Genesys Customer Experience Insights User's Guide

SSO for Genesys CX Insights

12/18/2025

Contents

SSO for Genesys CX Insights

Warning

Special Support Statement LDAP and SAML functionality is provided as a preview feature. Genesys will support it to the best of our efforts, however we cannot guarantee it works in all configurations. The customer is responsible for testing everything in a lab/test environment before making any plans for production deployment.

Important

In keeping with Genesys' commitment to diversity, equality, and inclusivity, beginning with release 9.0.019.01, some pod names are changed; this document refers to "gcxi-primary" and "gcxi-secondary" pods. In release 9.0.019.00 and earlier, these pods were named "gcxi-master" and "gcxi-slave".

You can optionally configure Genesys CX Insights to use LDAP or SAML. Single sign-on (SSO) allows logged-in users to navigate across multiple applications without re-entering their credentials. This page provides example steps that may require modification to suit your environment; contact Genesys Professional Services if you need further assistance:

- Enable and configure SAML / Manage SAML
- Enable and configure LDAP / Manage LDAP

Enable and configure SAML

To enable SAML, complete one of the procedures in this section:

- Enable SAML in deployments that use Helm
- Enable SAML in deployments that use Kubernetes descriptors

Procedure: Enable SAML in deployments that use Helm

Purpose: Enable SAML in scenarios where you deploy Genesys CX Insights with Kubernetes using Helm or OpenShift using Helm.

Prerequisites

Before you begin:

- GCXI release 9.0.019.00 or later is deployed in your environment. GCXI serves as the SAML service provider (SP).
- You have access to a SAML Identity provider (IdP).
- This procedure assumes that the default namespace is **gcxi**. (If yours is not, then you must add **-n gcxi** to all kubectl commands.)

Steps

1. Execute the following command to back up the GCXI meta database:

   ```
   helm upgrade <release_name> <path_to_chart> --reuse-values --set gcxi.deployment.deployJobBackup=true
   ```

   where:

   - <release_name> — GCXI Helm release name.
   - <path_to_chart> — the path to the Helm chart files.

     For example:

     ```
     helm upgrade gcxi-helm gcxi/ --reuse-values --set gcxi.deployment.deployJobBackup=true
     ```

1. Wait until job gcxi-backup is finished.

2. Execute the following command to stop one container:

   ```
   helm upgrade -f <values_file> --set gcxi.replicas.worker=1 <release_name> <path_to_chart>
   ```

   where:

   <values_file> — the name of your values YAML file. This is typically the same file you used when you deployed using the instructions on: Genesys CX Insights - Kubernetes using Helm.

   <release_name> — GCXI Helm release name.

   <path_to_chart> — the path to the Helm chart files.

   For example:

   ```
   helm upgrade -f values-test.yaml --set gcxi.replicas.worker=0 gcxi-helm gcxi/
   ```

3. Generate SAML configuration files — follow the instructions on the MicroStrategy website: How to Generate configuration files, but note the following:

   - Where the instructions on the MicroStrategy website ask for an *entity base URL*, use: `http://<server>:<port>/MicroStrategy/saml/config/` open. This URL works in many scenarios, but if it does not — for example, because your environment uses a reverse proxy or load balancer — see the MicroStrategy documentation.
   - You must connect using the Tomcat administrator credentials that you set using the `gcxi.env.TOMCAT_ADMINPWD` variable during installation. If you did not set a password, you can set one during upgrade, or contact Customer Care for assistance.

4. Execute the following command and examine the results to ensure that SAML-related files appear inside the running pod:

   ```
   kubectl exec gcxi-0 -- ls /opt/tomcat/webapps/MicroStrategy/WEB-INF/classes/resources/SAML
   ```

5. In a convenient location, create an empty directory called **saml_folder**. (You can optionally use another name, but this procedure assumes you used the name **saml_folder**.)

