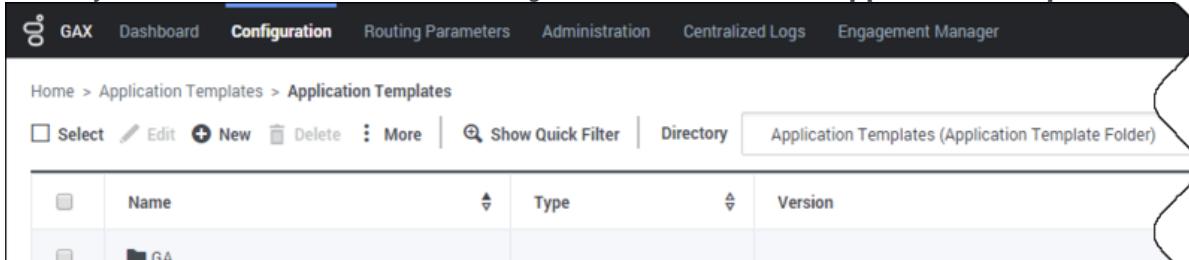
---

7. Install Pacing Server
8. Configure alarms
9. Configure a forward proxy (optional)

## Import the Pacing Service cluster application template

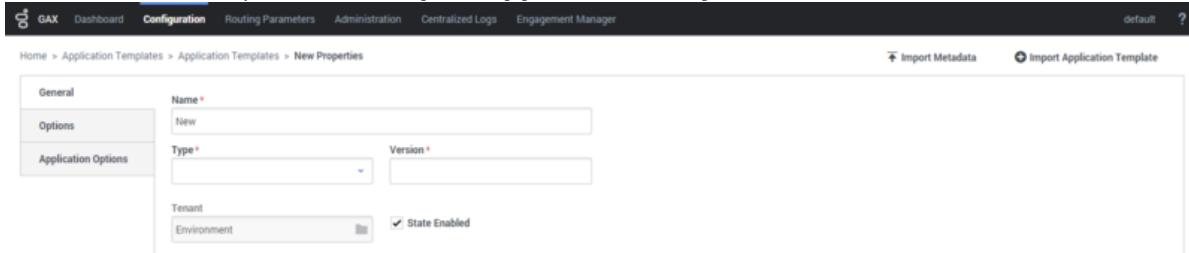
**Prerequisite:** Download the Pacing Server installation on the host where you are going to install it.

1. Open Genesys Administrator Extension and navigate to **Environment > Application Templates**:

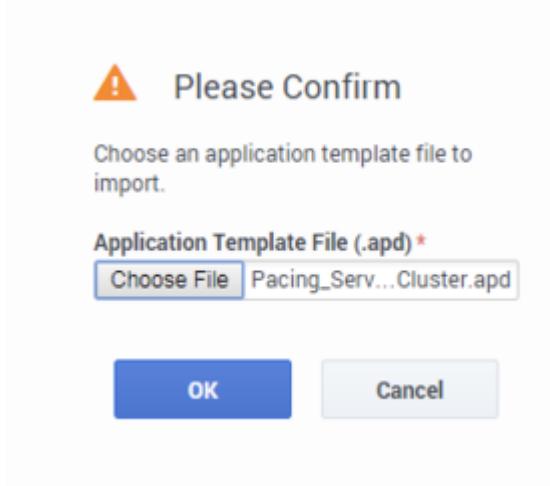


Name	Type	Version
GA		

2. Click **New**. In the **Tasks** panel, click **Import Application Template**:



3. Navigate to the **Pacing Server installation root/templates** folder and select **Pacing\_Server\_Cluster.apd**:



⚠ Please Confirm

Choose an application template file to import.

Application Template File (.apd)\*

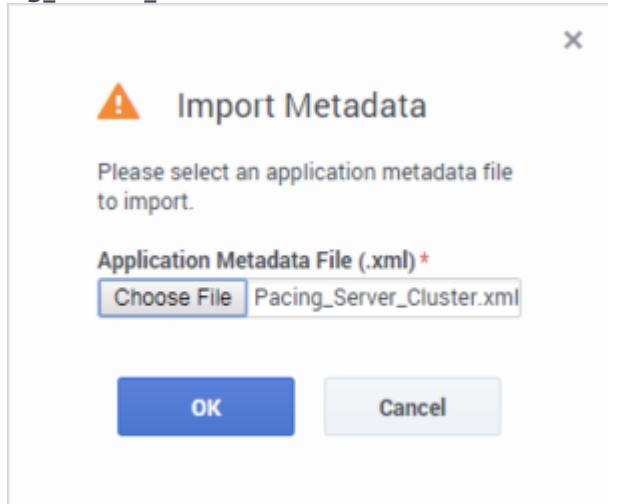
Choose File Pacing\_Serv...Cluster.apd

OK Cancel

## Install the Pacing Service

---

4. Click **Import Metadata**.
5. Navigate to the **Pacing Server installation root/templates** folder and select **Pacing\_Server\_Cluster.xml**:

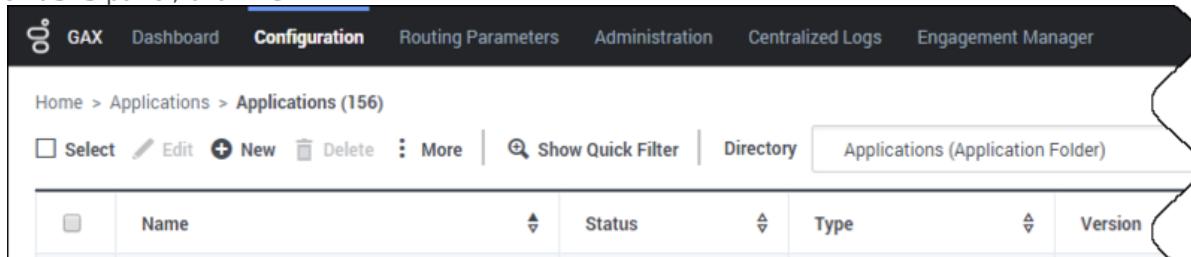


6. Click **Save**

## Create the cluster application

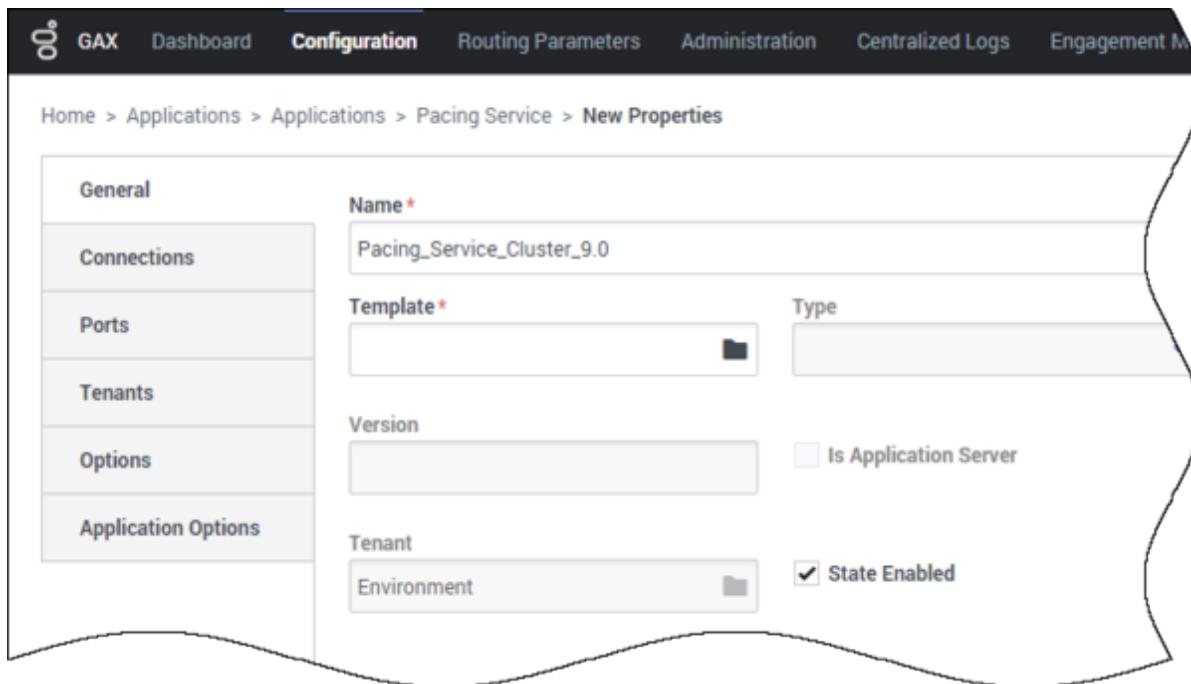
**Prerequisite:** Import the Pacing Service cluster application template

1. Open Genesys Administrator Extension and navigate to **Environment > Applications**.
2. In the **Tasks** panel, click **New**:



3. Specify the name of the Pacing Service cluster application (for example, *Pacing\_Service\_Cluster\_9.0*), then click to select a template:

## Install the Pacing Service



Home > Applications > Applications > Pacing Service > New Properties

**General**

Name \*: Pacing\_Service\_Cluster\_9.0

Template \*:

Type:

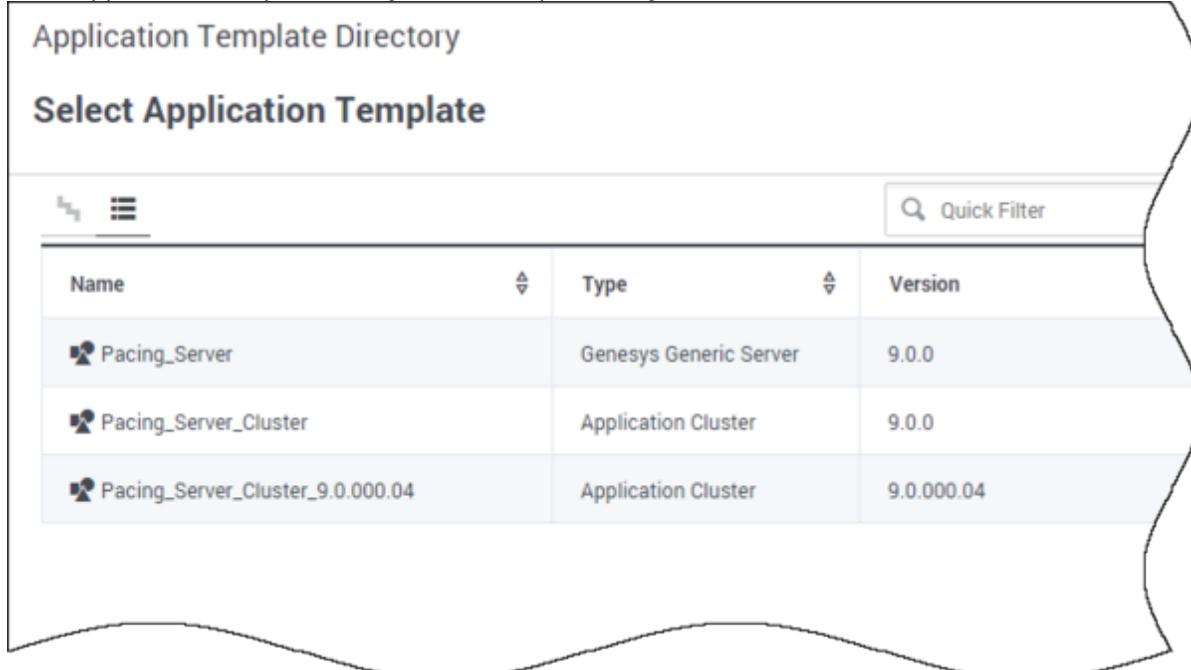
Version:

