# Genesys Data Processing Server Deployment Guide

Universal Contact Server 9.0.0

1/21/2022

# Table of Contents

# Genesys Data Processing Server Deployment Guide

Genesys Data Processing Server (GDPS) processes the complex, high-volume data produced by select Genesys products for a variety of uses.

## Before you begin

Because GDPS must process a lot of data, it needs to run on top of a high-speed—and highly scalable—cluster computing system. Genesys uses Apache Spark for this task.

Spark supports several types of clustering. GDPS works well with the simplest one, Spark Standalone Mode. This mode provides high availability by using a dedicated master node. A typical cluster deployment will consist of one master node and several worker nodes and is usually started by GDPS in the background.

Genesys recommends that you configure GDPS in Genesys Administrator, defining a single Application Cluster object and the appropriate number of individual node objects. You can configure these objects and their options using the procedures provided on the rest of this page.

### GDPS nodes

The Spark cluster consists of one master node and several worker nodes. Any GDPS node can be the master node in the cluster, as this role is defined by the value of the Spark startMode option in the node's configuration options. Here is more information about the two types of node:

- The **Master** node is represented by a Spark startMode of `both`, which indicates that both a Spark Master and a Spark Worker instance will be started on the node. Please note that the Spark host option should be set in agreement with startMode so that the hostname used in the Spark host setting belongs to the node that runs the Spark Master instance. To avoid problems with connectivity inside the Spark cluster, this hostname should be the primary one in the network configuration for this host. In other words, Java's **InetAddress.getLocalHost** should return the same value for the GDPS Master node.
- The **Worker** node is represented by a startMode of `worker`. Only a Spark worker instance will be launched at this node.

There is one additional mode available for the Spark startMode option. A mode of `off` means that no Spark processes will be launched on this host and that the role of the node in the GDPS cluster is undefined. This mode is for use in situations where you want to have an externally managed Spark cluster, and limits you to one GDPS node, which serves as an entry point for the Spark cluster. You cannot deploy GDPS with multiple nodes if you have set **startMode** to `off`. Also, if you use this mode, you must have an advanced understanding of how to work with and manage a Spark cluster.

## Configuring GDPS

We have included information about GDPS–related configuration options at the end of this page.

## Deploying GDPS

To deploy GDPS, follow these steps:

1. Importing the GDPS cluster template
2. Creating the cluster application
3. Configuring the cluster application
4. Importing the GDPS template
5. Creating a node application
6. Configuring a node application
7. Adding nodes to a cluster
8. Installing GDPS

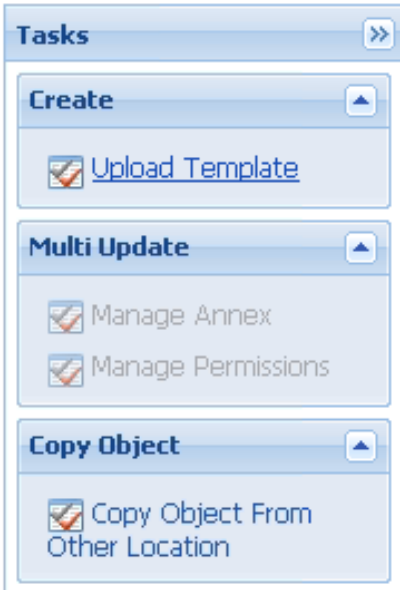
**Note:** For more information on how to work with templates and application objects in Genesys Administrator, consult Generic Configuration Procedures.

## Importing the GDPS cluster template

**Note:** For more information on how to work with templates in Genesys Administrator, consult Generic Configuration Procedures.

**Start**

1. Open Genesys Administrator and navigate to **Provisioning > Environment > Application Templates**.
2. In the **Tasks** panel, click **Upload Template**.




3. In the **Click 'Add' and choose application template (APD) file to import** window, click **Add**.
4. Browse to the **Data_Processing_Cluster.apd** file. The **New Application Template** panel opens.
5. Click **Save & Close**.

**End**

## Creating the cluster application

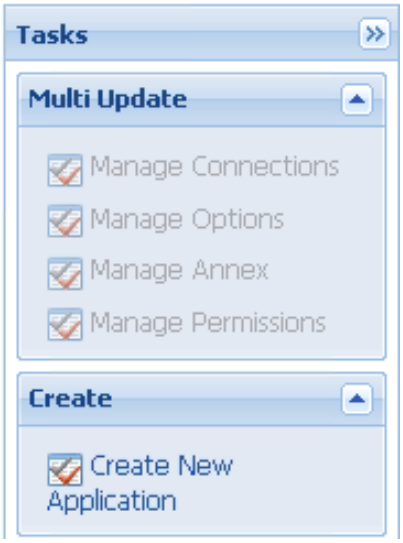
**Note:** For more information on how to work with application objects in Genesys Administrator, consult Generic Configuration Procedures.

