



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.

Genesys Pulse Deployment Guide

Deploying Pulse Collector and Pulse

Deploying Pulse Collector and Pulse

Important

- This Deployment Guide provides instructions for a new installation of Pulse. To migrate from an earlier version of Pulse, start with the [Pulse 8.5 Deployment Procedure](#).
- In these procedures, use the instructions provided in [Genesys Administrator Extension \(GAX\) Help](#) or the [Framework 8.x Deployment Guide](#), and add the object-specific configuration requirements listed here.

1. Confirm Software Requirements.

Confirm Software Requirements

You deploy Pulse as a GAX plug-in on a web application server to be accessed using a web browser through the GAX user interface. No additional Genesys software is needed on the client machine.

The following are prerequisites for Pulse deployment:

- GAX release 8.5.000.68 or higher must be installed and configured. See the [GAX Deployment Guide](#) for details.
- Tomcat is no longer a prerequisite to use GAX. For those who choose to use Tomcat instead of Jetty, refer to the [GAX Deployment Guide](#) for additional information.
- Genesys DB Server must be release 8.1.300.05 or higher. Refer to the [Framework 8.0 DB Server User's Guide](#) for details.
- Genesys Deployment Agent (GDA) must be installed on the computer on which you install Pulse and Pulse Collector if you plan to install Pulse by using GAX. GDA is required to deploy solution definitions and IPs through GAX. See the [GAX Deployment Guide](#) for details.
- Your Relational Database Management System (RDBMS) must be up and running. This release of Pulse supports the following relational database platforms:
 - Microsoft SQL Server 2008, 2012, 2012 Cluster
 - Oracle 11g, 12c, 12c RAC
 - PostgreSQL 9.0
- The computer on which you install Pulse must be running one of the following:
 - Windows Server 2008 R2 (64-bit), 2012 (64-bit), 2012 Server Hyper-V
 - Red Hat Enterprise Linux AS 6.x (64-bit), with Updates from RHN enabled

- If you use Red Hat Enterprise Linux AS, you must also install the following libraries to ensure backward compatibility for applications: compat-db
 - compat-expat1
 - compat-glibc
 - compat-libf2c-34
 - compat-libgcc-296
 - compat-libgfortran-41
 - compat-libstdc++-295
 - compat-libstdc++-296
 - compat-libstdc++-33
 - compat-libtermcap
 - compat-openldap
 - openssl098e

For more information, see linuxtopia.org.

- If you use Windows, you must also install Microsoft Visual C++ 2010 Redistributable Package (x86 for Collector x86 and x64 for Collector x64).
- Although not required for deployment, you must have Stat Server release 8.1.2 or higher installed for basic operation.

2. Prepare the Pulse Database.

Prepare the Pulse Database

In High Availability (HA) configurations, configure both Pulse applications to use the same database.

Microsoft SQL Server

1. Create a new empty Microsoft SQL Server database.
2. Create a new Microsoft SQL Server user account.
3. Set the new database as default database for the user.
4. Grant the new user sufficient privileges for the new database. User must be able to create database objects and select, insert, update, and delete data in tables.
5. Run the statement below for the Pulse database (DB). Replace <Pulse DB> with the name of the Pulse DB and make sure that there are no other connections to the Pulse DB when you run this statement:

```
ALTER DATABASE <Pulse DB> SET READ_COMMITTED_SNAPSHOT ON;
```

Oracle

1. Create a new user/schema to be used by Pulse. The user must have RESOURCE and CREATE VIEW privileges.

PostgreSQL

1. Create a new empty PostgreSQL database.
2. Create a new PostgreSQL user account.
3. Set the new database as default database for the user.
4. Grant the new user sufficient privileges for the new database. User must be able to create database objects and select, insert, update, and delete data in tables.

3. Deploy Pulse Collector.

Deploying Pulse Collector

To deploy Pulse Collector, which connects directly to Stat Server to collect statistics for contact center objects. Pulse Collector also connects to DB Server, through which it accesses the database where reporting layouts are stored.