Is Application Server

Tenant: Environment

State Enabled

4. Select the application template that you created previously:



Application Template Directory

Select Application Template

Name	Type	Version
Pacing_Server	Genesys Generic Server	9.0.0
Pacing_Server_Cluster	Application Cluster	9.0.0
Pacing_Server_Cluster_9.0.000.04	Application Cluster	9.0.000.04

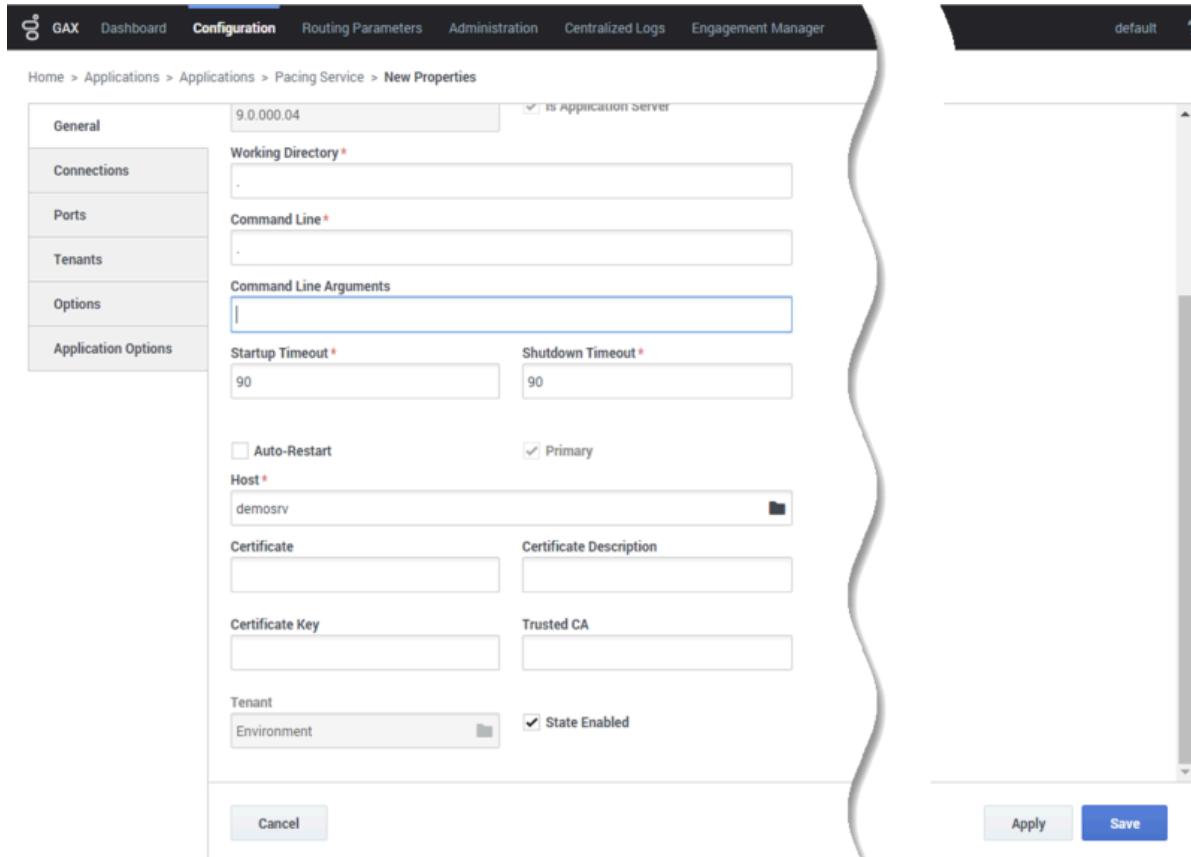
5. Select the **General** tab and:

- Set the value of the **Working Directory** and **Command Line** fields to . (a single dot).
- Make sure that **State** is enabled.
- Select the **Host** on which the Pacing Service Cluster will reside. This is usually the fully qualified

## Install the Pacing Service

domain name or IP address of the load balancer that provides access to the Pacing Service (that is, the set of Pacing Servers)

- Click **Save**.



Home > Applications > Applications > Pacing Service > New Properties

General

Working Directory \*

Command Line \*

Command Line Arguments

Startup Timeout \*

Shutdown Timeout \*

Auto-Restart

Host \*

Certificate

Certificate Description

Certificate Key

Trusted CA

Tenant

Environment

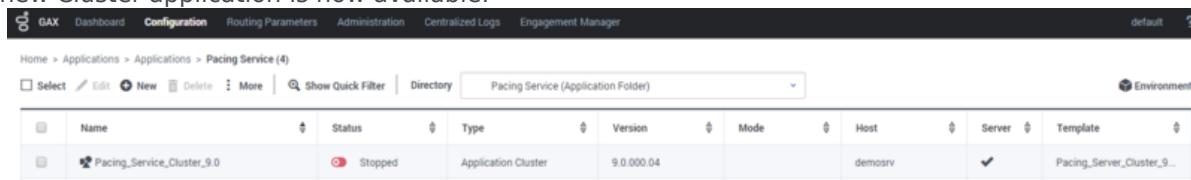
State Enabled

Cancel

Apply

Save

- The new Cluster application is now available:



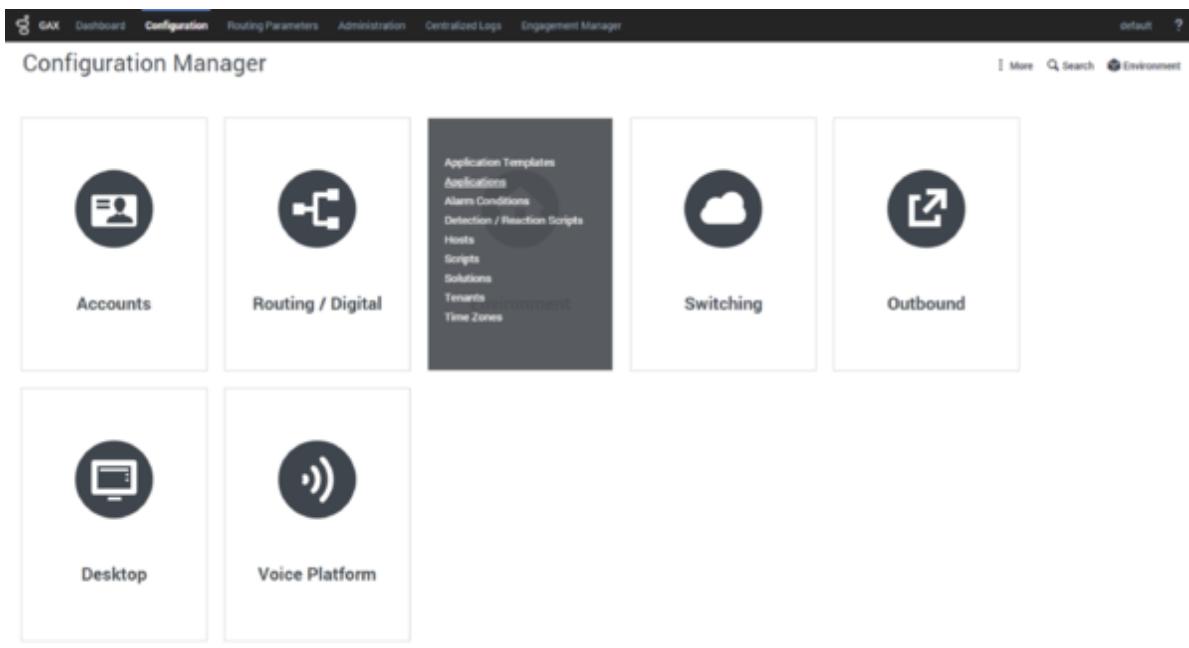
Name	Status	Type	Version	Mode	Host	Server	Template
Pacing_Service_Cluster_9.0	Stopped	Application Cluster	9.0.000.04		demosrv	✓	Pacing_Server_Cluster_9...

## Configure the cluster application

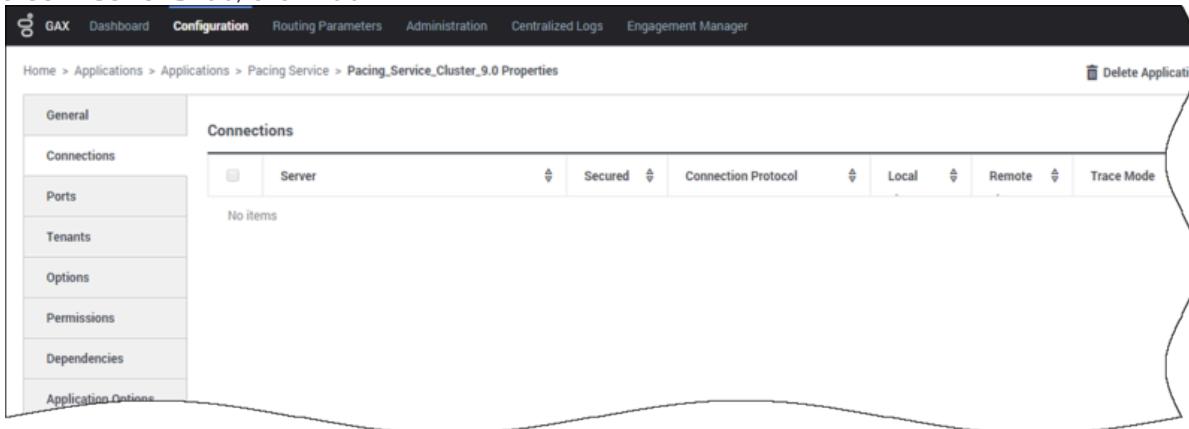
**Prerequisite:** Create the cluster application

- Open Genesys Administrator Extension and navigate to **Environment > Applications**, then open your cluster application:

## Install the Pacing Service



2. In the **Connections** tab, click **Add**:



3. Select the **Stat Server** application. Leave the connection port ID set to **default**. Click **OK**:

New

Server \*

Port ID \*

Connection Protocol

Local Timeout

Remote Timeout

Trace Mode \*

Transport Protocol Parameters

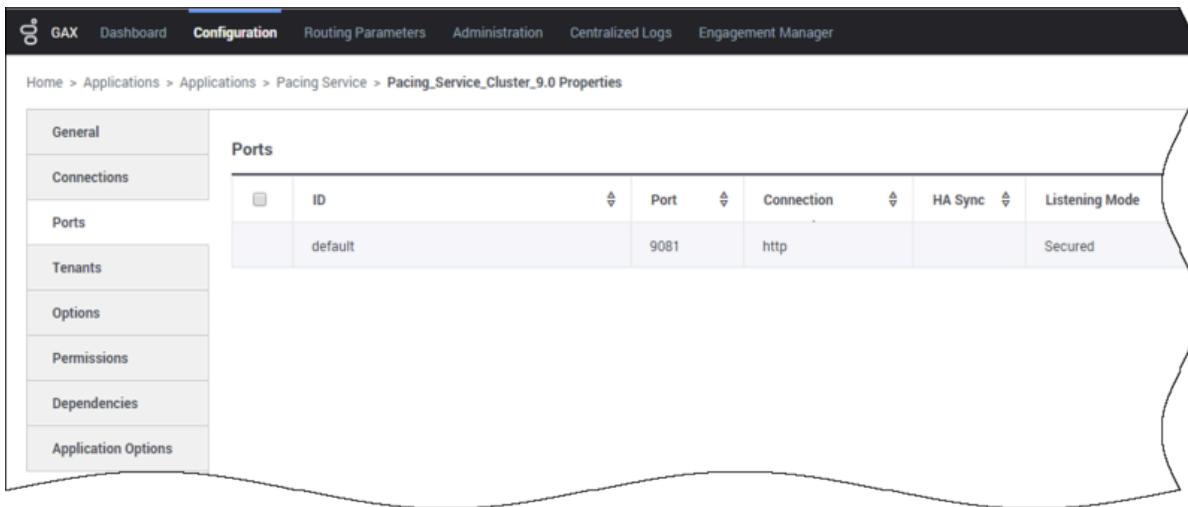
Application Parameters

4. If appropriate, you can also add a connection to Message Server (to apply the network logging options).
5. Navigate to the **Ports** tab and set the desired value of the default port. **Note:** The cluster application does not represent a particular server: you must treat as the load balancing entry point. Set the connection type to **http**. Genesys recommends that you use a secure connection.