### Prerequisites

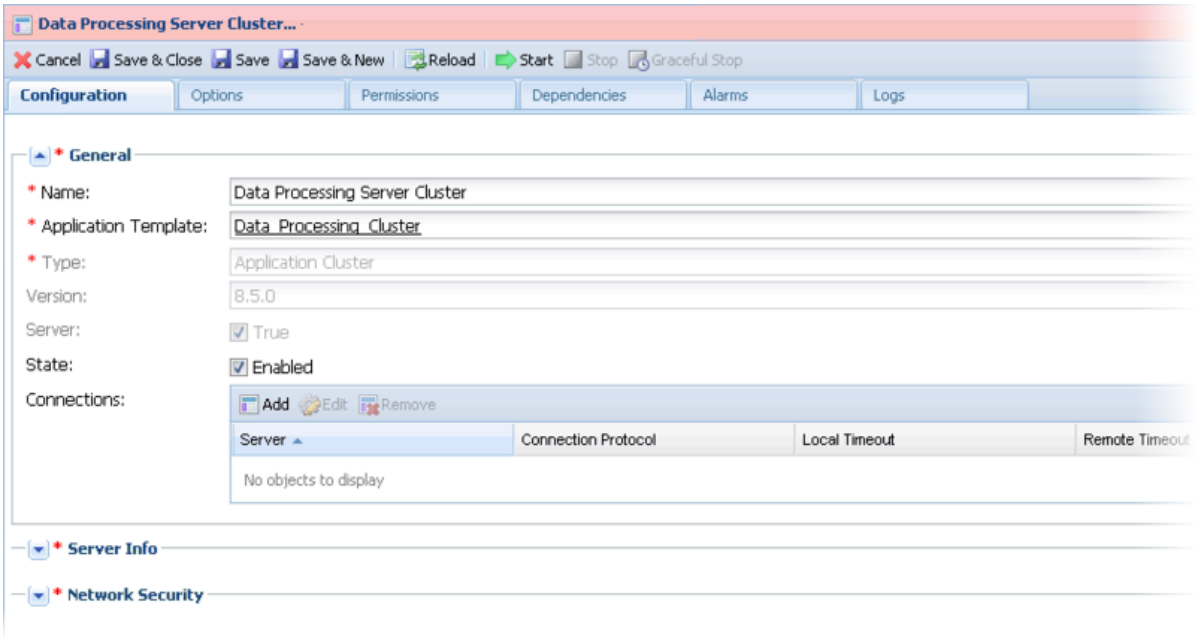
- You have completed Importing the GDPS cluster template.

### Start

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




3. In the **Select Application Template** panel, click **Browse for Template** and select the GDPS cluster template that you imported in Importing the GDPS cluster template. Click **OK**.
4. The template is added to the **Select Application Template** panel. Click **Next**.
5. In the **Select Metadata file** panel, click **Browse** and select the **Data_Processing_Cluster.xml** file. Click **Open**.
6. The metadata file is added to the **Select Metadata file** panel. Click **Next**.
7. In the **Specify Application parameters** tab:
   - Enter a name for your application. For instance, `Data_Processing_Server_Cluster`.
   - Make sure **State** is enabled.
   - Select the **Host** on which the GDPS cluster will reside.
   - Click **Create**.
8. The **Results** panel opens.
9. Enable **Opens the Application details form after clicking 'Finish'** and click **Finish**. The GDPS cluster application form opens and you can start configuring the GDPS cluster application.




**End**

## Configuring the cluster application

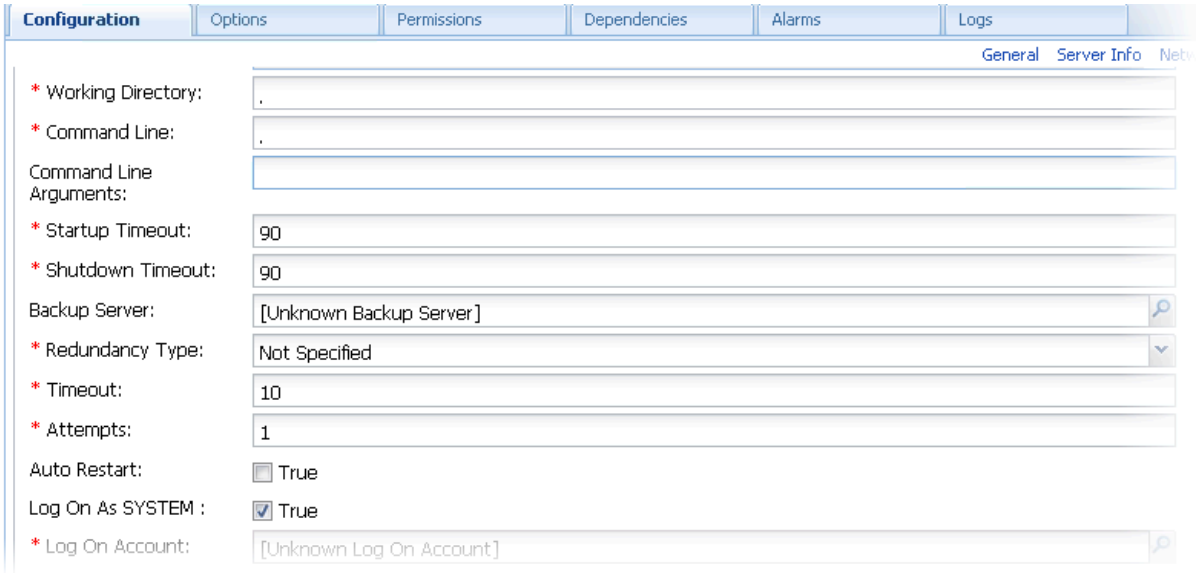
**Note:** For more information on how to work with application objects in Genesys Administrator, consult Generic Configuration Procedures.

### Prerequisites

- You completed Creating the cluster application.

### Start

1. If your Cluster application form is not open in Genesys Administrator, navigate to **Provisioning > Environment > Applications**. Select the application defined for the GDPS cluster and click **Edit...**.
2. Expand the **Server Info** pane.
3. If your **Host** is not defined, click the lookup icon to browse to the hostname of your application.
4. Ensure the **Working Directory** and **Command Line** fields contain "." (period).




5. Click **Save**.
6. In the **Listening Ports** section, create the default port by clicking **Add**. The **Port Info** dialog opens.
   - Enter the **Port**. For instance, `10081`.
   - Choose `http` for the **Connection Protocol**.
   - Click **OK**. The HTTP port with the default identifier appears in the list of **Listening ports**.

**End**

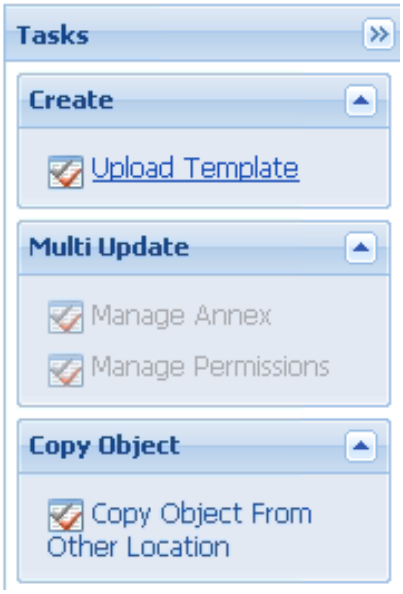
## Importing the GDPS template

### Prerequisites

- You completed Configuring the cluster application.