6. Copy the SAML configuration files from the pod to the saml_folder directory:

   ```
   kubectl cp gcxi-0:/opt/tomcat/webapps/MicroStrategy/WEB-INF/classes/resources/SAML/ saml_folder
   ```

   Check that files are copied successfully into the saml_folder, and that no other files are present in the directory.
7. Using the **saml_folder/SPMetadata.xml** file, register with the IdP — the exact steps vary depending on your IdP.
8. Copy the **IDPMetadata.xml** file provided by your IdP into the **saml_folder**.
9. Ensure that the **saml_folder** directory contains the following files:

   ```
   custom
   IDPMetadata.xml
   MstrSamlConfig.xml
   SamlKeystore.jks
   SPMetadata.xml
   SpringSAMLConfig.xml
   ```

   Genesys CX Insights does not use the custom folder.
10. Execute the following command to create a new config map - gcxi-saml:

    ```
    kubectl create cm gcxi-saml --from-file saml_folder
    ```

    Once this is complete, delete the **saml_folder** as it is no longer required.
11. Edit the **gcxi.properties** file add the following variables to enable SAML and set it as default login mode:

    ```
    MSTR_WEB_SAML_ON=true
    MSTR_WEB_DEFAULT_LOGIN_MODE= 1048576
    ```
12. Execute the following commands to stop the gcxi pod:

    ```
    helm upgrade -f <values_file> --set gcxi.replicas.worker=0 <release_name> <path_to_chart>
    ```

    where:

<values_file> — the name of your values YAML file. This is typically the same file you used when you deployed using the instructions on: Genesys CX Insights - Kubernetes using Helm.

<release_name> — GCXI Helm release name.

<path_to_chart> — the path to the Helm chart files.

For example:

```
helm upgrade -f values-test.yaml --set gcxi.replicas.worker=0 gcxi-helm gcxi/
```

Wait until pods are stopped before proceeding.

13. Execute the following commands to upgrade GCXI (using values from **gcxi.properties** to set container variables):

```
helm upgrade -f <values_file> --set-file gcxi.envext=gcxi.properties <release_name> <path_to_chart>
```

where:

<values_file> — the name of your values YAML file. This is typically the same file you used when you deployed using the instructions on: Genesys CX Insights - Kubernetes using Helm.

<release_name> — gcxi helm release name.

<path_to_chart> — the path to the Helm chart files.

For example:

```
helm upgrade -f values-test.yaml --set-file gcxi.envext=gcxi.properties gcxi-helm gcxi/
```

Wait while GCXI is installed.

14. Open MicroStrategy Web (http://<server>:<port>/MicroStrategy/servlet/mstrWeb); you will be automatically forwarded to the SAML IdP login page.

Procedure: Enable SAML in deployments that use Kubernetes descriptors

Purpose: Enable SAML in environments where you deploy GCXI using Kubernetes descriptors.

Prerequisites

Before you begin:

- GCXI release 9.0.019.00 or later is deployed in your environment. GCXI serves as the SAML service provider (SP).
- You have access to a SAML Identity provider (IdP).
- This procedure assumes that the default namespace is **genesys**. (If yours is not, then you must add **-n genesys** to all kubectl commands.)

Steps

1. Execute the following command to back up the GCXI meta database:

   ```
   kubectl apply -f k8s/gcxi-backup.yaml
   ```

   Wait until job gcxi-backup is finished.

2. Execute the following command to stop one GCXI container:

   ```
   kubectl scale --replicas=0 deployment gcxi-secondary
   ```

3. Generate SAML configuration files — follow the instructions on the MicroStrategy website: How to Generate configuration files, but note the following:
   - Where the instructions on the MicroStrategy website ask for an *entity base URL*, use: `http://<server>:<port>/MicroStrategy/saml/config/open`.
   - You must connect to `http://<server>:<port>/MicroStrategy/saml/config/open` using Tomcat administrator credentials. You can change the

Tomcat admin password by setting the variable TOMCAT_ADMINPWD (see Procedure: Configuring Kubernetes Secrets) or contact Customer Care for assistance.