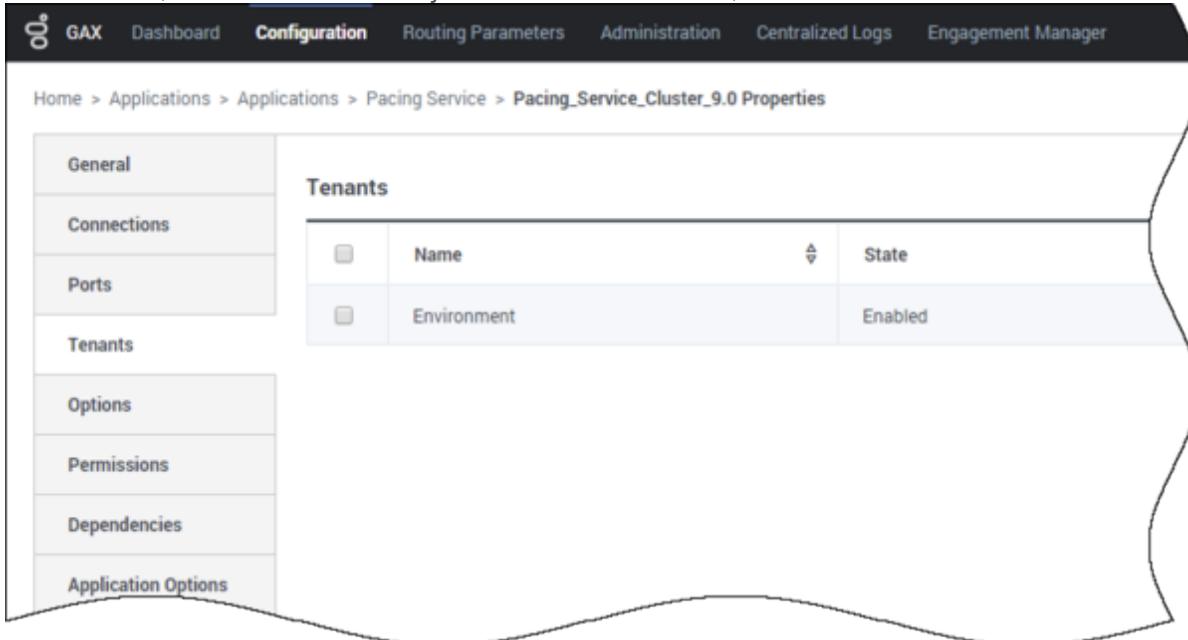
## Install the Pacing Service

---



ID	Port	Connection	HA Sync	Listening Mode
default	9081	http		Secured

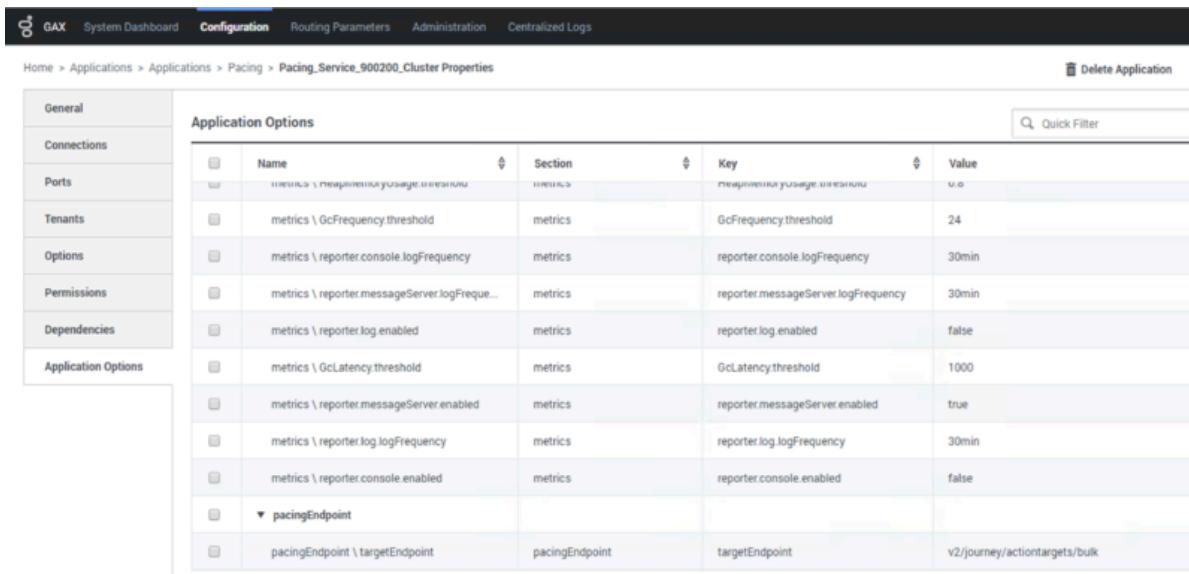
6. In the **Tenant** tab, click **Add** and select your tenant. For instance, *Environment*.



Name	State
Environment	Enabled

7. Navigate to the **Application Options** tab. You can keep the default values for all options except the ones in the **pacingEndpoint** section.

## Install the Pacing Service



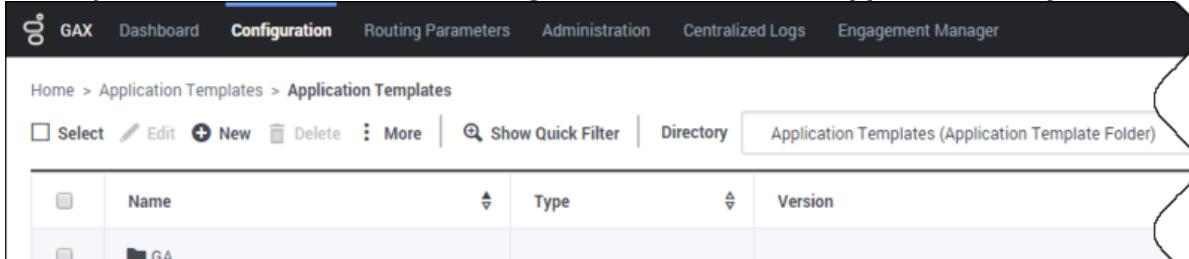
Application Options				
	Name	Section	Key	Value
	metrics \ measurementMemoryUsage.threshold	metrics	measurementMemoryUsage.threshold	0.0
	metrics \ GcFrequency.threshold	metrics	GcFrequency.threshold	24
	metrics \ reporter.console.logFrequency	metrics	reporter.console.logFrequency	30min
	metrics \ reporter.messageServer.logFrequency	metrics	reporter.messageServer.logFrequency	30min
	metrics \ reporter.log.enabled	metrics	reporter.log.enabled	false
	metrics \ GcLatency.threshold	metrics	GcLatency.threshold	1000
	metrics \ reporter.messageServer.enabled	metrics	reporter.messageServer.enabled	true
	metrics \ reporter.log.logFrequency	metrics	reporter.log.logFrequency	30min
	metrics \ reporter.console.enabled	metrics	reporter.console.enabled	false
	pacingEndpoint	pacingEndpoint	targetEndpoint	v2/journey/actiontargets/bulk
	pacingEndpoint \ targetEndpoint	pacingEndpoint	targetEndpoint	v2/journey/actiontargets/bulk

8. Click **Save**.

## Import the Pacing Server application template

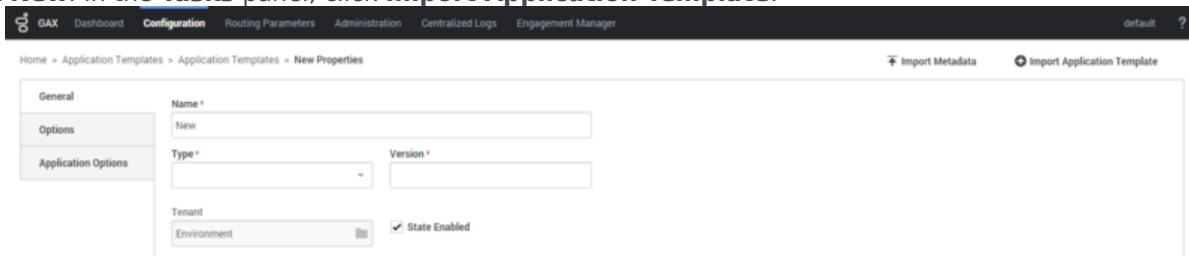
**Prerequisite:** Configure the cluster application

1. Open Genesys Administrator Extension and navigate to **Environment > Application Templates**:



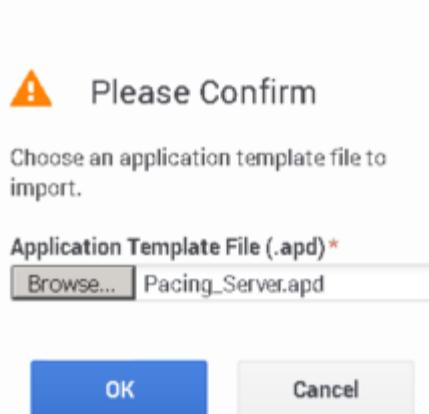
Name	Type	Version
GA		

2. Click **New**. In the **Tasks** panel, click **Import Application Template**:

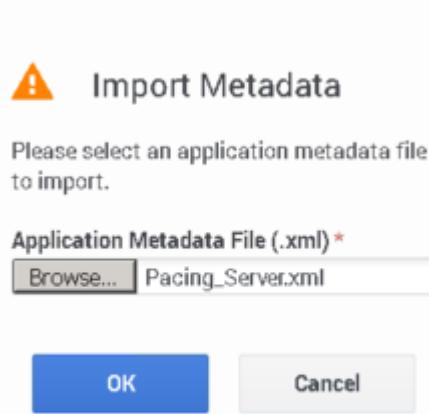


General	Name *	New
Options	Type *	Environment
Application Options	Version *	
Tenant		
Environment		
<input checked="" type="checkbox"/> State Enabled		