### Start

1. Open Genesys Administrator and navigate to **Provisioning > Environment > Application Templates**.
2. In the **Tasks** panel, click **Upload Template**.




3. In the **Click 'Add' and choose application template (APD) file to import** window, click **Add**.
4. Browse to the **Data_Processing_Server.apd** file and select it. The **New Application Template** panel opens.
5. Click **Save & Close**.

**End**

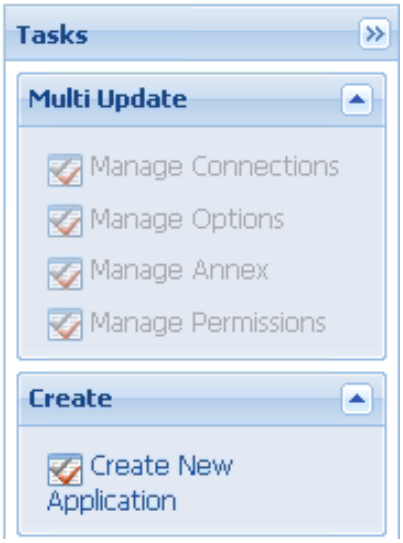
## Creating a node application

### Prerequisites

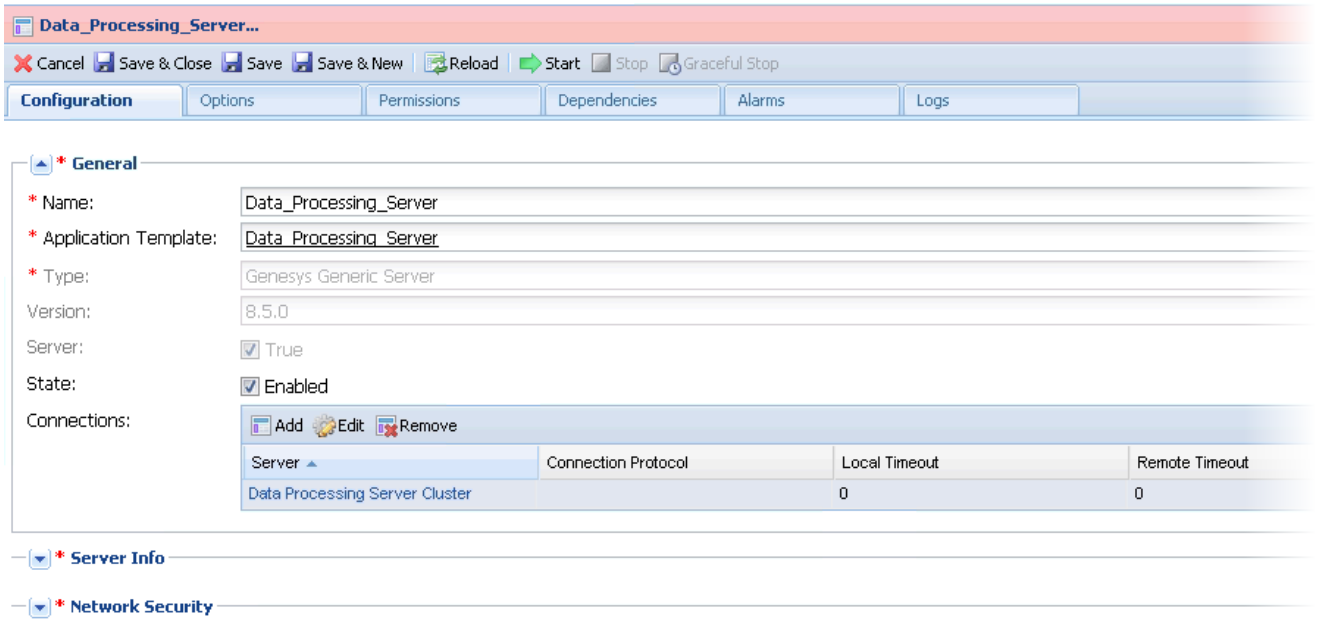
- You completed Importing the GDPS template.

### Start

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




3. In the **Select Application Template** panel, click **Browse for Template** and select the GDPS template that you imported in Importing the GDPS template. Click **OK**.
4. The template is added to the **Select Application Template** panel. Click **Next**.
5. In the **Select Metadata file** panel, click **Browse** and select the **Data_Processing_Server.xml** file. Click **Open**.
6. The metadata file is added to the **Select Metadata file** panel. Click **Next**.
7. In **Specify Application parameters**:
   - Enter a name for your application. For instance, `Data_Processing_Server`.
   - Make sure **State** is enabled.
   - Select the **Host** on which the node will reside.
   - Click **Create**.
8. The **Results** panel opens.
9. Click **Save & Close**. If the **Confirmation** dialog opens, click **Yes**.
10. Enable **Opens the Application details form after clicking 'Finish'** and click **Finish**. The `Data_Processing_Server` application form opens and you can start configuring the node application.