4. Execute the following command and examine the results to ensure that SAML-related files appear inside the running pod:

```
kubectl exec $(kubectl get po -o name | grep gcxi-primary) -- ls /opt/tomcat/webapps/MicroStrategy/WEB-INF/classes/
resources/SAML
```

5. In a convenient location, create an empty directory called **saml_folder**. (You can optionally use another name, but this procedure assumes you used the name **saml_folder**.)
6. Copy the SAML configuration files from the pod to the saml_folder directory:

```
kubectl cp $(kubectl get pod -l role=gcxi-primary -o jsonpath="{.items[0].metadata.name}"):/opt/tomcat/webapps/
MicroStrategy/WEB-INF/classes/resources/SAML/ saml_folder
```

   Check that files are copied successfully into the saml_folder, and that no other files are present in the directory.

7. Using the **saml_folder/SPMetadata.xml** file, register with the IdP — the exact steps vary depending on your IdP.
8. Copy the **IDPMetadata.xml** file provided by your IdP into the saml_folder.
9. Ensure that the saml_folder directory contains the following files:

```
custom
IDPMetadata.xml
MstrSamlConfig.xml
SamlKeystore.jks
SPMetadata.xml
SpringSAMLConfig.xml
```

   Genesys CX Insights does not use the custom folder.

10. Execute the following command to create a new config map - gcxi-saml:

```
kubectl create cm gcxi-saml --from-file saml_folder
```

Once this is complete, delete the saml_folder as it is no longer required.

11. Edit the **gcxi.properties** file add the following variables to enable SAML and set it as default login mode:

```
MSTR_WEB_SAML_ON=true
MSTR_WEB_DEFAULT_LOGIN_MODE= 1048576
```

12. Execute the following commands to recreate the **gcxi-config** configmap:

```
kubectl delete cm gcxi-config
kubectl create cm gcxi-config --from-env-file k8s/gcxi.properties
```

13. Edit the **gcxi.yaml** file and add gcxi-saml volume and volumeMount to definition of gcxi-secondary and gcxi-primary deployments:

```
          ...
          volumeMounts:
          - mountPath: /genesys/gcxi_config/saml
            name: gcxi-saml
                              ...
        volumes:
        - name: gcxi-saml
          configMap:
            name: gcxi-saml
            optional: true
          ...
```

14. Execute the following commands to recreate the gcxi pods and thereby apply the **gcxi.yaml** changes:

```
kubectl delete -f k8s/gcxi.yaml
kubectl create -f k8s/gcxi.yaml
```

Wait while GCXI is installed.

15. Open MicroStrategy Web (http://<server>:<port>/MicroStrategy/servlet/mstrWeb); you will be automatically forwarded to the SAML IdP login page.

Manage SAML

If you've enabled SAML, use the information in this section to manage it.

- Disable SAML in deployments that use Helm
- Disable SAML in deployments that use Kubernetes descriptors

Procedure: Disable SAML in deployments that use Helm

Purpose: Disable SAML when GCXI is deployed using Helm.

Prerequisites

This procedure assumes that the default namespace is **gcxi**. (If yours is not, then you must add **-n gcxi** to all kubectl commands.)

Steps

1. Execute the following commands to stop GCXI:

   ```
   helm upgrade -f <values_file> --set gcxi.replicas.worker=0 <release_name> <path_to_chart>
   ```

   where:

   <values_file> — the name of your values YAML file. This is typically the same file you used when you deployed using the instructions on: Genesys CX Insights - Kubernetes using Helm.

   <release_name> — GCXI Helm release name.

   <path_to_chart> — the path to the Helm chart files.