1. Upload Pulse Collector Installation Package and Template:
 - a. In GAX, go to **Administration > Installation Packages** and click on the plus sign.
 - b. Select **Installation Package Upload (template uploaded separately)** and click **Next**.
 - c. For **Upload a Package**, select the zipped file that contains the Pulse Collector Installation Package (Pulse Collector IP). The zip file should have in its root the files from the IP folder (such as ip_description.xml and read_me.html).
 - d. For **Upload an XML template**, select the XML Template file (Collector.xml from the Templates Installation Package directory).
 - e. For **Upload an APD template**, select the APD Template file (Collector.apd from Templates Installation Package directory).
 - f. Click **Finish**.
2. Deploy the Pulse Collector Installation Package and Template:
 - a. Click on the Pulse Collector Installation Package to open the Properties tab.

Important

The Pulse Collector Installation Package status should be Complete.

- b. Click on the related icon and choose **Install** to open the IP Deployment Wizard.
- c. Fill required fields and finish the installation.

Important

The InstallPath should point to an empty folder where you plan to install Pulse Collector.

Tip

You can also install Pulse Collector directly on the server by executing the Pulse Collector installation procedure from the Pulse Collector installation package, then uploading the Pulse Collector template (Collector.apd) using Configuration Manager.

3. Optional: To change the default options for Pulse Collector in GAX, open the Pulse Collector Application object modify the values of configuration options described in [Pulse Collector Application Object](#).
 4. For a high-availability (HA) deployment, repeat step 2 to install the backup instance to separate folder. You need a Pulse Collector for each instance of the Pulse application.
 5. In GAX, add the Pulse Collector Application object to the connections of your GAX Application object.
 6. For a load-balanced environment configuration with two GAX applications and Pulse plug-ins, associate one GAX with the primary Pulse Collector and the second GAX with the backup Pulse Collector as follows:
 - a. Add the primary Pulse Collector Application object to the connections of the first (or primary) GAX Application object.
 7. Add the backup Pulse Collector Application object to the connections of the second (or backup) GAX Application object.
 8. Add the backup Pulse Collector Application object to the **General** tab of the primary Pulse Collector Application object as a backup server.

Note: Several GAX instances can be connected to Collector. For example if you have 4 load balanced GAX applications you can connect 2 of them to Primary Collector and 2 of them to Backup Collector.
- Create a new Database Access Point (DAP), which is necessary for connectivity to the Pulse database through the DB Server:
 - a. Select the Default connection type.

- b. Specify Database Server Application object in **DB Server** field.
 - c. Specify the connectivity parameters for your RDBMS.
 - d. On the **Ports** tab, change the default port value to the value of your DB Server port.
- In the connections of the Pulse Collector Application, add the DAP that is to be used by Pulse Collector.
 - In the connections of the Pulse Collector Application, add the primary Stat Server application that is to be used by Pulse Collector.
 - Add the Tenant objects to the Tenants tab for all Pulse Collectors that you plan to monitor in Pulse.

4. Deploy Pulse.

Deploying Pulse

Configure the necessary objects required by Pulse using GAX.

1. Upload Pulse Installation Package and Template:
 - a. In GAX, go to **Administration > Installation Packages** and click on the plus sign.
 - b. Select **Installation Package Upload (template uploaded separately)** and click **Next**.

For **Upload a Package**, select the zipped file that contains the Pulse Installation Package (Pulse IP). The zip file should have in its root the files from the IP folder (such as ip_description.xml and read_me.html).
 - c. For **Upload an XML template**, select the XML Template file (Pulse.xml from the Templates Installation Package directory).
 - d. For **Upload an APD template**, select the APD Template file (Pulse.apd from Templates Installation Package directory).
 - e. Click **Finish**.
2. Deploy the Pulse Installation Package and Template:
 - a. Click on the Pulse Installation Package to open the **Properties** tab.

Important

The Pulse Installation Package status should be Complete.

- b. Click on the related icon and choose **Install** to open the IP Deployment Wizard.
- c. Fill required fields and finish the installation.