3. Navigate to the **Pacing Server installation root /templates** folder and select **Pacing\_Server.apd**:



4. Navigate to the **Pacing Server installation root/templates** folder and select **Pacing\_Server.xml**:



5. Click **Save**.

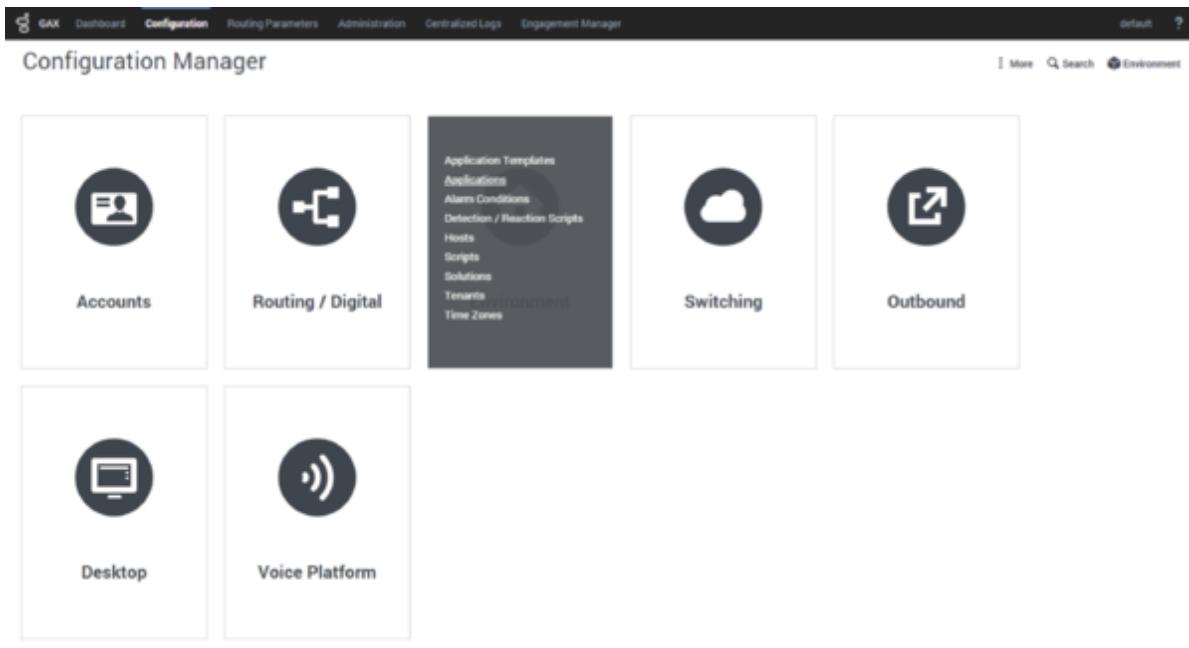
## Create and configure a node application

You must create and configure every node that you add to the cluster, using the instructions in this section and the following one.

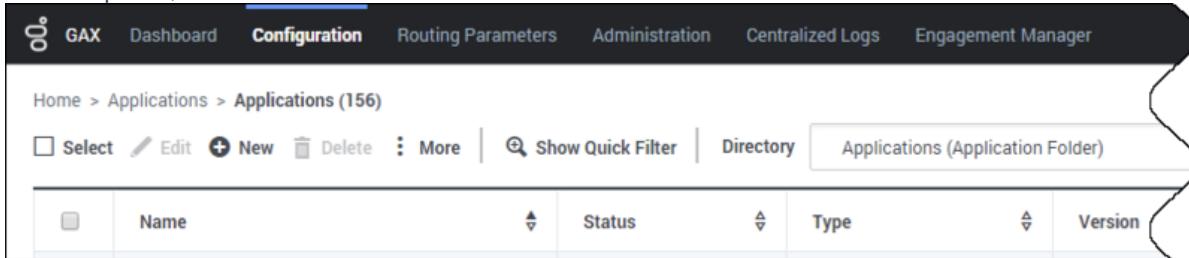
**Prerequisite:** Import the Pacing Server application template

1. Open Genesys Administrator Extension and navigate to **Environment > Applications**.

## Install the Pacing Service

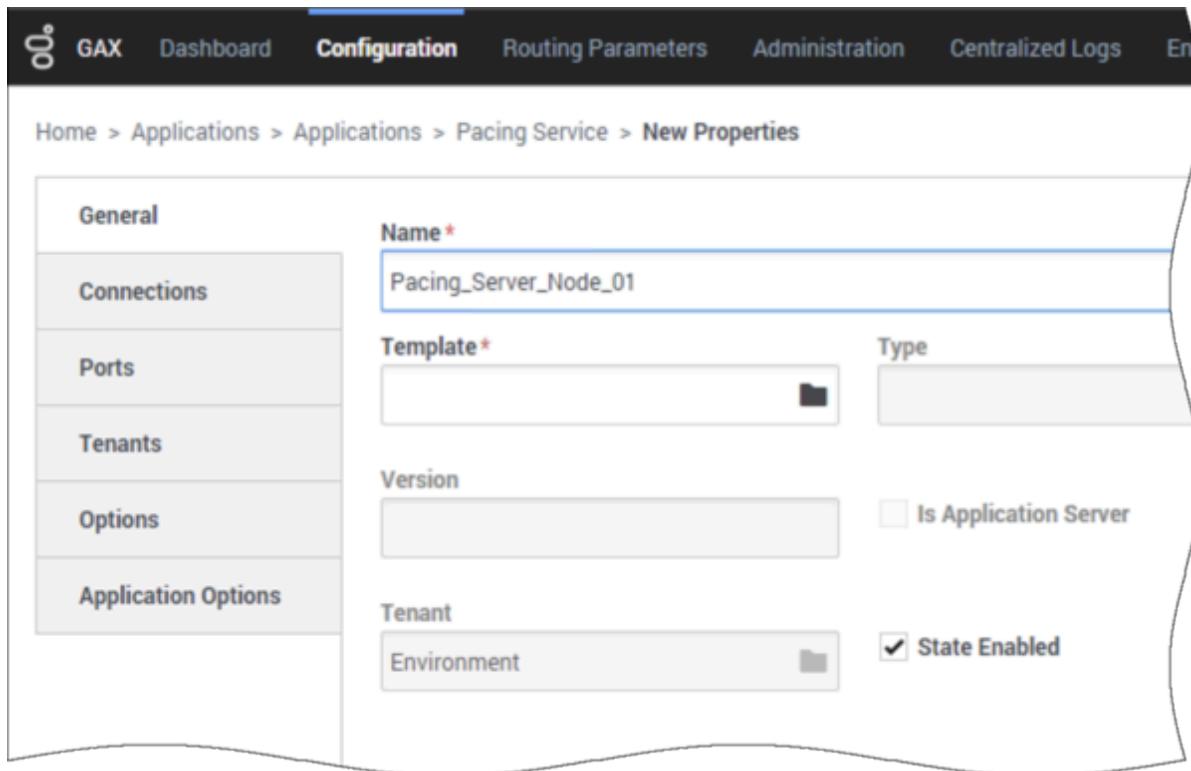


2. In the **Tasks** panel, click New:



3. Specify the name of the Pacing Server application, for example, *Pacing\_Server\_Node\_01*, then click to select a template for creating the application:

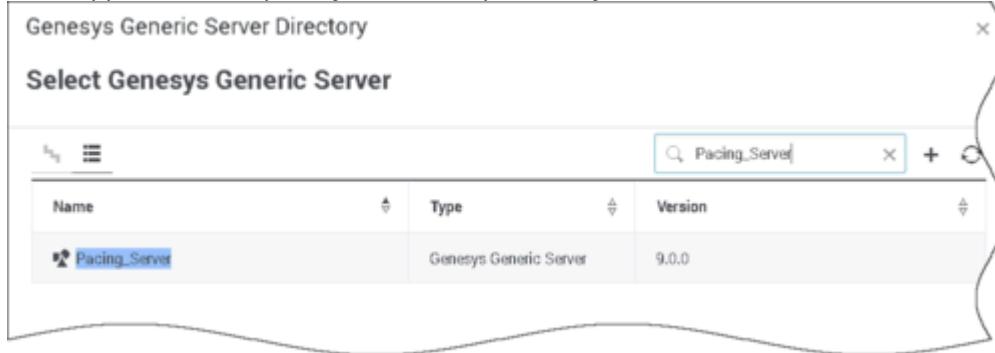
## Install the Pacing Service



Home > Applications > Applications > Pacing Service > New Properties

General	Name *	Pacing_Server_Node_01
Connections	Template *	
Ports	Type	
Tenants	Version	<input type="checkbox"/> Is Application Server
Options	Tenant	
Application Options	Environment	<input checked="" type="checkbox"/> State Enabled

4. Select the application template you created previously:

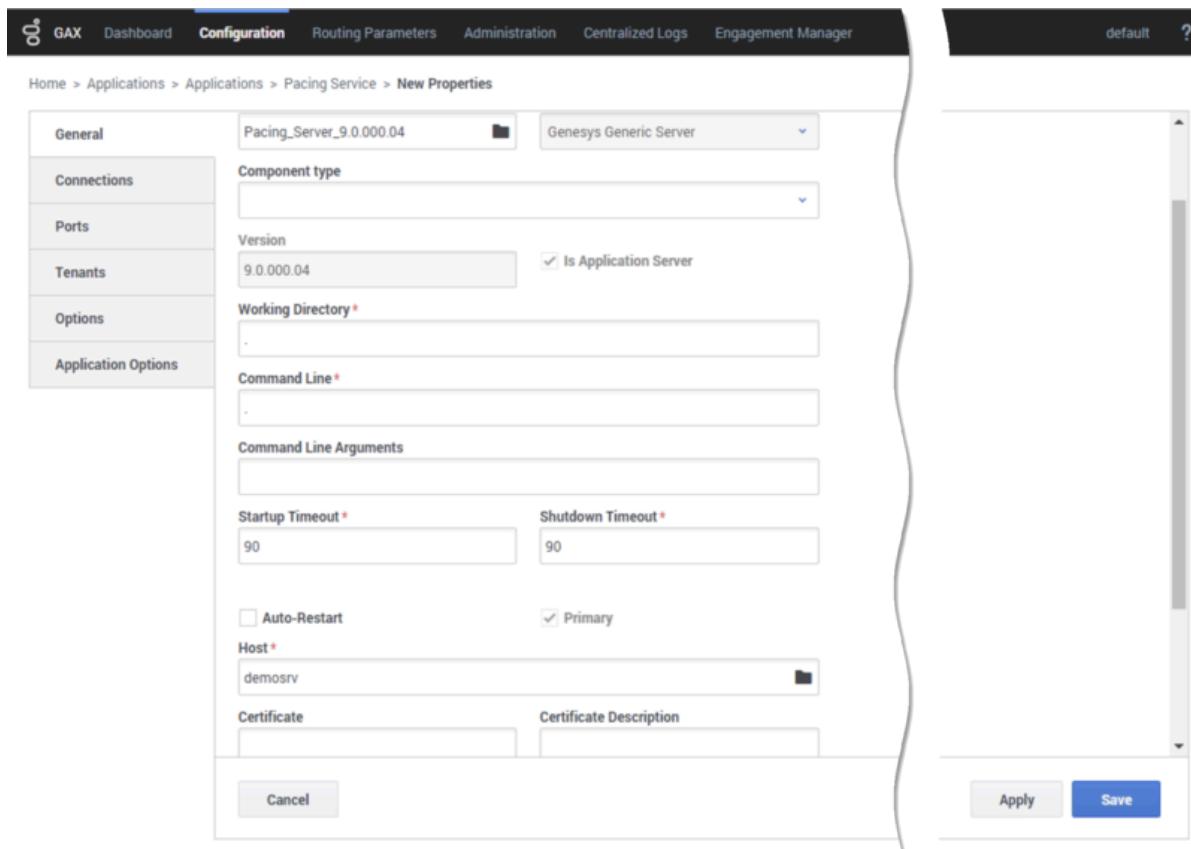


Name	Type	Version
Pacing_Server	Genesys Generic Server	9.0.0

5. Select the **General** tab and:

1. Set the value of the **Working Directory** and **Command Line** fields to . (a single dot).
2. Make sure that **State** is enabled.
3. Select the **Host** on which the Pacing Server node will reside.
4. Click **Save**.