**End**

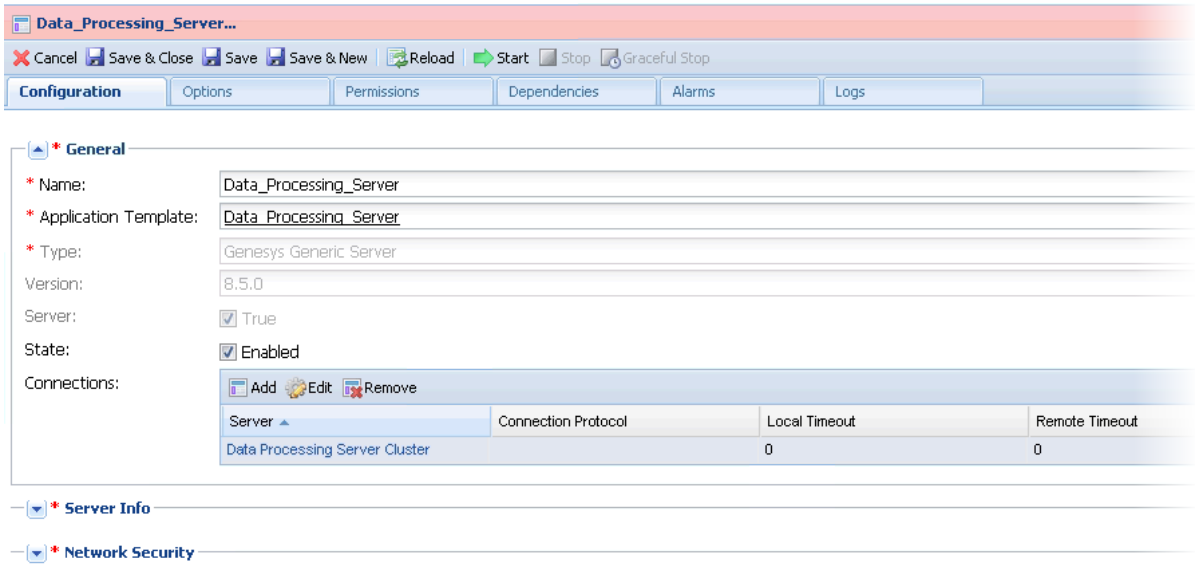
# Configuring a node application

## Prerequisites

- You completed Creating a node application.

## Start

1. If your node application form is not open in Genesys Administrator, navigate to **Provisioning > Environment > Applications**. Select the application defined for the node and click **Edit...**.
2. In the Connections section of the **Configuration** tab, click **Add**. The **Browse for applications** panel opens. Select the GDPS cluster application you defined above, then click **OK**.




3. Expand the **Server Info** pane.
4. If your **Host** is not defined, click the lookup icon to browse to the hostname of your application.
5. In the Listening Ports section, create the default port by clicking **Add**. The **Port Info** dialog opens.
   - Enter the **Port**. For instance, `10081`.
   - Choose `http` for the **Connection Protocol**.
   - Click **OK**. The HTTP port with the default identifier appears in the list of **Listening ports**.
6. Click **Save**.

**End**

## Adding nodes to a cluster

To create more nodes:

**Start**

1. Follow the instructions above for Creating a node application, but use a different name for the new node.
2. Configure the new node application, as shown above, but point to a different port.

**End**

# Installing GDPS

Install the GDPS on Windows or Linux.

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

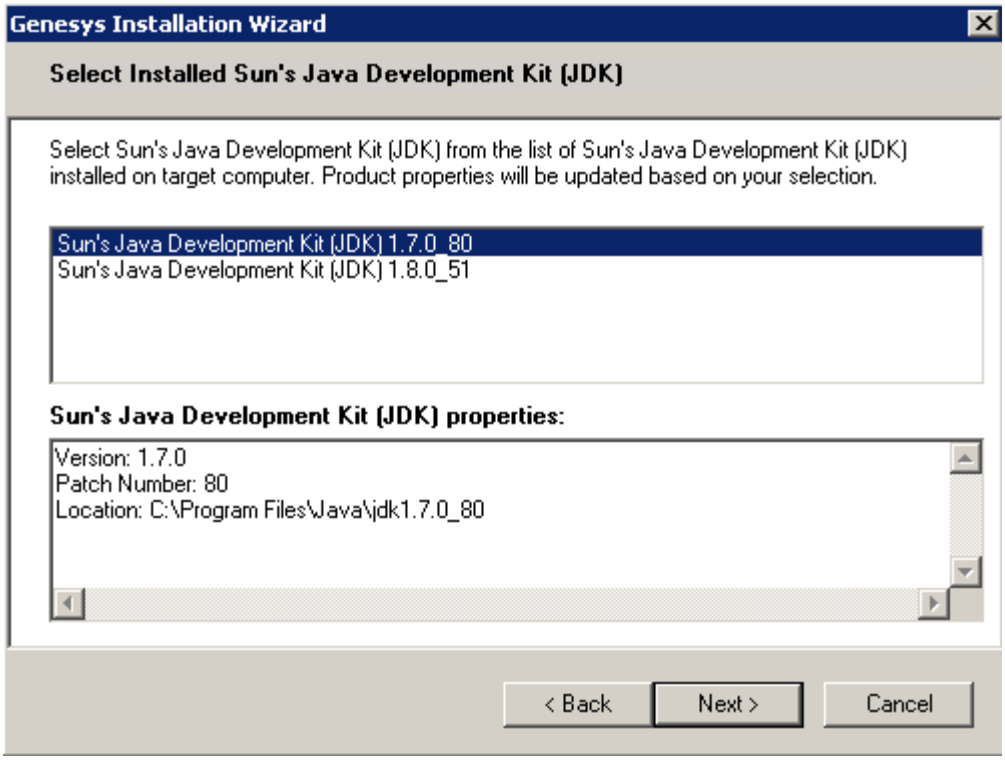
## Windows

### Prerequisites

- Configuring a node application
- A supported JDK is installed.

### Start

1. In your installation package, locate and double-click the **setup.exe** file. The Install Shield 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 on to Configuration Server.
5. Click **Next**. The **Select Application** screen appears.
6. Select the GDPS 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 Argument**s information previously entered in the **Server Info** and **Start Info** tabs of the selected Application object.
   **Note:** For multi-node clusters, you must install the GDPS Application into exactly the same directory on every node. For example, if the path for Node 1 is `/genesys/gdps/gdps_n1`, it cannot be `/genesys/gdps/gdps_n2` for any of the other nodes. This requires manual intervention, since the installation package offers a default installation path based on the application name, which is therefore different for each node.
7. Click **Next**. The **Choose Destination Location** screen appears.
8. Under **Destination Folder**, keep the default value or browse for the desired installation location. Note that you probably do not want to use the Windows Program Files folder as your destination folder.
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. In the **Pulse Collector Configuration** window, ensure that **Use Pulse Collector** is unchecked:
12. Click **Next**. Select the appropriate JDK:




13. Click **Next**. The **Ready to Install** screen appears.
14. Click **Install**. The Genesys Installation Wizard indicates it is performing the requested operation for GDPS. When through, the **Installation Complete** screen appears.
15. Click **Finish** to complete your installation of the GDPS.
16. Inspect the directory tree of your system to make sure that the files have been installed in the location that you intended.