For example:

```
helm upgrade -f values-test.yaml --set gcxi.replicas.worker=0 gcxi-helm gcxi/
```

2. Execute the following commands to upgrade GCXI (using values from **gcxi.properties** to set container variables) and restart the pods:

   ```
   helm upgrade -f <values_file> <release_name> <path_to_chart>
   ```

   where:

   <values_file> — the name of your values YAML file. This is typically the same file you used when you deployed using the instructions on: Genesys CX Insights - Kubernetes using Helm.

   <release_name> — GCXI Helm release name.

   <path_to_chart> — the path to the Helm chart files.

   For example:

   ```
   helm upgrade -f values-test.yaml gcxi-helm gcxi/
   ```

   Wait while GCXI is upgraded.

Procedure: Disable SAML in deployments that use Kubernetes descriptors

Purpose: Disable SAML when GCXI is deployed using Kubernetes descriptors.

Prerequisites

This procedure assumes that the default namespace is **gcxi**. (If yours is not, then you must add **-n gcxi** to all kubectl commands.)

Steps

1. Edit the **gcxi.properties** file, and remove the following variables:

```
MSTR_WEB_SAML_ON=true
MSTR_WEB_DEFAULT_LOGIN_MODE= 1048576
```

2. Execute the following commands to recreate the gcxi-config configmap:

```
kubectl delete cm gcxi-config
kubectl create cm gcxi-config --from-env-file k8s/gcxi.properties
```

3. Execute the following commands to recreate the gcxi pods:

```
kubectl delete -f k8s/gcxi.yaml
kubectl create -f k8s/gcxi.yaml
```

Update SAML

If you enabled SAML in Genesys CX Insights 100.0.026.0000 or earlier, follow the instructions in this section before upgrading Genesys CX Insights. If you enabled SAML in Genesys CX Insights 100.0.026.0001 or later, ignore this section.

- UpdateSAML in deployments that use Helm
- Update SAML in deployments that use Kubernetes descriptors

Procedure: Update SAML in deployments that use Helm

Purpose: Update SAML when GCXI is deployed using Helm.

Prerequisites

This procedure assumes that the default namespace is **gcxi**. (If yours is not, then you must add **-n gcxi** to all kubectl commands.)

Steps

If **SpringSAMLConfig.xml** *is not* customized in your existing deployment:

1. Execute the following command to edit **gcxi-saml** ConfigMap file:

   ```
   kubectl edit cm gcxi-saml
   ```

2. Remove `SpringSAMLConfig.xml`, and save the file.
3. Execute the following command to verify that **gcxi-saml** contains now four variables:

   ```
   kubectl get cm gcxi-saml
   ```

   The output should be similar to the following:

   ```
   NAME        DATA   AGE
    gcxi-saml   4      127m
   ```

4. Upgrade Genesys CX Insights using the steps in Upgrade GCXI using Helm.

OR, if **SpringSAMLConfig.xml** *is* customized in your existing deployment, use the following steps:

1. Execute the following command to back up the **SpringSAMLConfig.xml** file:

```
kubectl cp gcxi-0:/opt/tomcat/webapps/MicroStrategy/WEB-INF/classes/resources/SAML/SpringSAMLConfig.xml
 SpringSAMLConfig.xml_backup
```

2. Execute the following command to edit **gcxi-saml** ConfigMap file:

   ```
   kubectl edit cm gcxi-saml
   ```

3. Remove `SpringSAMLConfig.xml`, and save the file.
4. Execute the following command to verify that **gcxi-saml** contains now four variables:

   ```
   kubectl get cm gcxi-saml
   ```

   The output should be similar to the following:

   ```
   NAME        DATA   AGE
   gcxi-saml   4      127m
   ```

5. Complete the steps in Upgrade GCXI using Helm. Wait until pods are running before your continue with this procedure. Note that the updated version of **SpringSAMLConfig.xml** is created inside gcxi containers.
6. Execute the following command to copy the SAML configuration files from the pod to a temporary location:

   ```
   kubectl cp gcxi-0:/opt/tomcat/webapps/MicroStrategy/WEB-INF/classes/resources/SAML/ <saml_folder_upgrade>
   ```

   where < `saml_folder_upgrade`> is a temporary folder.