## Install the Pacing Service

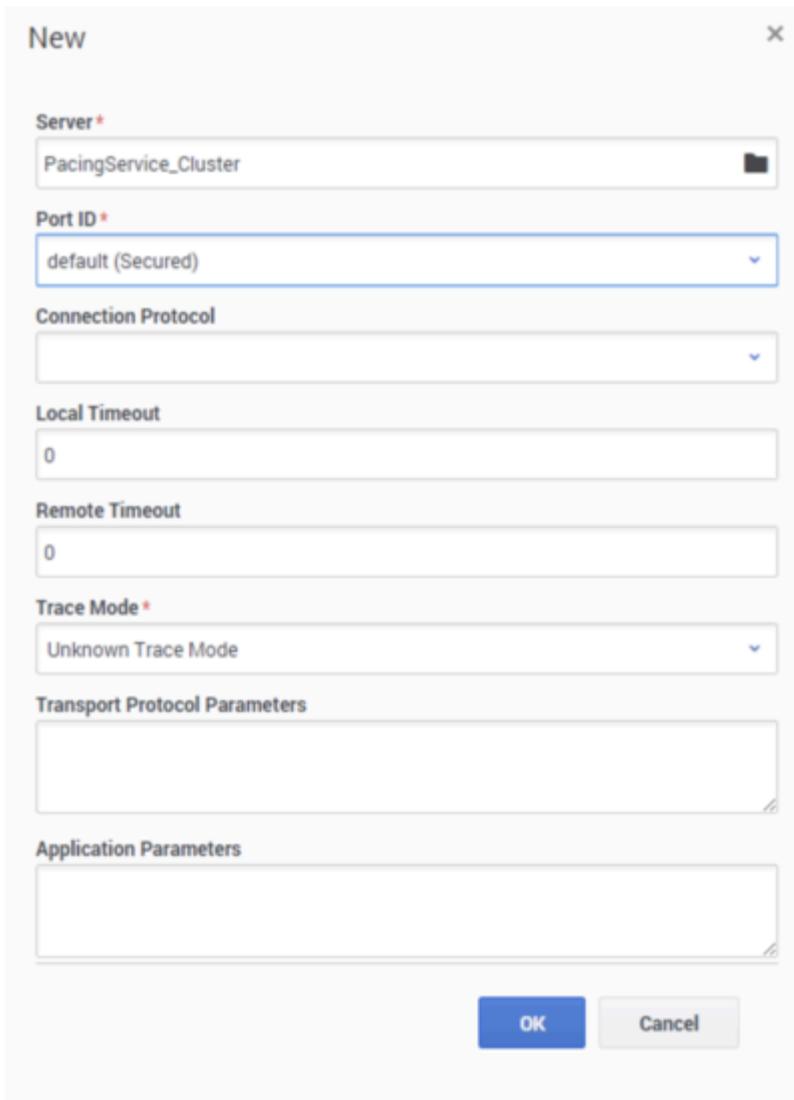


The screenshot shows the GAX Configuration interface with the 'Configuration' tab selected. The main screen is titled 'New Properties' for the application 'Pacing\_Server\_9.0.000.04'. The 'General' tab is active, displaying the following configuration:

- Component type:** Genesys Generic Server
- Version:** 9.0.000.04
- Is Application Server:**
- Working Directory:** -
- Command Line:** -
- Command Line Arguments:** -
- Startup Timeout:** 90
- Shutdown Timeout:** 90
- Auto-Restart:**
- Primary:**
- Host:** demosrv
- Certificate:** -
- Certificate Description:** -

At the bottom of the screen are 'Cancel', 'Apply', and 'Save' buttons.

6. Navigate to the **Connections** tab and click **Add**. Select the Pacing Service cluster application you created previously, then click **OK**:

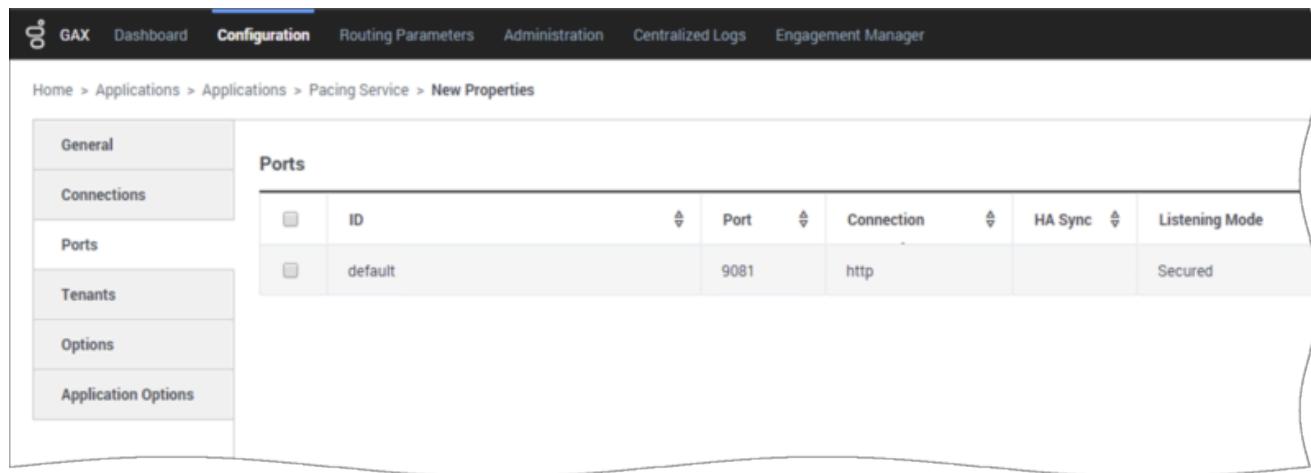


7. Navigate to the **Ports** tab and make sure that the value of the default port is set to the port that this Pacing Server node should listen on. Also, set the protocol type to **http** or **https**.

### Important

- Genesys recommends that you always use the HTTPS protocol.
- Genesys recommends that all instances of Pacing Server listen on the same ports.
- Pacing Servers are connected in the cluster through a dedicated port on the host where the Pacing Server instance is installed. By default, this port is 7800. If this port is already in use, go to the Ports tab and add another port with ID **clustering** and the desired value. You don't have to specify a connection type for this port.

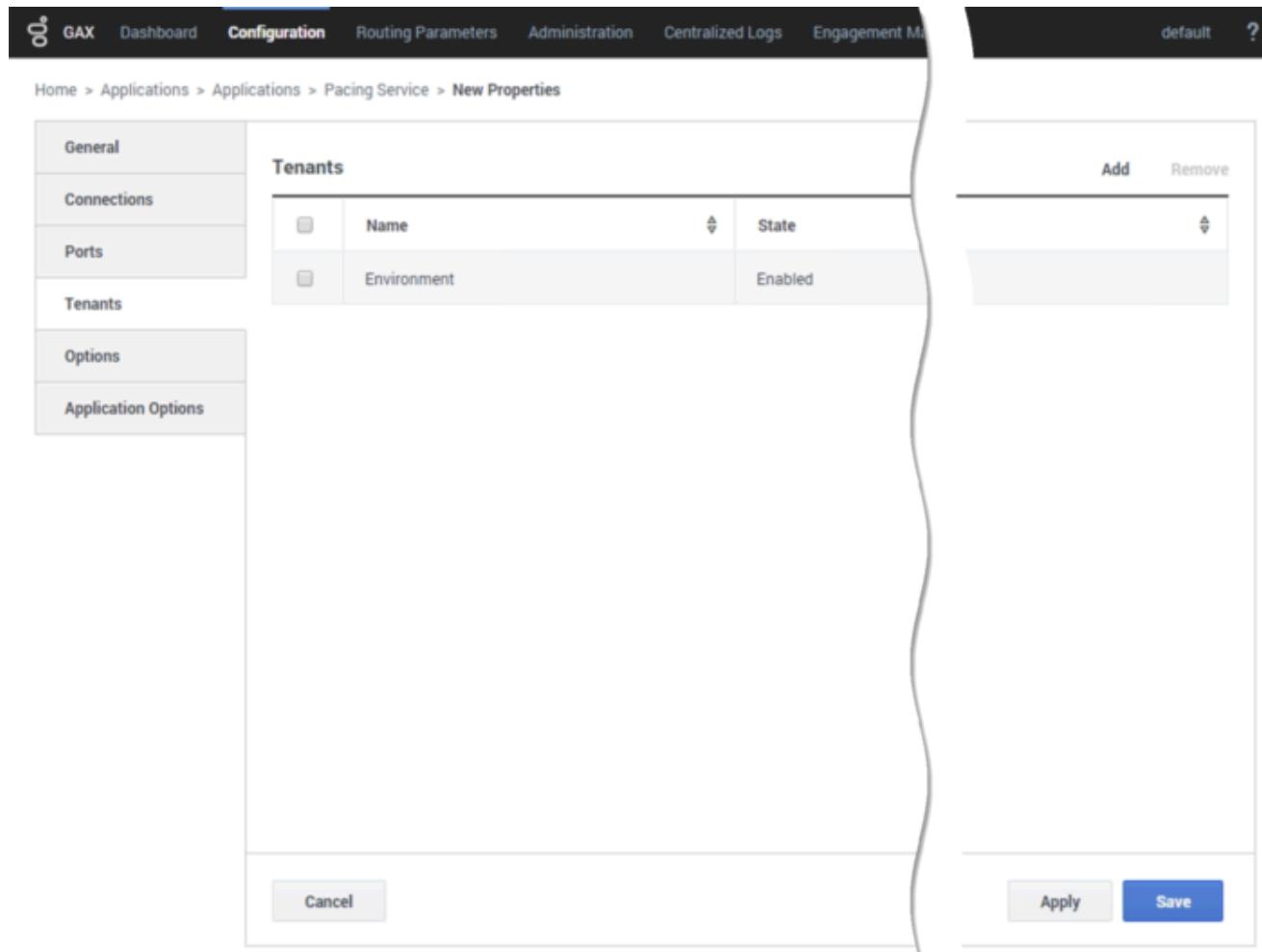
## Install the Pacing Service



The screenshot shows the GAX Configuration interface with the 'Configuration' tab selected. The left sidebar has tabs for General, Connections, Ports, Tenants, Options, and Application Options. The main content area is titled 'Ports' and contains a table with one row. The table columns are ID, Port, Connection, HA Sync, and Listening Mode. The row shows 'default' as the ID, '9081' as the Port, 'http' as the Connection, and 'Secured' as the Listening Mode.

ID	Port	Connection	HA Sync	Listening Mode
default	9081	http		Secured

8. Navigate to the **Tenants** tab and click **Add**. Select the same tenant that is specified for the Pacing Service Cluster application:



The screenshot shows the GAX Configuration interface with the 'Configuration' tab selected. The left sidebar has tabs for General, Connections, Ports, Tenants, Options, and Application Options. The main content area is titled 'Tenants' and contains a table with one row. The table columns are Name and State. The row shows 'Environment' as the Name and 'Enabled' as the State. On the right side, there are 'Add' and 'Remove' buttons. At the bottom, there are 'Cancel', 'Apply', and 'Save' buttons.

Name	State
Environment	Enabled

9. Click **Save**.

Add a node application to the cluster

**Prerequisites:**

- Create the cluster application
- Configure the cluster application
- Create and Configure a node application

Although a single-node configuration works well in a lab environment, in order to provide high availability in production, you must use multiple nodes.

**Important**

- Every time you establish a new node, you must complete the steps described in Create and Configure a node application.
- If you use multiple nodes, you must set up load balancing in your environment.

Install Pacing Server

**Prerequisite:** Create and Configure a node application

**Important**

For more information on how to install apps that you have configured in Genesys Administrator Extension, consult Generic Installation Procedures.

Windows

Windows

1. In your installation package, locate and double-click **setup.exe**. InstallShield opens the welcome screen.
2. Click **Next**. The **Connection Parameters to the Configuration Server** screen appears.