Important

- For "IPCommon InstallPath", use an empty folder where the Pulse plugin will be installed (for example, C:\123).
- When installing Pulse on Linux, a second path leading to the GAX installation (the field is called GAX directory) must be specified in addition to InstallPath.

Tip

You can also install Pulse directly on the server by executing the Pulse installation procedure from the Pulse installation package.

If the installation procedure fails to find the GAX installation, you can manually copy the Pulse plugin files to enable Pulse in GAX.

After installation, find all jar files in root directory of installed Pulse plugin and copy them to:

Linux: <GAX root>/plug-ins (if the directory exists) and <GAX root>/webapp/WEB-INF/lib

Windows: <GAX root>\plug-ins (if the directory exists) and <GAX root>\webapp\WEB-INF\lib.

3. Optional: Configure the [\[Pulse\] options](#) on **Application Options** tab for the GAX Application object:

Important

During startup, GAX looks for all options that are required for Pulse operation and adds them to the GAX Application object if they are not explicitly configured. If a required option is either not configured or specifies an invalid option value, GAX uses the option's default value.

4. Create a new Database Access Point, which is necessary for connectivity to the Pulse database:
 - a. Enter a **Database Name**.
 - b. On the **General** tab, set the **Connection Type** to **JDBC** and specify the following field values:
 - Role = Main

- Debug = false
- JDBC Query Timeout: 15
- Case Conversion: any

- c. On the **Ports** tab, change the value of the default port to the value of your RDBMS port.
- d. On the **Application Options** tab, add the role configuration option with its value set to pulse in the GAX section to identify this Database Access Point as the database schema created for Pulse.
- e. For a PostgreSQL database, add the `postgre_71_compatible` configuration option with its value set to false in the GAX section on the **Application Options** tab.

Refer to the [Framework 8.0 DB Server User's Guide](#) for additional information about configuring this application.

- Add the Database Access Point to the connections of your GAX Application object.
- Navigate to **Configuration > Environment > Application**, select the GAX Application, **Permissions** tab and configure the SYSTEM account to have Read, Update, and Execute permissions.
- For a High Availability (HA) deployment,
 - a. Complete Steps 2-6, for each GAX Application object to be used with the Pulse plug-in.
 - b. Each GAX application should have the other GAX application in its connections and there must be no relation between the GAX applications as Backup Servers.
 - c. Navigate to GAX application object properties and select Application options tab to create a new [pulse] section.
 - d. Optionally, set the `access_url option` in the [pulse] Section on the **Application Options** tab of the GAX Application object for each instance of GAX.

Important

Both Pulse instances must use the same database.

- From the user account created when you prepared the Pulse database, execute the SQL statements in the appropriate initialization script deployed during installation (scripts folder)—either:
 - `pulse_init_mssql.sql`
 - `pulse_init_oracle.sql`
 - `pulse_init_postgres.sql`
- Restart GAX. See the [GAX Deployment Guide](#) for information about how to start GAX.

5. Optional: Deploy RabbitMQ.

Optional: Deploying RabbitMQ for Quick Widget Updates

Use RabbitMQ for Quick Widget Updates.

1. To use RabbitMQ to work with Pulse, use the following software versions:

- RabbitMQ server version 3.3.5.
- Use identical versions of Erlang on all hosts running RabbitMQ:
 - Windows, use Erlang version OTP 17.3
 - Linux, use the latest Erlang version available (R14B-04 or newer)

2. Determine whether to use RabbitMQ with Pulse using Cluster or Single-Node configurations:

- **Cluster configuration**—Use at least the same number of RabbitMQ instances as the number of Pulse Collector applications. RabbitMQ can run on any host: either the one where Pulse runs or any other host accessible over reliable network.

The default configuration should be one RabbitMQ instance running on every host where Pulse Collector runs. For example, Primary host (host1) and Backup host (host2).

- **Single node configuration**—This simple configuration uses a single RabbitMQ instance running either on a host with Pulse Collector or any other host accessible over reliable network.