7. Verify that the destination folder contains the following files (the custom folder is not used):

   ```
   custom
   IDPMetadata.xml
   MstrSamlConfig.xml
   SamlKeystore.jks
   SPMetadata.xml
   SpringSAMLConfig.xml
   ```

8. Using the instructions in Upgrade a Customized SAML System, edit `<saml_folder_upgrade>/SpringSAMLConfig.xml`.

9. Execute the following commands to recreate **gcxi-saml** ConfigMap:

```
kubectl delete cm gcxi-saml
kubectl create cm gcxi-saml --from-file saml_folder_upgrade
```

   This adds `SpringSAMLConfig.xml` to the gcxi-saml file.

10. Restart the pod.

Procedure: Update SAML in deployments that use Kubernetes

Purpose: Update SAML when GCXI is deployed using Kubernetes descriptors.

Prerequisites

This procedure assumes that the default namespace is **gcxi**. (If yours is not, then you must add **-n gcxi** to all kubectl commands.)

Steps

If **SpringSAMLConfig.xml** *is not* customized in your existing deployment:

1. Execute the following command to edit **gcxi-saml** ConfigMap file:

```
kubectl edit cm gcxi-saml
```

2. Remove `SpringSAMLConfig.xml`, and save the file.

3. Execute the following command to verify that **gcxi-saml** contains now four variables:

   ```
   kubectl get cm gcxi-saml
   ```

   The output should be similar to the following:

   ```
   NAME        DATA   AGE
   gcxi-saml   4      127m
   ```

4. Upgrade Genesys CX Insights using the steps in Upgrading Genesys CX Insights.

OR, if **SpringSAMLConfig.xml** *is* customized in your existing deployment, use the following steps:

1. Execute the following command to back up the **SpringSAMLConfig.xml** file:

   ```
   kubectl cp gcxi-0:/opt/tomcat/webapps/MicroStrategy/WEB-INF/classes/resources/SAML/SpringSAMLConfig.xml
   SpringSAMLConfig.xml_backup
   ```

2. Execute the following command to edit **gcxi-saml** ConfigMap file:

   ```
   kubectl edit cm gcxi-saml
   ```

3. Remove `SpringSAMLConfig.xml`, and save the file.
4. Execute the following command to verify that **gcxi-saml** contains now four variables:

   ```
   kubectl get cm gcxi-saml
   ```

   The output should be similar to the following:

   ```
   NAME        DATA   AGE
   gcxi-saml   4      127m
   ```

5. Complete the steps in Upgrading Genesys CX Insights. Wait until pods are running before your continue with this procedure. Note that the updated version of **SpringSAMLConfig.xml** is created inside the gcxi container.

6. Execute the following command to copy the SAML configuration files from the pod to a temporary location:

   ```
   kubectl cp gcxi-0:/opt/tomcat/webapps/MicroStrategy/WEB-INF/classes/resources/SAML/ <saml_folder_upgrade>
   ```

   where < saml_folder_upgrade> is a temporary folder.

7. Verify that the destination folder contains the following files (the custom folder is not used):

   ```
   custom
   IDPMetadata.xml
   MstrSamlConfig.xml
   SamlKeystore.jks
   SPMetadata.xml
   SpringSAMLConfig.xml
   ```

8. Using the instructions in Upgrade a Customized SAML System, edit <saml_folder_upgrade>/SpringSAMLConfig.xml.
9. Execute the following commands to recreate **gcxi-saml** ConfigMap:

   ```
   kubectl delete cm gcxi-saml
   kubectl create cm gcxi-saml --from-file saml_folder_upgrade
   ```

   This adds SpringSAMLConfig.xml to the gcxi-saml file.