## **End Note:** Genesys recommends that you regularly clear the following:

- The Spark temporary directory. You can find it in the system temporary directory with a name template of `spark-*`. The default location for this directory is ***system_disk*:\Users*user_name*\AppData\Local\Temp directory**. You can also use the system disk clean-up procedure.
- Within the GDPS directory itself, the **<GDPS directory>/spark/work** folder.

# Linux

### Prerequisites

- Configuring a node application
- A supported JDK is installed.

### Start

1. Open the UCS IP, and run the **Install.sh** file. 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 in to Configuration Server:
   - The hostname. For instance, `demosrv.genesyslab.com`.
   - The listening port. For instance, `2020`.
   - The user name. For instance, `demo`.
   - The password.
4. If you have a backup Configuration Server, enter the **Host name** and **Port**.
   If the connection settings are successful, a list of keys and applications is displayed.
5. Enter the key for the GDPS application—that is, the Node app you created above in Configuration Server.
6. Use the key for Genesys Pulse to disable the Pulse Collector
7. Enter the location where GDPS is to be installed on your server.
   **Note:** This location must match the previous settings that you entered in Configuration Server.
   **Note:** For multi-node clusters, you must install the GDPS Application into exactly the same directory on every node. For example, if the path for Node 1 is `/genesys/gdps/gdps_n1`, it cannot be `/genesys/gdps/gdps_n2` for any of the other nodes. This requires manual intervention, since the installation package offers a default installation path based on the application name, which is therefore different for each node.
8. If the installation is successful, the console displays the following message:
   `Installation of Genesys GDPS, version 8.5.x has completed successfully.`
9. Inspect the directory tree of your system to make sure that the files have been installed in the location that you intended.

### End

**Note:** Genesys recommends that you regularly clear the following:

- The Spark temporary directory. You can find it in the system temporary directory with a name template of `spark-*`. The default location for this directory will be the **/tmp** folder of the user running the GDPS host (something like **/<user>/tmp**).
- Within the GDPS directory itself, the **<GDPS directory>/spark/work** folder.

## Configuration options

The following configuration options can be useful in setting up GDPS and your Spark cluster.

## [log] Section options

The [log] section configuration options are applied to the GDPS environment in a way that is similar to how they are used with UCS. UCS log options are documented here.

## spark Options

GDPS launches a dedicated Spark cluster and all of the GDPS nodes need to share the coordinates of the Spark Master node. In addition to this, each individual node has options that can be used to configure the mode with which Spark starts on its box. Default values should be sufficient in most circumstances.

### host

**Description:** The name of the Spark Master host. The value should be the same as what Java's **InetAddress.getLocalHost()** would return for the specified host.
**Default Value:** None
**Valid Values:** *hostname of the Spark Master node*
**Mandatory:** No
**Changes Take Effect:** After start/restart

### port

**Description:** The port number of the Spark Master host.
**Default Value:** `7077`
**Valid Values:** Valid port number
**Mandatory:** No
**Changes Take Effect:** After start/restart

### startMode

**Description:** The mode that will be used when starting Spark. If set to `off`, Spark will not be started by GDPS, and will instead have its state managed externally. If set to `worker`, only a worker node will be started. If set to `both`, both a worker node and a master node are started. **Note:** Genesys recommends that you set this option for each node to clearly specify the role. However, you can set the Cluster object to `worker` mode and override that value for the master node by setting that node to `both`.
**Default Value:** `worker`
**Valid Values:** `off`, `worker`, or `both`
**Mandatory:** No
**Changes Take Effect:** After start/restart

### masterWebPort

**Description:** The number of the TCP port that the Spark Master web UI will listen on. Note that this option is provided for cases when the default port has already been used by another service.
**Default Value:** `8080`
**Valid Values:** Valid port number
**Mandatory:** No

**Changes Take Effect:** After start/restart

### workerWebPort

**Description:** The number of the TCP port that the Spark Worker web UI will listen on. Note that this option is provided for cases when the default port has already been used by another service.
**Default Value:** 8081
**Valid Values:** Valid port number
**Mandatory:** No
**Changes Take Effect:** After start/restart

### executorMemory

**Description:** Use this option to manage the amount of memory used by Spark for executing tasks on each node. Genesys recommends at least two gigabytes per node, but more memory can improve performance if hardware allows. For information about the format, consult the Spark documentation.
**Default Value:** None
**Valid Values:** Valid memory limit
**Mandatory:** No
**Changes Take Effect:** After start/restart

### sparkHeartbeatTimeout

**Description:** The timeout value in seconds between two heartbeat calls to the Spark metrics API.
**Default Value:** 60
**Valid Values:** Positive integer
**Mandatory:** No
**Changes Take Effect:** After start/restart

### sparkStartTimeout

**Description:** The timeout value in seconds between a Spark start or restart and the first time its API is checked. On slower machines, it makes sense to increase this value so that Spark has enough time to start successfully (without initiating a restart cycle).
**Default Value:** 20
**Valid Values:** Positive integer
**Mandatory:** No
**Changes Take Effect:** After start/restart

### uri

**Description: Advanced.** For situations when Spark is running externally, you must set the URI instead of the host and port. The URI must include the protocol, in addition to the host and port.
**Default Value:** None
**Valid Values:** Valid Spark URI
**Mandatory:** No
**Changes Take Effect:** After start/restart

## spark.context

**Advanced.** This entire section is copied into **SparkContext**, so it can be used to tune the Spark options. You must have an in-depth understanding of Spark configuration if you are going to use this section.