3. Under **Host**, specify the host name and port number where Configuration Server is running. (This is the main listening port entered in the **Server Info** tab for Configuration Server.)
4. Under **User**, enter the user name and password for logging into Configuration Server.
5. Click **Next**. The **Select Application** screen appears.
6. Select the **Genesys Generic Server** application—that is, the Node app you created above—that you are installing. The **Application Properties** area shows the **Type**, **Host**, **Working Directory**, **Command Line executable**, and **Command Line Arguments** information previously entered in the **Server Info** and **Start Info** tabs of the selected Application object.
7. Click **Next**. The **Choose Destination Location** screen appears.
8. Under **Destination Folder**, either keep the default value or browse for the desired installation location.
9. Click **Next**. The **Backup Configuration Server Parameters** screen appears.
10. If you have a backup Configuration Server, enter the **Host** name and **Port**.
11. Click **Next**. The **Ready to Install** screen appears.
12. Click **Install**. The Genesys Installation Wizard indicates it is performing the requested operation for Pacing Server. When it is finished, the **Installation Complete** screen appears.
13. Click **Finish** to complete your installation of the Pacing Server.

## Linux

### Linux

1. Open the Pacing Server IP in a terminal window, and run **Install.sh**. The Genesys Installation starts.
2. Enter the **hostname** of the host on which you are going to install.
3. Enter the connection information to log into Configuration Server:
  1. The **hostname**. For instance, *demosrv.genesyslab.com*.
  2. The **listening port**. For instance, *2020*.
  3. The **user name**. For instance, *demo*.
  4. The **password**.
5. If the connection settings are successful, a list of keys and Pacing Server applications is displayed.
4. Enter the key for the Pacing Server application—that is, the Node app you created above—that you are installing.
5. Enter the location where Pacing Server is to be installed on your host.

**Note:** This location must match the previous settings that you entered in Configuration Server.

1. If you have a backup Configuration Server, enter the **Host** name and **Port**.
2. If the installation is successful, the console displays the following message:

*Installation of Pacing Server, version 9.0.x has completed successfully.*

## Important

This installation procedure automatically provisions all of the Pacing Server-related configuration information. For more information, see Provisioning.

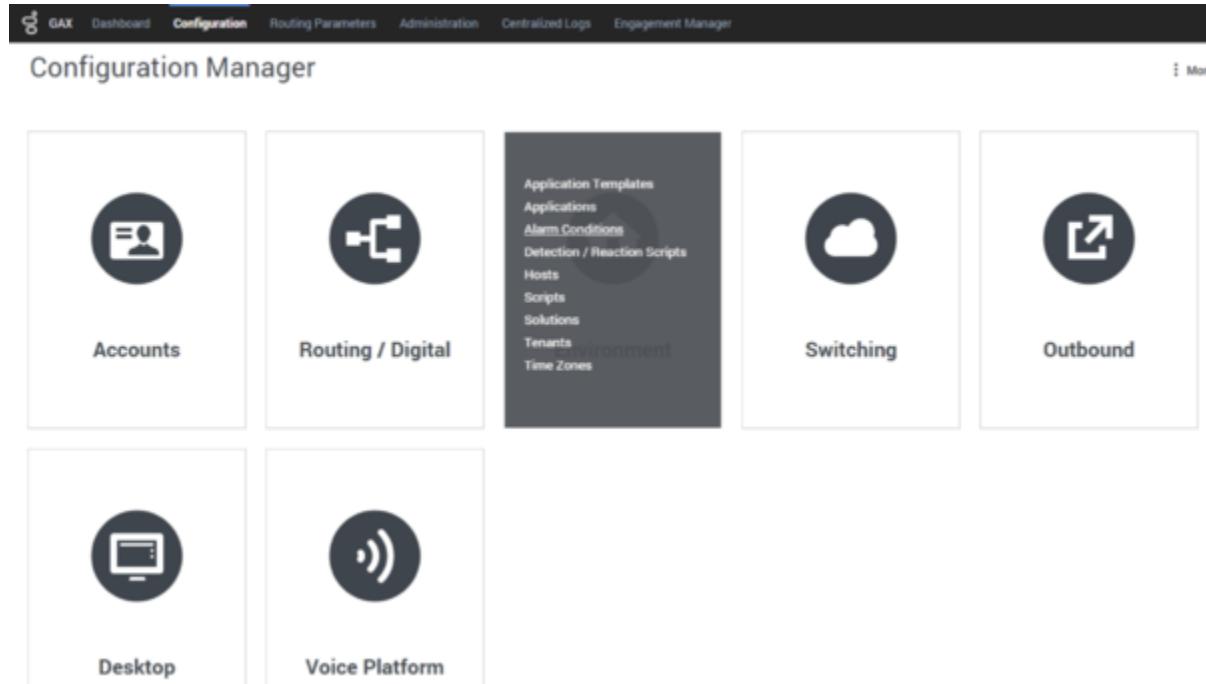
## Configure alarms

Genesys recommends that you tune the following Pacing Server-related alarms:

- GC Latency
- Heap Memory Usage
- PacingRequestsFailed
- PacingResponseTargetException
- PacingResponseValidationException

Although these alarms are created automatically during installation, you can remove them if necessary and then re-create them manually.

To access the alarms in Genesys Administrator Extension, open **Environment > Alarm Conditions**.



The automatic provisioning procedure places the Pacing Server-related alarms in the dedicated **PacingService** alarm folder.

## Configure a forward proxy (optional)

### Important

This feature is available in release 9.0.000.10 and higher.

If your environment permits connections to the Internet only through a forward proxy service (such as DMZ or your local intranet), configure the forward-proxy options, so the Pacing Service can connect to Genesys Predictive Engagement.

# Provisioning

## Contents

- 1 About Provisioning
- 2 Transaction object for hybrid integrations
- 3 Created objects
  - 3.1 Alarm conditions
  - 3.2 Genesys Stat Server application
  - 3.3 Files specific to Pacing Server
- 4 Run the provisioning tool

Use the provisioning tool or manually provision the Pacing Server.

## About Provisioning

You can use Genesys Administrator Extension to manually create all of the Pacing Server-related configuration information in Configuration Server. Or you can do this automatically by running the provisioning tool, located in the **Pacing Server installation directory/tools/provisioning** folder.

### Important

The transaction object, **hybrid\_integration**, is not controlled by the provisioning tool and should be manually managed.

### Tip

- The provisioning tool is run automatically during the installation process.
- You can run the provisioning tool under Windows or Linux.

## Transaction object for hybrid integrations

A transaction object is needed for Genesys components to authenticate with Genesys Cloud CX.

Create the transaction object with name (and alias) **hybrid\_integration** and type **List** in the **Transactions** folder of the tenant that is specified for Pacing Service (for more information, see *Configure the cluster application* in *Install the Pacing Service*). Be sure to include the following sections in the annex of the transaction object:

### 1. Section **general**

- Option **base\_auth\_url**: This is the base URL that is used for integration with Genesys Cloud CX authentication services. The **base\_auth\_url** is expected to be `https://[region_host]/oauth/token`. `[region_host]` will be the authentication-based FQDN for the appropriate region. For more information see *Regions*.

- Option **base\_service\_url**: This is the base URL that is used to access Genesys Cloud CX services. The **base\_service\_url** is expected to be `https://[region_host]/api/`. [region\_host] will be the API-based FQDN for the appropriate region. The rest of the url is service- and version-specific. For example, for the Agent Pacing Service, this url is specified in the **targetEndpoint** option.

### 2. Section **ewt**

- Option **client\_id**: This is the Client Credential Grant Client ID.
- Option **password**: This is the Client Credential Grant Client secret.

#### Important

Remember to restart the Agent Pacing Service if you change the values of these options.

## Created objects

The provisioning tool connects to Configuration Server and reads the configuration information for the Pacing Server applications. It creates the Genesys objects used by the Pacing Servers and edits the configuration files required to launch the Pacing Servers.

The following objects are created or updated when you run the provisioning tool.

### Alarm conditions

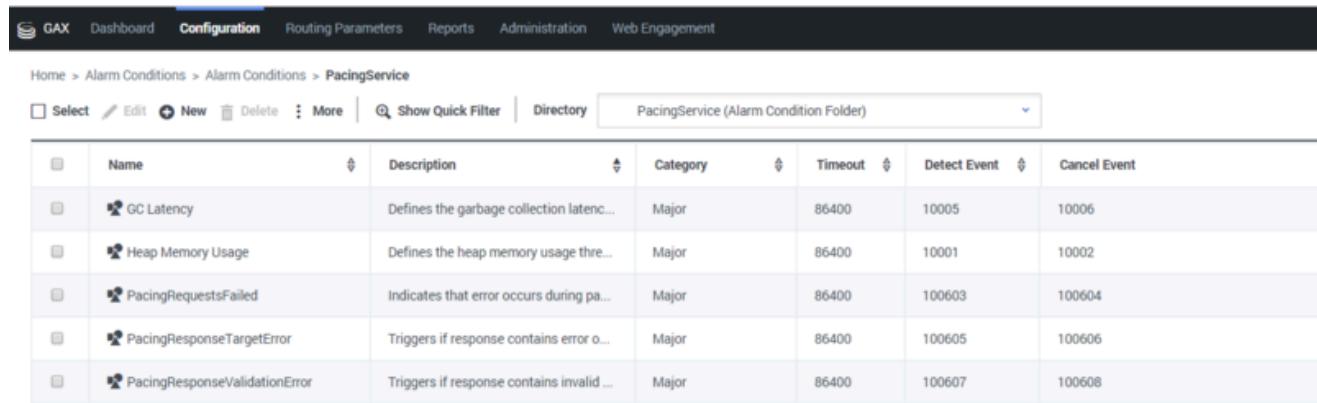
Provisioning creates the following default alarm conditions:

- GC Latency
- Heap Memory Usage
- PacingRequestsFailed
- PacingResponseTargetError
- PacingResponseValidationErrors

You can access these Alarm Conditions in Genesys Administrator Extension by navigating to the **PacingService** folder under **Environment > Alarm Conditions**:

## Provisioning

---



The screenshot shows the GAX Configuration interface with the 'PacingService' folder selected. The table lists six alarm conditions:

Name	Description	Category	Timeout	Detect Event	Cancel Event
GC Latency	Defines the garbage collection latency threshold.	Major	86400	10005	10006
Heap Memory Usage	Defines the heap memory usage threshold.	Major	86400	10001	10002
PacingRequestsFailed	Indicates that an error occurs during pacing requests.	Major	86400	100603	100604
PacingResponseTargetError	Triggers if the response contains an error target.	Major	86400	100605	100606
PacingResponseValidationError	Triggers if the response contains an invalid validation error.	Major	86400	100607	100608

## Genesys Stat Server application

Provisioning adds statistics and filters into the options of the Stat Server application connected to the Pacing Service Cluster application. If absent, the following statistics are added:

- PACS\_VQ\_Abandoned
- PACS\_VQ\_Accepted
- PACS\_VQ\_Current

The following filter is also added:

- PACS\_Proactive\_Interaction

## Files specific to Pacing Server

The provisioning tool can fill in pacing-related placeholder values in the following files, which are located in the **Pacing Server installation directory/server** folder:

- launcher.ini
- setenv.bat
- setenv.sh

Because this replacement is usually carried out successfully by the installation script, this feature of the provisioning tool is rarely used.

## Run the provisioning tool

Because the provisioning tool is triggered as part of the installation process, you don't usually need to run it manually. However, there are times when you do need to do that. For example, if pacing-related statistics for Stat Server application or pacing-related Alarm Conditions have been lost, you

should run the provisioning tool manually.

### Prerequisites

- The configuration applications for the Pacing Servers were created in Configuration Server.
- The connections for the Pacing Service Cluster application include the Stat Server applications.