10. Restart the pods.

Enable and configure LDAP

To enable LDAP, complete one of the procedures in this section:

- Enable LDAP in deployments that use Helm
- Enable LDAP in deployments that use Kubernetes descriptors

In either case, you must then complete the section Configure LDAP.

Procedure: Enable LDAP in deployments that use Helm

Purpose: Enable LDAP when GCXI is deployed using Helm.

Prerequisites

Ensure that GCXI release 9.0.014.00 or later is deployed in your environment. This procedure assumes that the default namespace is **gcxi**. (If yours is not, then you must add **-n gcxi** to all kubectl commands.)

Steps

Before you begin, complete the following steps to configure GCXI:

1. Locate the OpenLDAP library inside the container:
    1. Execute the following command to ensure that the OpenLDAP library is installed, and locate the library's path:

        ```
        kubectl exec gcxi-0 -- ldconfig -p | grep ldap
        ```

    2. Copy or note the path, for example **/usr/lib64/libldap-2.4.so.2.**
2. Execute the following command to back up the GCXI meta database:

    ```
    helm upgrade <release_name> <path_to_chart> --reuse-values --set gcxi.deployment.deployJobBackup=true
    ```

    where:

    <release_name> — GCXI Helm release name.

<path_to_chart> — the path to the Helm chart files.

For example:

```
helm upgrade gcxi-helm gcxi/ --reuse-values --set gcxi.deployment.deployJobBackup=true
```

Wait until job gcxi-backup is finished.

3. Execute the following command to prevent the backup job from running every time release is upgraded:

```
helm upgrade <release_name> <path_to_chart> --reuse-values --set gcxi.deployment.deployJobBackup=false
```

where:

<release_name> — GCXI Helm release name.

<path_to_chart> — the path to the Helm chart files.

For example:

```
helm upgrade gcxi-helm gcxi/ --reuse-values --set gcxi.deployment.deployJobBackup=false
```

4. Execute the following command to stop gcxi pods:

```
helm upgrade <release_name> <path_to_chart> --reuse-values --set gcxi.replicas.worker=0
```

where:

<release_name> — GCXI Helm release name.

<path_to_chart> — the path to the Helm chart files.

For example:

```
helm upgrade gcxi-helm gcxi/ --reuse-values --set gcxi.replicas.worker=0
```

Wait until pods are stopped before proceeding.

5. Edit the **gcxi.properties** file and add the following variables to enable LDAP and set it as default login mode:

   ```
   MSTR_WEB_LDAP_ON=true
   MSTR_WEB_DEFAULT_LOGIN_MODE=16
   ```

   Add no comments, extra spaces, or blank lines.

6. Execute the following command to upgrade and restart GCXI (using values from **gcxi.properties** to set container variables) and restart the pods:

   ```
   helm upgrade <release_name> <path_to_chart> --reuse-values --set-file gcxi.envext=gcxi.properties --set
   gcxi.replicas.worker=2
   ```

   where:

   <release_name> — GCXI Helm release name.

   <path_to_chart> — the path to the Helm chart files.

   For example:

   ```
   helm upgrade gcxi-helm gcxi/ --reuse-values --set-file gcxi.envext=gcxi.properties --set gcxi.replicas.worker=2
   ```

   Wait while GCXI is upgraded.

LDAP is now enabled in GCXI, and remains enabled after container restart.

Procedure: Enable LDAP in deployments that use Kubernetes descriptors

Purpose: Gather information you will need to configure LDAP, and enable LDAP in GCXI.

Prerequisites

- GCXI release 9.0.014.00 or later is deployed in your environment
- This procedure assumes that the default namespace is **genesys**. (If yours is not, then you must add **-n genesys** to all kubectl commands.)