To enable GDPS to handle dates in UTC format, the following option, set the value of **spark.executor.extraJavaOptions** to:

```
-Duser.timezone=UTC
```

# New in this Release

## 9.0.000.04

- Genesys Data Processing Server (GDPS) now supports Elasticsearch 6.2.4.
- GDPS now supports multiple job packages, each with its own SparkContext.
- GDPS no longer requires Cassandra to save data.

# Enabling a Secure Connection via HTTPS for GDPS

This article describes how to enable a secure HTTPs connection in GDPS for a Linux environment. There is a detailed discussion on how to configure SSL for Jetty at the following link: https://wiki.eclipse.org/Jetty/Howto/Configure_SSL

## Process

Enabling a secure HTTPS connection in GDPS involves the following steps:

**Server side**

1. Creating a secure port.
2. Creating/importing a certificate.
3. Updating the Jetty configuration to inject the certificate.

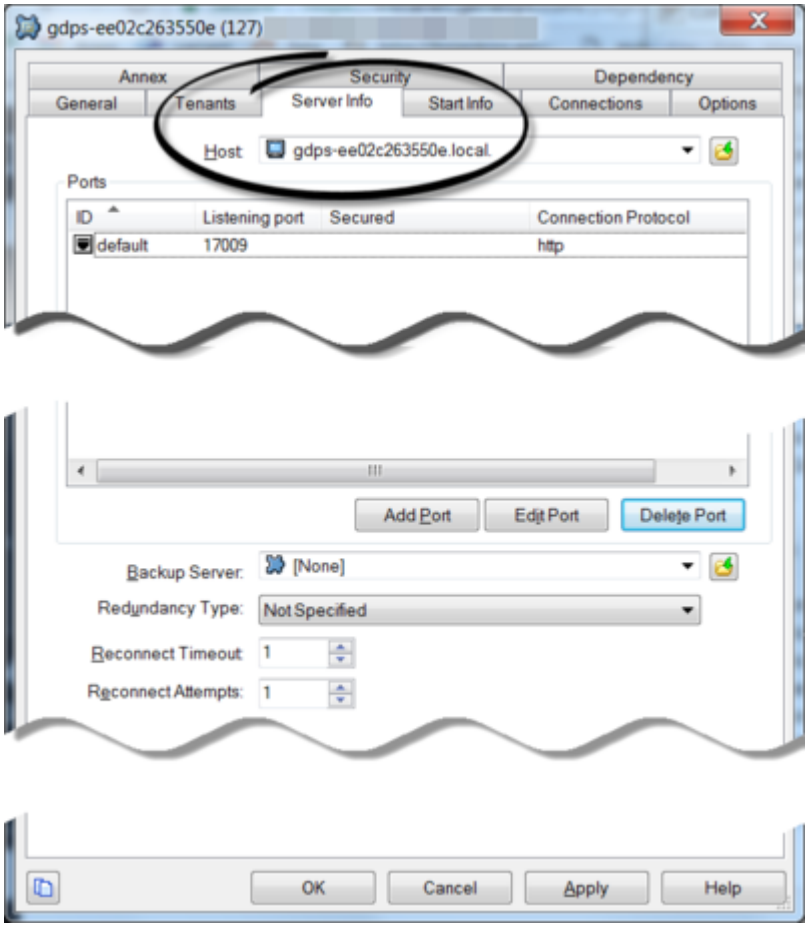
**Client side**

4. Confirming/testing with a browser or with curl.
5. Displaying the certificate.

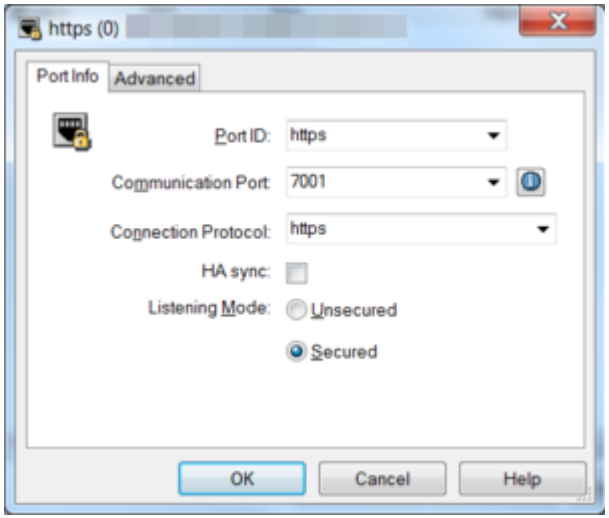
# Server side

## Creating a secure port

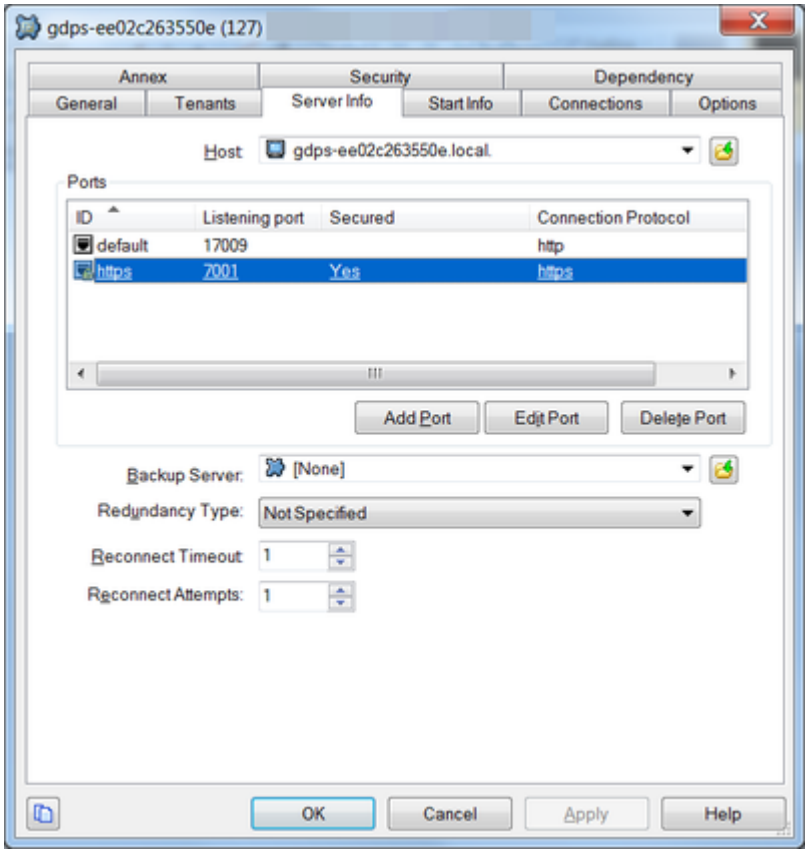
1. Login to Configuration Server and open the GDPS application.
2. Select the **Server Info** tab.