Note: If this RabbitMQ instance fails or the whole host fails, Quick Widget Updates stop working. If you choose this configuration, you still need to go through all steps to deploy RabbitMQ unless clearly stated otherwise. The host with RabbitMQ is called host1 in the remainder of this deployment.

3. On every host, install Erlang:

- On Windows Server 2008/2012:
 - i. Download [Erlang OTP 17.3 Windows installer](#). For Windows x64, use the 64-bit version.
 - ii. Run installer.
 - iii. Go through the installation wizard and install Erlang.

- On Linux, as the **root** user, run the following commands:

i. To enable **EPEL**:

```
wget http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
rpm -ivh epel-release-6-8.noarch.rpm
```

ii. To install Erlang:

```
yum install erlang
```

4. On every host, install and configure RabbitMQ:

a. Download and install RabbitMQ:

- On Windows Server 2008/2012:
 - i. Download [RabbitMQ installer](#)
 - ii. Run the installer.
 - iii. Go through the installation wizard and install RabbitMQ.
- On Linux, as the **root** user, issue the following commands:
 - i. Download RabbitMQ package:

```
wget https://www.rabbitmq.com/releases/rabbitmq-server/v3.3.5/rabbitmq-server-3.3.5-1.noarch.rpm
```

ii. Import RabbitMQ key:

```
rpm --import http://www.rabbitmq.com/rabbitmq-signing-key-public.asc
```

iii. Install RabbitMQ server:

```
yum install rabbitmq-server-3.3.5-1.noarch.rpm
```

b. On every host, first start RabbitMQ server to generate Erlang cookie file.

- On Windows, start RabbitMQ service. If it is already started proceed to the next step.
- On Linux, from the command line as the **root** user:

i. Run: `service rabbitmq-server start`

ii. If you want RabbitMQ to start automatically when system boots, run:

```
chkconfig rabbitmq-server on
```

Important

To run the **rabbitmqctl** tool that is installed as a part of RabbitMQ server.

- On Linux, run it as **root** right after installing the server.
- On Windows, navigate to `${RabbitMQ installation directory}\rabbitmq_server-3.3.5\sbin` directory and run **rabbitmqctl** from there.

Important

When the entire cluster is brought down, the last node to go down must be the first node to be brought online. If this doesn't happen, the nodes will wait 30 seconds for the last disc node to come back online, and fail afterwards.

[-rabbitmq.com](https://www.rabbitmq.com)

When node fails due to the above limitation RabbitMQ's `startup_log` contains message like this:

```
BOOT FAILED
=====
Error description:
  {could_not_start,rabbit,
    {bad_return,
      {{rabbit,start,[normal,[]]},
       {'EXIT',
        {rabbit,failure_during_boot,
          {error,
```

```
{timeout_waiting_for_tables,
```

C. (Cluster configuration only) Reset RabbitMQ on every RabbitMQ host using the following commands:

```
rabbitmqctl stop_app  
rabbitmqctl reset
```

d. (Cluster configuration only) Stop RabbitMQ server on every host. You must stop RabbitMQ on both servers before starting any one of them. On both hosts, do the following:

- On Windows, stop RabbitMQ service.
- On Linux, as the **root** user, run:

```
service rabbitmq-server stop
```

e. (Cluster configuration only) Copy Erlang cookie from one host to all others. These files must be identical on all hosts that form RabbitMQ cluster.

- On Windows: Copy to both C:\Users\<user account under which Erlang runs>.erlang.cookie and C:\Windows\erlang.cookie
- On Linux: Copy to /var/lib/rabbitmq/.erlang.cookie

f. (Cluster configuration only) Configure RabbitMQ cluster consisting of several RabbitMQ instances running on host1, host2, ..., hostn.

i. Create the rabbitmq.config file on all of those hosts:

- On Windows, in %APPDATA%\RabbitMQ\.
- On Linux, in /etc/rabbitmq.

This how configuration might look for a 2-node cluster:

```
[  
  {rabbit,  
    [  
      {cluster_nodes, [['rabbit@host1', 'rabbit@host2'], disc]}  
    ]  
  }  
].
```

Change host1 and host2 with the names of your hosts.