Steps

Before you begin, complete the following steps to configure GCXI:

1. Locate the OpenLDAP library inside the container:
    1. Execute the following command to ensure that the OpenLDAP library is installed and locate the library's path:

        ```
        kubectl exec  $(kubectl get po -o name |  grep gcxi-primary) -- ldconfig -p | grep ldap
        ```

    2. Copy or note the path, for example **/usr/lib64/libldap-2.4.so.2.**
2. Enable LDAP:
    1. Execute the following command to back up the GCXI meta database:

        ```
        kubectl apply -f k8s/gcxi-backup.yaml
        ```

        Wait until job gcxi-backup is finished.

    1. Execute the following commands to stop currently running containers:

        ```
        kubectl scale --replicas=0 deployment gcxi-secondary
        ```

        ```
        kubectl scale --replicas=0 deployment gcxi-primary
        ```

    2. Edit the **gcxi.properties** file, and set **MSTR_WEB_LDAP_ON=true**.

3. To make LDAP the default login mode, edit the **gcxi.properties** file, and set **MSTR_WEB_DEFAULT_LOGIN_MODE=16**
4. Execute the following commands to load gcxi.properties into Kubernetes:

   ```
   kubectl delete configmap gcxi-config
   ```

   ```
   kubectl create configmap gcxi-config --from-env-file=k8s/gcxi.properties
   ```

5. Execute the following command to start the PRIMARY container:

   ```
   kubectl scale --replicas=1 deployment gcxi-primary
   ```

   Wait until PRIMARY is done (wait until Tomcat is up, and MicroStrategyWeb page is available).
6. Execute the following command to start the SECONDARY container:

   ```
   kubectl scale --replicas=1 deployment gcxi-secondary
   ```

   LDAP is now enabled in GCXI, and remains enabled after container restart.

If you plan to use LDAP, you must also mount the folder where your certificates reside. See LDAP with SSL for more information.

Configure LDAP

For both Helm and Kubernetes deployments, use the instructions in this section to configure LDAP.

Procedure: Configure LDAP connection settings

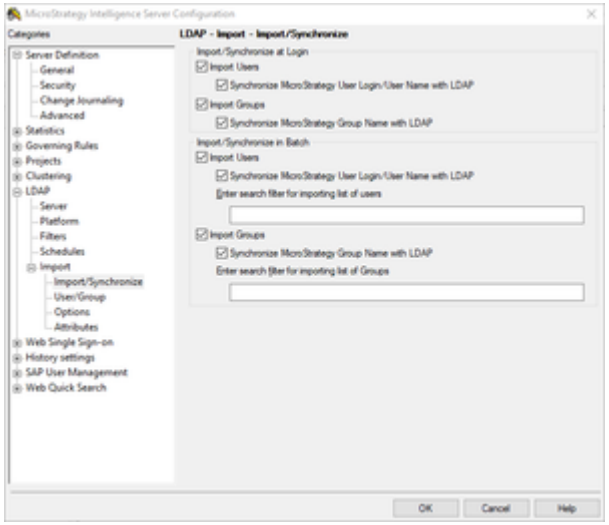
Purpose: To set up a connection to the LDAP server.

Steps

1. Log in to MicroStrategy Developer.
2. Click **Administration** > **Server** > **Configure MicroStrategy Intelligence Server**.
3. Click **LDAP**, and in the **Server** section, fill in the following parameters:
   - **Host**: `<hostname_or_ip_of_ldap_server>`
   - **Port**: `<port_of_ldap_server>`
   - **Security connection**: `Cleartext`
   - **Authentication method**: `Binding`
   - **Authentication User** (This user account is used when GCXI has to access Active Directory, make a search, list all users, and so on. It is a 'service user'.)
     For example:
     - `DN: CN=<user_cn>,OU=<ou>,…, DC=<dc>,…`
     - `Password: <domain password>`
4. Click **LDAP** > **Platform**, and fill in the following parameters:
   - **LDAP server vendor name**: `<your_vendor>` (for example, Microsoft Active Directory)
   - **LDAP Connectivity Driver**: `<your connectivity driver>` (for example, Open LDAP)
   - **Intelligence Server platform**: Linux
   - **LDAP connectivity file names**: `<path_to_openldap_lib>` (This is the path inside the gcxi container, which you discovered in Enable LDAP in deployments that use Kubernetes descriptors.)