For more information, see *Create and configure a node application* in *Install the Pacing Service*.

1. Navigate to the Pacing Server installation directory and open the **Pacing Server installation directory/tools/provisioning** folder.

2. From the command line, run the following command:

**provisioning.bat -host Configuration Server host name or IP address -port Configuration Server port -user Configuration Server user ID -password password for the specified user ID -app Pacing Server Application name**

For Linux, use the same command, but instead of **provisioning.bat**, specify **provisioning.sh**.

### Tip

- The user and password options may be optional, depending on your Configuration Server settings.

You can also use the **overwrite** option when you run the provisioning tool. In overwrite mode, the provisioning tool replaces old objects with new objects. Pacing-specific objects that already exist will be removed and new objects will be created instead. You will lose any changes you have made manually on pacing-specific objects. The command looks like this: **provisioning.bat -host Configuration Server host name or IP address -port Configuration Server port -user user -password password -app Pacing Server Application name -overwrite**

If provisioning is successful, the following message appears: *Provisioning script successfully finished his work*

# Configuration

## Contents

- [1 About Configuration](#)
- [2 Pacing targets](#)
- [3 Virtual queue](#)
- [4 Agent groups](#)
  - [4.1 Configure a regular agent group](#)
  - [4.2 Configure a blended agent group](#)
- [5 Pacing service](#)

Configure your pacing targets and how the Agent Pacing Service works.

## About Configuration

You must configure your pacing targets and your pacing service to make Agent Pacing Service work with your contact center.

### Important

Pacing Service is a single-tenant application—it only has access to the tenant, objects, and transaction list of the tenant it is configured for. In a multi-tenant environment, you can configure a Pacing Service for each individual tenant.

## Pacing targets

Each pacing target consists of:

- A media-specific virtual queue
- A set of one or more agent groups that are associated with this queue

## Virtual queue

Although contact centers typically contain many virtual queues, only some of them are appropriate for monitoring by the pacing service. The pacing service only monitors virtual queues that contain:

- A **pacing** section in the options
- A **media** option in the **pacing** section. The value of this option must be the name of a single media type. You cannot specify a list of media types.

The **pacing** section can include the following additional options:

- **optimizationGoal**—Specifies the highest allowable percentage of proactively triggered interactions that can be closed by visitors prior to an agent joining the session.

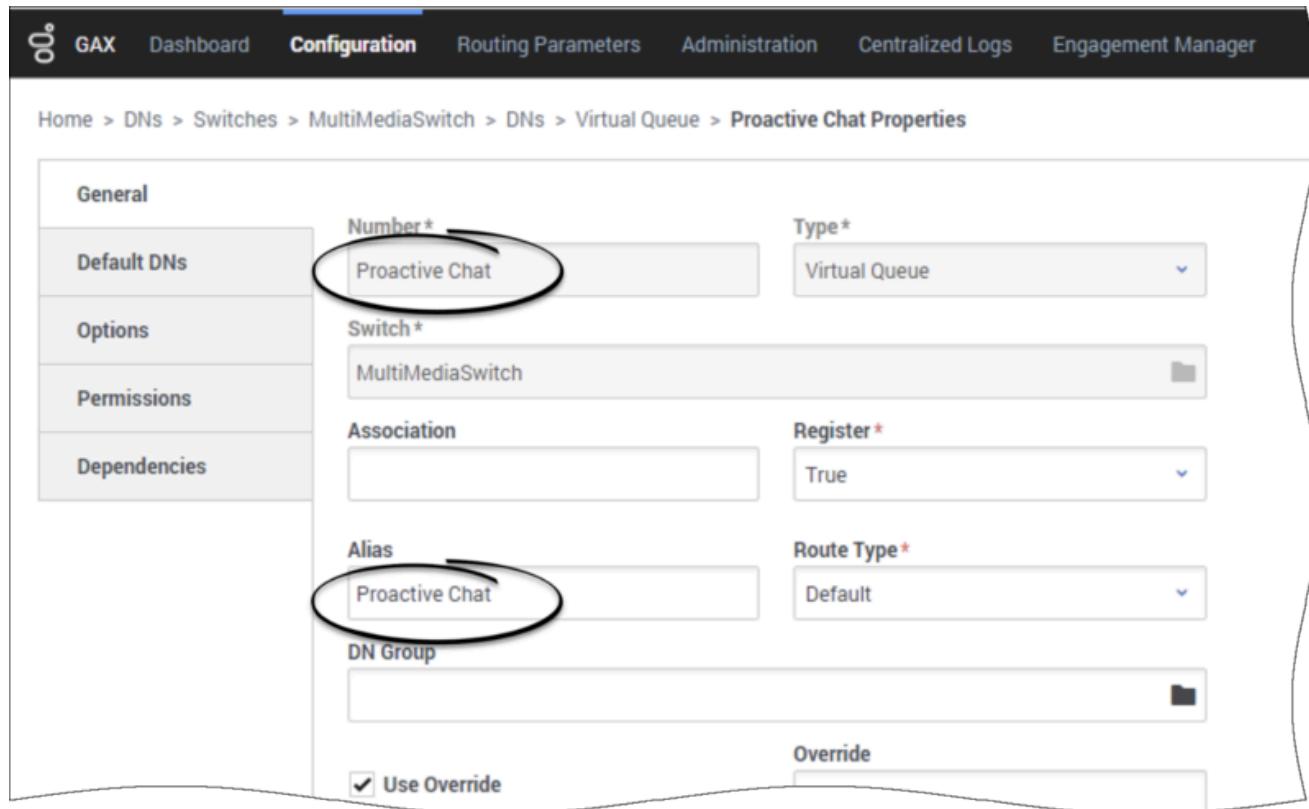
- The value must be a float between **0** and **100**
- The default value is **3** The pacing service considers all suitable virtual queues, regardless of which Switch objects they are associated with.

Here is a sample virtual queue configuration that is accessible in Genesys Administrator Extension:

Name	Section	Key	Value
pacing \ optimizationGoal	pacing	optimizationGoal	3
pacing \ media	pacing	media	chat

### Important

When configuring the virtual queue, you must set the value of **Alias** equal to the value of **Number**



The screenshot shows the 'Proactive Chat Properties' page within the GAX Configuration interface. The 'Number' field (containing 'Proactive Chat') and the 'Alias' field (also containing 'Proactive Chat') are circled in red. The 'Type' field is set to 'Virtual Queue'. The 'Switch' field is set to 'MultiMediaSwitch'. The 'Register' field is set to 'True'. The 'Route Type' field is set to 'Default'. The 'Use Override' checkbox is checked.

## Agent groups

Agent Pacing Service only considers agent groups for processing as part of a pacing target if they include the relevant virtual queue in their list of Origination DNs. The media type specified in the virtual queue is used to limit the *Ready* agents in an agent group to those who can process that media type.

Agent groups can specify more than one virtual queue, and each virtual queue can specify a different media type. For example, **Proactive Chat** could specify **chat** and **Proactive Voice** could specify **voice**. This capability is important for agents in the agent group who have *blended* capabilities—that is, those who can work with several media types simultaneously—as they can be considered for pacing targets that use each of their available media types.

### Configure a regular agent group

The following image shows an agent group called **Proactive Voice**. This agent group is associated with a voice-based virtual queue called **Proactive Voice**. The agent group and virtual queue form a pacing target called **Proactive Voice - voice**.

## Configuration

Home > Agent Groups > Agent Groups > Proactive Voice Properties

	Number	Type	Switch	Alias
<input type="checkbox"/>	Proactive Voice	Virtual Queue	MultiMediaSwitch	Proactive Voice

The next two images show two agent groups that are associated with the **Proactive Chat** virtual queue. The first one is **Proactive Chat** and the second is **Proactive Chat Sales**. These agent groups and the virtual queue form the **Proactive Chat - chat** pacing target.

Home > Agent Groups > Agent Groups > Proactive Chat Pro...

	Number	Type	Switch	Alias
<input type="checkbox"/>	Proactive Chat	Virtual Queue	MultiMediaS...	Proactive Chat

Home > Agent Groups > Agent Groups > Proactive Chat Sales Properties

**General**

Name \*

Proactive Chat Sales

Supervisors

Agents

Origination DNs

Options

Permissions

Dependencies

Capacity Table

Quota Table

Cost Contract

Site

Script

Tenant

### Configure a blended agent group

The next image shows a *blended* Agent Group—that is, an agent group that can handle more than one media type. The **Proactive Blended** agent group supports two virtual queues:

- **Proactive Chat**, which is part of the **Proactive Chat - chat** pacing target
- **Proactive Voice**, which is part of the **Proactive Voice - voice** pacing target

Home > Agent Groups > Agent Groups > Proactive Blended Properties

Delete

**General**

Supervisors

Agents

Origination DNs

Options

Permissions

Dependencies

**Origination DNs**

	Number	Type	Switch	Alias
<input type="checkbox"/>	Proactive Chat	Virtual Queue	MultiMediaS...	Proactive Chat
<input type="checkbox"/>	Proactive Voice	Virtual Queue	MultiMediaS...	

## Pacing service

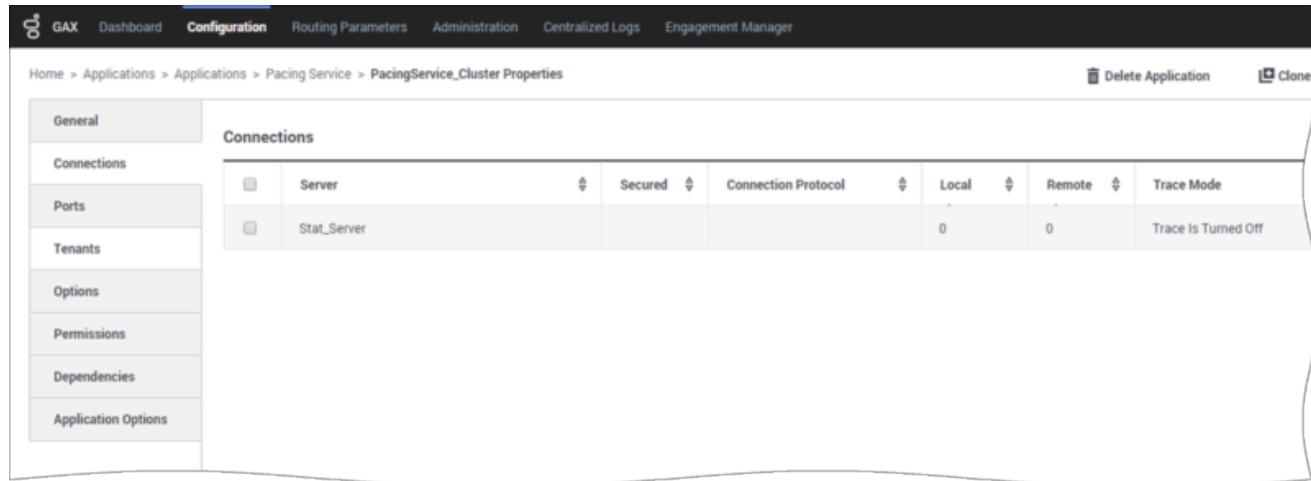
To install Agent Pacing Service, you need:

- A dedicated Genesys application with a type of **Application Cluster**
- A set of applications with a type of **Genesys Generic Server**, with one application for each instance of Pacing Service Server.