Important

Hostnames are case sensitive. You must use uppercase characters for Windows hosts and lowercase characters for Linux hosts.

- ii. On Windows only: In `${RabbitMQ installation directory}\rabbitmq_server-3.3.5\sbin` run the following command: `rabbitmq-service.bat install`
- g. (Cluster configuration only) Start RabbitMQ server on all hosts.
 - On Windows, start RabbitMQ service.
 - On Linux, as the **root** user run:


```
service rabbitmq-server start
```
- h. (Cluster configuration only) Confirm that RabbitMQ instances have formed a cluster. Run:


```
rabbitmqctl cluster_status
```

The output should contain the following line for the cluster consisting of two nodes:

```
running_nodes,['rabbit@host1','rabbit@host2']
```
- i. Create a vhost with a name `/pulse` for Pulse. If you create a vhost with a different name, you must specify it in Pulse Collector configuration options. On any of the hosts run the following command:


```
rabbitmqctl add_vhost /pulse
```
- j. Create a user for Pulse with name **'pulse'** and password **'pulse'**. If you create a user with a different name and password, you must specify them in Pulse Collector configuration options. On any of the hosts run the following command:


```
rabbitmqctl add_user pulse pulse
```
- k. Grant user access to vhost. Here is how to grant user **'pulse'** access to vhost **'/pulse'** with permissions to create exchanges with names starting with **'pulse'**. On any of the hosts run the following command:


```
rabbitmqctl set_permissions -p /pulse pulse "^pulse.*" ".*" ".*"
```
- 5. To configure Pulse Collector to work with RabbitMQ cluster you need to have `transport-rabbitmq` section configured in options of Pulse Collector application object. The following options should be added to that section

option	value	default value	comment
disabled	no	no	You can completely remove this option from the section.
nodes	host1:5672,host2:5672	localhost:5672	List a single host name with a port for single node configuration, and all cluster nodes host names and ports for cluster configuration.

option	value	default value	comment
			Note: If Pulse Backed and Pulse Collector it connects to run on one of the hosts in the list, put this host as the first entry in the list.
exchange-prefix	pulse.collector	pulse.collector	The value of this field should be compatible with the permissions given to a user during configuration. The default value is compatible with the values used in this guide.
vhost	/pulse	/pulse	If needed, change the virtual host name to the one used during the RabbitMQ configuration.
username	pulse	pulse	If needed, change the user name to the one used during the RabbitMQ configuration.
password	pulse	pulse	If needed, change the password to the one used during the RabbitMQ configuration.
max-queue-length	1000	1000	Read the section on RabbitMQ memory usage for details.

6. Restart Pulse Collector and GAX to apply changes.

Important

RabbitMQ memory and disc usage

RabbitMQ instances should not store any Pulse application data on disc, so the disc usage is insignificant unless the message queue for any of the Pulse applications grows too large. To ensure that the message queue does not grow too big in some exceptional cases there is a limit on queue length in Pulse, which is controlled by option `max-queue-length` in section `transport-rabbitmq` of Pulse Collector application options.

Use the value of this option to estimate possible RabbitMQ memory usage. To roughly estimate upper limit of possible memory usage use this formula:

$$\text{<Max memory usage>} = \text{<RabbitMQ idle memory usage>} + 3 * (\text{max-queue-length} * \text{<Average size of a delta snapshot>} * 4)$$

For example, for an average change in message size of 10KB, RabbitMQ idle memory usage of 100MB, and `max-queue-length` of 1000, we obtain 220MB of memory usage.

6. Configure the Stat Server Application object.

Configuring the Stat Server **Application** object

Important

Stat Server requires Java Environment configuration for several templates to function properly. See the [Stat Server Deployment Guide](#) for details. Without this additional configuration, statistics in some Queue templates (Email Queue Activity, eServices Queue KPIs, IWD Queue Activity) will not work.

1. Set the `accept-clients-in-backup-mode` option in the `[statsserver]` section on **Application Options** tab to `yes` for both the primary and backup Stat Server Application objects.