3. Click **Add Port.**




4. Click **Add Port**.
5. Create the new secured port by adding the Port ID `https`. Note that you can't update the default port from `http` to `https`.
6. Set the value of **Communication Port** to a free TCP port.
7. Set the value of **Connection Protocol** to `https`.
8. For the value of **Listening Mode**, check **Secured**.
9. On the **Advanced** tab, make sure that the **Transport Protocol Parameters** field contains the string `tls=1`.
10. Click OK to save your changes. When saving, you can safely ignore the Listening Mode certificate warning.
11. The configuration should look like this:


gdps-ee02c263550e (127)
Annex
Security
Dependency
General
Tenants
Server Info
Start Info
Connections
Options
Host:
gdps-ee02c263550e.local
Ports
ID
Listening port
Secured
Connection Protocol
default
17009
http
https
7001
Yes
https
Add Port
Edit Port
Delete Port
Backup Server:
[None]
Redundancy Type:
Not Specified
Reconnect Timeout:
1
Reconnect Attempts:
1
OK
Cancel
Apply
Help

## Creating/importing a certificate

### Creating a certificate

On the GDPS server, you can create a self-signed certificate (for internal or testing purposes) with the Java keytool as follows:

```
keytool -genkey \
-v \
-alias gdps2 \
-dname #:"CN=192.168.99.100,OU=IT,O=JPC,C=GB" \
-keypass password \
-keystore gdps2.jks \
-storepass password \
-keyalg "RSA" \
-sigalg "MD5withRSA" \
-keysize 2048 \
-validity 365
```

This will create a JKS store named `gdps2.jks`.

### Importing a certificate

Import your signed certificate by using the command below. For a production environment, import your signed certificate into a JKS store.

```
keytool -keystore keystore -importcert -alias alias -file certificate_file -trustcacerts
```

where:

- `keystore` is the name of your JSSE keystore.
- `alias` is the unique alias for your certificate in the JSSE keystore.
- `certificate_file` is the name of your certificate file. For example, `jetty.crt`.

## Updating Jetty server to inject certificate

Once you have a certificate (self-signed or imported one), you need to modify Jetty (see the procedure below) so that it fetches your certificate. Jetty provides by default a certificate out of the box. It should be replaced by your certificate.

1. Log into the GDPS server.
2. Navigate to the GDPS install directory.
3. Find the `jetty-ssh.xml` file. Example: `root@ee02c263550e:# cd /gdps/etc root@ee02c263550e:/gdps/etc# ls jetty-ssl.xml jetty-ssl.xml`
4. Edit this file, replacing the following lines with the path and password of your certificate:
   **Original Jetty SSL**

```
<Set name="KeyStorePath"><Property name="jetty.base" default="." />/<Property name="jetty.keystore" default="etc/keystore"/></Set>
 <Set name="KeyStorePassword"><Property name="jetty.keystore.password" default="OBF:1vny1zlo1x8e1vnw1vn61x8g1zlu1vn4"/></Set>
 <Set name="KeyManagerPassword"><Property name="jetty.keymanager.password" default="OBF:1u2u1wml1z7s1z7a1wnl1u2g"/></Set>
 <Set name="TrustStorePath"><Property name="jetty.base" default="." />/<Property name="jetty.truststore" default="etc/
 keystore"/></Set>
 <Set name="TrustStorePassword"><Property name="jetty.truststore.password" default="OBF:1vny1zlo1x8e1vnw1vn61x8g1zlu1vn4"/></Set>
```

   **New Jetty SSL**

```
<Set name="KeyStorePath"><Property name="jetty.base" default="." />/<Property name="jetty.keystore" default="gdps2.jks"/></Set>
 <Set name="KeyStorePassword"><Property name="jetty.keystore.password" default="password"/></Set>
 <Set name="KeyManagerPassword"><Property name="jetty.keymanager.password" default="password"/></Set>
 <Set name="TrustStorePath"><Property name="jetty.base" default="." />/<Property name="jetty.truststore" default="gdps2.jks"/></Set>
 <Set name="TrustStorePassword"><Property name="jetty.truststore.password" default="password"/></Set>
```

   Note that the password value should be obfuscated in production mode, using the prefix `OBF:` .
5. Save the file.
6. Restart GDPS.

## Resources

See documentation on jetty: https://www.eclipse.org/jetty/documentation/9.3.x/configuring-security-secure-passwords.html

### Example

```
export JETTY_VERSION=9.2.18.v20160721
java -cp lib/jetty-util-$JETTY_VERSION.jar org.eclipse.jetty.util.security.Password root password

2017-11-24 11:12:50.355:INFO::main: Logging initialized @249ms
password
OBF:1v2j1uum1xtv1zej1zer1xtn1uvk1v1v
MD5:5f4dcc3b5aa765d61d8327deb882cf99
CRYPT:rox7Jdqy.byUU
```

# Client side

## Confirming/testing with a browser or from the command line

Once GDPS has restarted, test from the client side using either a browser or from the command line.

### Browser

From a browser, navigate to the new secured URL—for example, `https://192.168.99.100:7001/data/package/packages`.

### Using curl

From the command line, use curl.