The Application Cluster application must include:

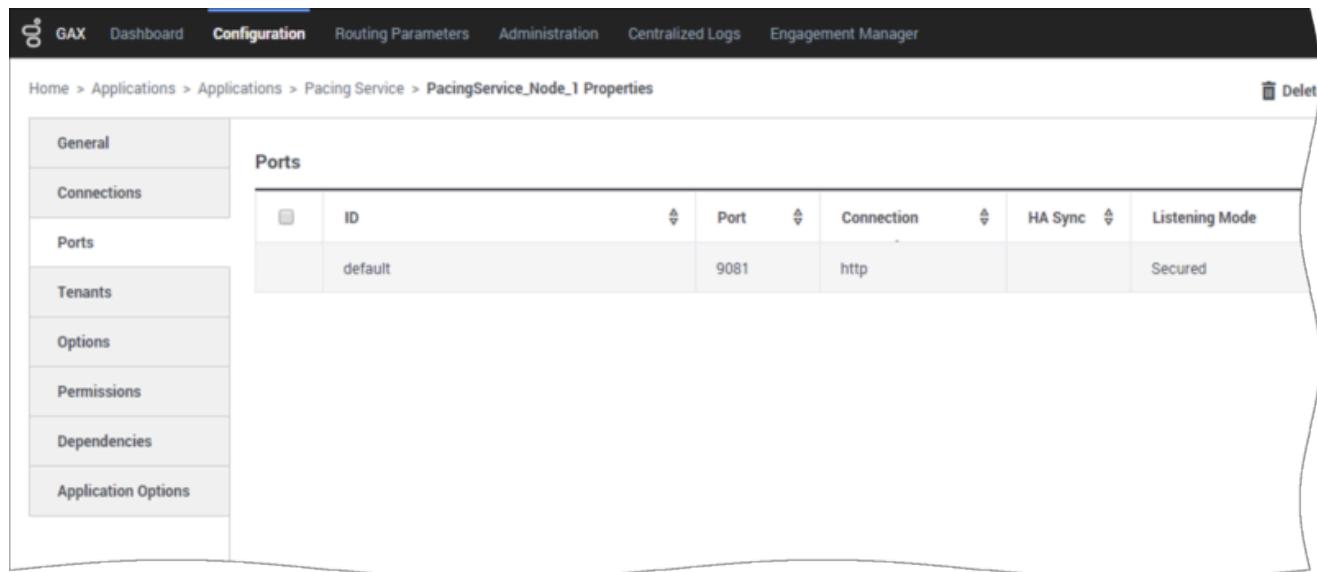
- Descriptions of all of the options (which are inherited by the nodes)
- Connections to the **Stat Server** and (optionally) **Message Server** applications.

The node-related applications must specify the hosts and ports for their server instances and for their connections to the cluster application. The following screenshot shows some of the setup for configuring a cluster and two nodes.



The screenshot shows the GAX Configuration interface. The top navigation bar includes 'GAX', 'Dashboard', 'Configuration' (which is selected and highlighted in blue), 'Routing Parameters', 'Administration', 'Centralized Logs', and 'Engagement Manager'. Below the navigation is a breadcrumb trail: 'Home > Applications > Applications > Pacing Service > PacingService\_Cluster Properties'. On the right side of the header are 'Delete Application' and 'Clone' buttons. The main content area has a left sidebar with tabs: 'General' (selected), 'Connections', 'Ports', 'Tenants', 'Options', 'Permissions', 'Dependencies', and 'Application Options'. The 'Connections' tab is active and displays a table titled 'Connections'. The table has columns: 'Server' (checkbox), 'Secured' (checkbox), 'Connection Protocol' (checkbox), 'Local' (checkbox), 'Remote' (checkbox), and 'Trace Mode' (checkbox). One row is present, showing 'Stat\_Server' in the 'Server' column and '0' in the 'Local' and 'Remote' columns. A note 'Trace Is Turned Off' is displayed next to the 'Trace Mode' column.

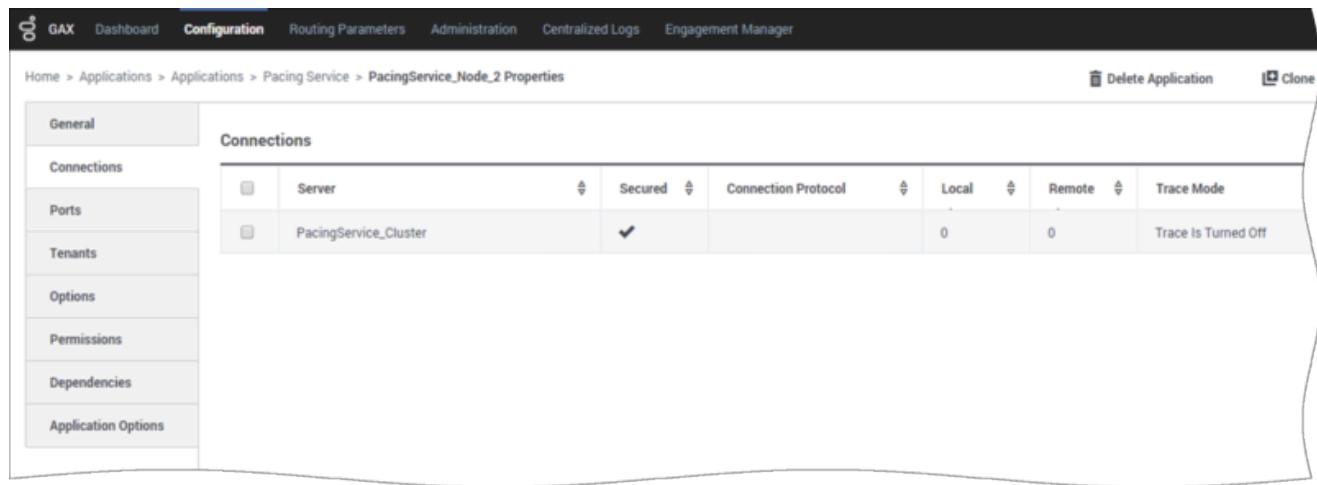
## Configuration



Home > Applications > Applications > Pacing Service > PacingService\_Node\_1 Properties

Ports

ID	Port	Connection	HA Sync	Listening Mode
default	9081	http		Secured



Home > Applications > Applications > Pacing Service > PacingService\_Node\_2 Properties

Connections

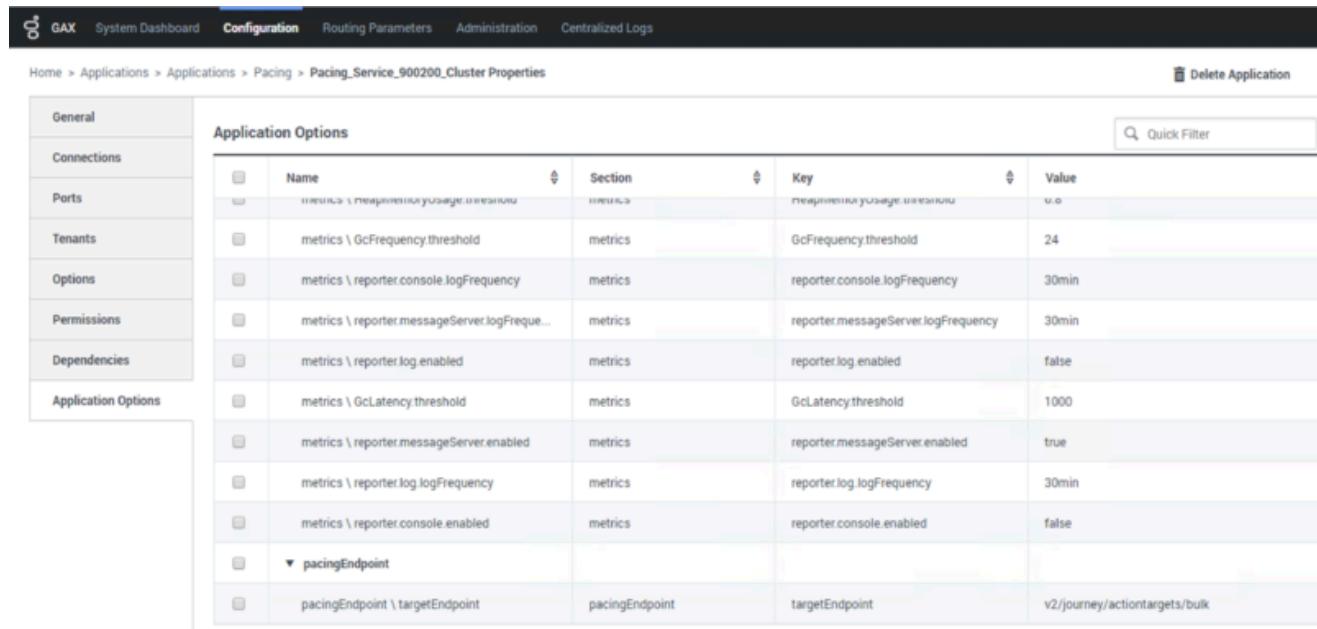
Server	Secured	Connection Protocol	Local	Remote	Trace Mode
PacingService_Cluster	✓		0	0	Trace Is Turned Off

The cluster application specifies the following options:

- log Section—Behavior of the logging subsystem
- metrics Section—Metrics produced by the nodes
- pacing Section—Pacing-related configuration, including the optimization goal
- pacingEndpoint Section—Genesys Predictive Engagement endpoint configuration that specifies the URI path where Genesys Predictive Engagement listens to pacing REST requests. This path will be combined with the base URL specified in the transaction object hybrid\_integration
- forward-proxy Section—Connection options for a forward proxy

## Configuration

---



The screenshot shows the GAX Configuration interface. The top navigation bar includes links for GAX, System Dashboard, Configuration (which is the active tab), Routing Parameters, Administration, and Centralized Logs. The breadcrumb navigation path is Home > Applications > Applications > Pacing > Pacing\_Service\_900200\_Cluster Properties. On the right side of the header, there is a 'Delete Application' button and a 'Quick Filter' search bar. The main content area is titled 'Application Options' and contains a table with the following data:

	Name	Section	Key	Value
	metrics \ measurementUsage.ureasonu	metrics	measurementUsage.ureasonu	v.0
	metrics \ GcFrequency.threshold	metrics	GcFrequency.threshold	24
	metrics \ reporter.console.logFrequency	metrics	reporter.console.logFrequency	30min
	metrics \ reporter.messageServer.logFreque...	metrics	reporter.messageServer.logFrequency	30min
	metrics \ reporter.log.enabled	metrics	reporter.log.enabled	false
	metrics \ GcLatency.threshold	metrics	GcLatency.threshold	1000
	metrics \ reporter.messageServer.enabled	metrics	reporter.messageServer.enabled	true
	metrics \ reporter.log.logFrequency	metrics	reporter.log.logFrequency	30min
	metrics \ reporter.console.enabled	metrics	reporter.console.enabled	false
	▼ pacingEndpoint	pacingEndpoint	targetEndpoint	v2/journey/actiontargets/bulk
	pacingEndpoint \ targetEndpoint	pacingEndpoint	targetEndpoint	v2/journey/actiontargets/bulk

# Troubleshooting

## Contents

- 1 About troubleshooting
- 2 Abandoned statistics are not calculated (always 0)
- 3 Interactions are not delivered to the groups specified in the pacing target

Find solutions to some of the common problems with Agent Pacing Service.

## About troubleshooting

Common problems with Agent Pacing Service include:

- Abandoned statistics are not calculated (always 0)
- Interactions are not delivered to the groups specified in the pacing target

### Abandoned statistics are not calculated (always 0)

When you configure virtual queue objects, you must ensure that the value of the **Alias** parameter equals the value of the **Number** parameter. For more information, see *Virtual queue in Configuration*.

### Interactions are not delivered to the groups specified in the pacing target

Verify that your routing strategy is able to consume the pacing target and virtual queue specified in the **User Data** of your proactively created media interactions.