Important

This option is required even if there is no backup application specified for the Stat Server application.

2. Update Stat Types specified in the pulse_statistics.cfg file within the scripts folder in Pulse installation to use the particular social media that is configured in your eServices solution (facebook is used in default file version). See [eServices documentation](#) for more details.
3. You must import pulse_statistics.cfg file to both the primary and backup Stat Server Application objects to create the statypes that Pulse should monitor. To monitor the file:
 - a. Click **More** on the **Application Options** tab.
 - b. Select **Import**, uncheck **Override**, and browse the file.

Important

- To calculate the **% Ready Time** in the Queue KPIs template, set the queue-use-pseudo-actions option in the [statsserver] section of Stat Server Application object to false.
- Some Stat Server filters used in Pulse templates rely on certain user data or reasons attached to the call (for example, VoiceCall_No_Wait, ReasonLunch, and others that have PairExists in their definition).

You may need to adjust definitions of these filters to use Attached Data or Reasons according to your environment or adjust your routing strategies or desktop application to attach data used by those statistics. Otherwise, statistics that rely on these filters will show 0 (zero).

4. Restart both Stat Server applications.

7. Configure user access.

Configuring User Access

1. In GAX, navigate to **Configuration > Accounts > Roles** and create a new Role object to provide access to Pulse functionality.

Important

When creating a new Role, a dialog box with Role Template Selection appears. Do not select a template (leave the selection empty). Press **OK**.

- a. Define the privileges granted by the Role on the **Assigned Privileges** tab in the Pulse section.

[+] **Privilege Details**

Privilege Details:

- **Pulse View Dashboard**—View dashboard.

- **Pulse Edit Widget Display**—Edit widget display options.
- **Pulse Manage Widget**—Remove, create, clone, and edit widgets.
- **Pulse Manage Templates and Default Dashboard**—Remove, create, clone, and edit custom templates. Change the default dashboard.

The following privileges are for users or third-party applications that connect to Pulse using a Web API.

- **Pulse Read All Layouts**—View all Pulse layouts using a Web API.
- **Pulse Write Snapshot**—Upload layout snapshots using a Web API.

Important

Users without permission to edit their own Person object cannot save their Custom dashboard configuration. Users without permission to edit both their own Person object and their Tenant object cannot save the Default dashboard configuration. For example, such a user can make changes, such as creating widgets, but cannot save the changes. The next time the page is loaded, the changes are lost.

Pulse privileges use GAX logic, so the high-level privileges do not include lower-level privileges. For example, to configure role with full control access, you have to assign all privileges to it.

For role members to create Widgets, you must assign **Pulse View Dashboard** and **Pulse Edit Widget Display** in addition to **Pulse Manage Widgets**.

Important

You must assign at least the **Pulse View Dashboard** privilege to each Role object.

- b. Assign the Role to Persons and Access Groups in the Role Members section as required.
2. Update the GAX User to have the following permissions:
 - Read and execute permissions on the client application (usually called default)
 - Read and execute permissions on Tenant
 - Read and update permission on their own Person object

Important

Users without permission to edit their own Person object cannot save their Custom dashboard configuration. Users without

permission to edit both their own Person object and their Tenant object cannot save the Default dashboard configuration. For example, such a user can make changes, such as creating widgets, but cannot save the changes. The next time the page is loaded, the changes are lost.

- a. Provide the user permission to modify the tenant configuration object.
 - i. Select Configuration.
 - ii. From the Environment pane, choose Tenants.
 - iii. Click to open the required Tenant to view its properties.
 - iv. Select Permissions tab.
 - v. Add the Person or Access group containing this user.
 - vi. Add Update permission in addition to Read and Execute already required to access GAX.
- b. Provide the user permission to modify the user's own configuration object:
 - i. Select Configuration.
 - ii. From the Accounts pane, choose Persons.
 - iii. Click to open the required Person's properties.
 - iv. Select Permissions tab.
 - v. Add the Person or Access group containing this user.
 - vi. Add set Update permission.