1. Make sure that the curl package you are using does support HTTPS by verifying the protocols supported by issuing a curl `--version` command as follows:

   ```
   $ curl --version
    curl 7.43.0 (x86_64-w64-mingw32) libcurl/7.43.0 OpenSSL/1.0.2d zlib/1.2.8 libidn/1.32 libssh2/1.6.0 librtmp/2.3
    Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtmp rtsp scp sftp smtp smtps telnet tftp
    Features: IDN Largefile SSPI Kerberos SPNEGO NTLM SSL libz TLS-SRP
   ```

2. Check that HTTPS is listed. If not, you will get a 'not supported' error message like this one:

```
curl --cacert C:\tmp\ca.crt https://192.168.99.100:7001/data/package/packages
 curl: (1) Protocol https not supported or disabled in libcurl
```

3. Once curl is sucessfully checked, retrieve the certificate by using an `openssl` command. The public certificate is stored in the `ca.crt` on the client side. Retrieve the certificate by using a command like this one:

   ```
   openssl s_client -connect 192.168.99.100:7001 -showcerts < /dev/null | openssl x509 -outform PEM > /c/tmp/ca.crt
   ```

4. Then issue the curl command with the retrieved certificate:

   ```
   curl --cacert /c/tmp/ca.crt https://192.168.99.100:7001/data/package/packages
   ```

### Disabling the certificate check

Alternatively, you can disable certicate check with curl by using the `-k` option (insecure flag) using a command like this one:

```
curl -k https://192.168.99.100:7001/data/package/packages
```

## Displaying the certificate

To display the certificate content from the client, use a command like this one:

```
openssl s_client -connect 192.168.99.100:7001 -showcerts
```

Example:

```
$ openssl s_client -connect 192.168.99.100:7001 -showcerts
Loading 'screen' into random state - done
CONNECTED(00000134)
depth=0 C = GB, O = JPC, OU = IT, CN = 192.168.99.100
verify error:num=18:self signed certificate
verify return:1
depth=0 C = GB, O = JPC, OU = IT, CN = 192.168.99.100
verify return:1
---
Certificate chain
0 s:/C=GB/O=JPC/OU=IT/CN=192.168.99.100
i:/C=GB/O=JPC/OU=IT/CN=192.168.99.100
```

```
-----BEGIN CERTIFICATE-----
MIIDITCCAgmgAwIBAgIESfSLVTANBgkqhkiG9w0BAQQFADBBMQswCQYDVQQGEwJH
QjEMMAoGA1UEChMDSlBDMQswCQYDVQQLEwJJVDEXMBUGA1UEAxMOMTkyLjE2OC45
OS4xMDAwHhcNMTcxMTI0MDgyNTM5WhcNMjcxMTIyMDgyNTM5WjBBMQswCQYDVQQG
EwJHQjEMMAoGA1UEChMDSlBDMQswCQYDVQQLEwJJVDEXMBUGA1UEAxMOMTkyLjE2
OC45OS4xMDAwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCZiNgEWisR
jGbMDfGLC52dMYblQtC305J0kKqTz675Cc/6AIa/i2KrxJ33nUYb8T9D8b9Y66rt
vzdAZfhirUE9A1xMiIyQqMNa+PahxhpgaYK26u/ev2YGI3EGrQfDXT1lNbp7pqaS
LaGLK5Dmpo21Ef0sfVnYa8VTc4uI3Q0kqhp2l0MdiHXDvgp6peursRoe6YVbAoV0
acrc19PRFcN6Iir+32wsSjOf8DlxbypfbQf1BK4V26SyNWi5DD4gEajPfVQfuh+s
dOum1egvvvO0j60HNPSFxJLWPsG2x+3L+9pA+WsmJ8DSRktCimsdH69V9jNKp36k
Kjz1CMCQogrvAgMBAAGjITAfMB0GA1UdDgQWBBQMqJkbaW5STz1JrNb9jh6ySlQy
bjANBgkqhkiG9w0BAQQFAAOCAQEAaEAkdMEd0P9yC1o3cWkKXWTs0aQduGKV4K1x
drCGTnwzPG2bszOz4VfZWpHjKlBv+Yeo+FR/Bz6Y/ByBmjjkGEAKXwoHnon8qcs+
xEc9l79c904vbB6WOW3BG5LmoyyXeYuOpJ0qHFUVpHmrz8sDwK57FS3kASomH0Y1
5daTXWSt/XHyNccaJdRTf1PAtzE7fo/tTw2l8jvQzBycPXi88fV2gTiTXst2Jzxq
Hvp+JqkRc9P2KMqq3TW3lmVJ2jHNyegrNAsfHFh/oHWu4Z0xxDD25KtWrR7Nxu+p
6BXNI2E4ZvHiMCYwna+ThjgT+OaH+HhXcRtfQX/Hjkl+aqdoPQ==
-----END CERTIFICATE-----
---
Server certificate
subject=/C=GB/O=JPC/OU=IT/CN=192.168.99.100
issuer=/C=GB/O=JPC/OU=IT/CN=192.168.99.100
---
No client certificate CA names sent
Peer signing digest: SHA512
Server Temp Key: ECDH, P-256, 256 bits
---
SSL handshake has read 1289 bytes and written 444 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES128-GCM-SHA256
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
Protocol : TLSv1.2
Cipher : ECDHE-RSA-AES128-GCM-SHA256
Session-ID: 5A17F573EBE27C68EBD5213F42349E364EBDC39325D36E59D0BAF9965BC4905C
Session-ID-ctx:
Master-Key: B353A0A3A8158CE021F9ED2516AB8422D410DBF8EFA4D6AC8445C07AF7A1B14AAEF8BC6E3BB677EA7FDEFEA2048A07EE
Key-Arg : None
```

```
PSK identity: None
PSK identity hint: None
SRP username: None
Start Time: 1511519603
Timeout : 300 (sec)
Verify return code: 18 (self signed certificate)
---
closed
```

## Other resources

- https://www.eclipse.org/jetty/documentation/9.3.x/configuring-security-secure-passwords.html
- Download a curl for Windows that supports HTTPS from this location: https://curl.haxx.se/download.html