5. Click **LDAP** > **Filters**, and fill in the following parameters:
   - **Root DN**: `DC=<dc>,DC=<dc>…` (This is the starting point of the LDAP search.)
   - **User search filter**: `<user_search>` (The value you enter here depends on LDAP settings. For example, (&(objectclass=person)(sAMAccountName=#LDAP_LOGIN#)).)
   - **Group search filter**: `<group_search>` (The value you enter here depends on LDAP settings. For example, (&(objectclass=group) (member=#LDAP_DN#)))
   - **Number of group levels above to import**: 1
6. To verify that the connection between the LDAP server and Intelligence server can be established, attempt to log in to MicroStrategy Web (<IP>/MicroStrategy/servlet/mstrWeb) using LDAP credentials. Note that even when you successfully log in, user privileges are not yet set. Privileges can be manually assigned to each user in mstrServerAdmin, or imported and configured as described in LDAP Import, synchronization and linking.

LDAP Import, synchronization and linking



LDAP - Import

You must configure privileges for users who log in using LDAP credentials. Detailed information about this step is available in the following MicroStrategy document: KB18506: Importing and linking users using LDAP integration with the MicroStrategy Intelligence Server 9.x and newer.

LDAP with SSL

To support SSL with LDAP, you must run the container with a mounted folder containing LDAP certificates, and ensure that the folder is mounted on all servers in the cluster. For more information, see How to configure LDAP connectivity using Clear text.

Manage LDAP

If you've enabled LDAP, use the information in this section to manage it.

- Disable LDAP in deployments that use Helm
- Disable LDAP in deployments that use Kubernetes descriptors

Procedure: Disable LDAP in deployments that use Helm

Purpose: Disable LDAP when GCXI is deployed using Helm.

Prerequisites

This procedure assumes that the default namespace is **gcxi**. (If yours is not, then you must add **-n gcxi** to all kubectl commands.)

Steps

1. Execute the following commands to stop GCXI:

   ```
   helm upgrade <release_name> <path_to_chart> -n gcxi --reuse-values --set gcxi.replicas.worker=0
   ```

   where:

   <release_name> — GCXI Helm release name.

   <path_to_chart> — the path to the Helm chart files.

   For example:

```
helm upgrade gcxi-helm gcxi/ -n gcxi --reuse-values --set gcxi.replicas.worker=0
```

2. Execute the following commands to start the GCXI pods without LDAP enabled:

   ```
   helm upgrade <release_name> <path_to_chart> -n gcxi --reuse-values --set gcxi.replicas.worker=2 --set-file gcxi.envext=
   ```

   where:

   <release_name> — GCXI Helm release name.

   <path_to_chart> — the path to the Helm chart files.

   For example:

   ```
   helm upgrade gcxi-helm gcxi/ -n gcxi --reuse-values --set gcxi.replicas.worker=2 --set-file gcxi.envext=
   ```

Procedure: Disable LDAP in deployments that use Kubernetes descriptors

Purpose: Disable LDAP when GCXI is deployed using Kubernetes descriptors.

Prerequisites

This procedure assumes that the default namespace is **genesys**. (If yours is not, then you must add **-n genesys** to all kubectl commands.)

Steps

1. Edit the **gcxi.properties** file, and remove the following variables:

```
MSTR_WEB_LDAP_ON=true
MSTR_WEB_DEFAULT_LOGIN_MODE=16
```

2. Execute the following commands to recreate the gcxi-config configmap:

```
kubectl delete cm gcxi-config
kubectl create cm gcxi-config --from-env-file k8s/gcxi.properties
```

3. Execute the following commands to recreate the gcxi pods:

```
kubectl delete -f k8s/gcxi.yaml
kubectl create -f k8s/gcxi.yaml
```