Genesys Customer Experience User's Guide

SSO pour Genesys CX Insights

12/15/2025

Contents

SSO pour Genesys CX Insights

Avertissement

Déclaration de support spécial Les fonctionnalités LDAP et SAML sont fournies en avant-première. Genesys les prendra en charge du mieux possible, mais nous ne pouvons pas garantir qu'elles fonctionnent dans toutes les configurations. Il incombe au client de tout tester dans un environnement de laboratoire/test avant de planifier un déploiement en production.

Important

Conformément à l'engagement de Genesys en matière de diversité, d'égalité et d'inclusivité, à partir de la version 9.0.019.01, certains noms de pod ont été modifiés ; le présent document fait référence aux pods « gcxi-primary » et « gcxi-secondary ». Dans la version 9.0.019.00 et les versions antérieures, ces pods étaient nommés « gcxi-master » et « gcxi-slave ».

Vous pouvez configurer Genesys CX Insights pour qu'il utilise LDAP ou SAML. L'authentification unique (SSO) permet aux utilisateurs connectés de naviguer entre plusieurs applications sans avoir à saisir de nouveau leurs informations d'identification. Cette page fournit des exemples d'étapes qui peuvent nécessiter des modifications pour s'adapter à votre environnement ; contactez Genesys Professional Services si vous avez besoin d'une aide supplémentaire :

- Activer et configurer SAML/Gérer SAML
- Activer et configurer LDAP/Gérer LDAP

Activer et configurer SAML

Pour activer SAML, suivez l'une des procédures de cette section :

- Activer SAML dans les déploiements qui utilisent Helm
- Activer SAML dans les déploiements qui utilisent des descripteurs Kubernetes

Procedure: Activer SAML dans les déploiements qui utilisent Helm

Purpose: Activez SAML dans les scénarios où vous déployez Genesys CX Insights avec Genesys CX Insights - Kubernetes via Helm ou OpenShift via Helm.

Prerequisites

Avant de commencer :

- La version 9.0.019.00 ou une version ultérieure de GCXI est déployée dans votre environnement. GCXI fait office de fournisseur de services (SP) SAML.
- Vous avez accès à un fournisseur d'identité (IdP) SAML.

Steps

1. Exécutez la commande suivante pour effectuer une sauvegarde de la base de données méta GCXI :

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

2. Exécutez la commande suivante pour arrêter un conteneur :

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

   où :

   <values_file> — nom de votre fichier de valeurs YAML. Il s'agit généralement du même fichier que celui que vous avez utilisé lors du déploiement à l'aide des instructions figurant sur : Genesys CX Insights - Kubernetes via Helm.

   <release_name> — nom de la version Helm GCXI.

   <path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

   Par exemple :

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

3. Générez les fichiers de configuration SAML — suivez les instructions du site Web de MicroStrategy : Comment générer les fichiers de configuration, mais notez les points suivants :

   - Lorsque les instructions du site Web de MicroStrategy demandent une *URL de base d'entité*, utilisez : `http://<server>:8080/MicroStrategy/saml/config/open`.
   - Vous devez vous connecter en utilisant les informations d'identification de l'administrateur Tomcat que vous avez définies à l'aide de la variable `gcxi.env.TOMCAT_ADMINPWD` lors de

l'installation. Si vous n'avez pas défini de mot de passe, vous pouvez en définir un pendant la mise à niveau ou contacter le service client pour obtenir de l'aide.

4. Exécutez la commande suivante et examinez les résultats pour vous assurer que les fichiers liés à SAML apparaissent dans le pod en cours d'exécution :

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

5. À un emplacement approprié, créez un répertoire vide appelé **saml_folder**. (Vous pouvez utiliser un autre nom, mais cette procédure suppose que vous avez utilisé le nom **saml_folder**).
6. Copiez les fichiers de configuration SAML du pod dans le répertoire saml_folder :

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

   Vérifiez que les fichiers sont correctement copiés dans saml_folder et qu'aucun autre fichier ne figure dans le répertoire.

7. À l'aide du fichier **saml_folder/SPMetadata.xml**, enregistrez-vous auprès de l'IdP — les étapes exactes varient suivant votre IdP.
8. Copiez le fichier **IDPMetadata.xml** fourni par votre IdP dans **saml_folder**.
9. Vérifiez que le répertoire **saml_folder** contient les fichiers suivants :

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

   Genesys CX Insights n'utilise pas le dossier personnalisé.

10. Exécutez la commande suivante pour créer une carte de configuration - gcxi-saml :

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

    Une fois cette opération terminée, supprimez **saml_folder**, car il n'est plus nécessaire.

11. Modifiez le fichier **gcxi.properties** en ajoutant les variables suivantes pour activer SAML et le définir comme mode de connexion par défaut :

    ```
    MSTR_WEB_SAML_ON=true
    MSTR_WEB_DEFAULT_LOGIN_MODE=64
    ```

12. Exécutez les commandes suivantes pour arrêter le pod gcxi :

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

    où :

    <values_file> — nom de votre fichier de valeurs YAML. Il s'agit généralement du même fichier que celui que vous avez utilisé lors du déploiement à l'aide des instructions figurant

sur : Genesys CX Insights - Kubernetes via Helm.

<release_name> — nom de la version Helm GCXI.

<path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

Par exemple :

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

Attendez que les pods soient arrêtés avant de poursuivre.

13. Exécutez les commandes suivantes pour mettre à niveau GCXI (en utilisant les valeurs de **gcxi.properties** pour définir les variables de conteneur) :

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

    où :

    <values_file> — nom de votre fichier de valeurs YAML. Il s'agit généralement du même fichier que celui que vous avez utilisé lors du déploiement à l'aide des instructions figurant sur : Genesys CX Insights - Kubernetes via Helm.

    <release_name> — nom de la version helm gcxi.

    <path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

    Par exemple :

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

    Attendez que GCXI soit installé.

14. Ouvrez MicroStrategy Web (http://<server>:8080/MicroStrategy/servlet/mstrWeb) ; vous serez automatiquement redirigé vers la page de connexion de l'IdP SAML.

Procedure: Activer SAML dans les déploiements qui utilisent des descripteurs Kubernetes

Purpose: Activez SAML dans les environnements dans lesquels vous déployez GCXI via des descripteurs Kubernetes.

Prerequisites

Avant de commencer :

- La version 9.0.019.00 ou une version ultérieure de GCXI est déployée dans votre environnement. GCXI fait office de fournisseur de services (SP) SAML.
- Vous avez accès à un fournisseur d'identité (IdP) SAML.
- Cette procédure suppose que l'espace de noms par défaut est **genesys**. (Si le vôtre ne l'est pas, vous devez ajouter **-n genesys** à toutes les commandes kubectl).

Steps

1. Exécutez la commande suivante pour effectuer une sauvegarde de la base de données méta GCXI :

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

2. Exécutez la commande suivante pour arrêter un conteneur GCXI :

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

3. Générez les fichiers de configuration SAML — suivez les instructions du site Web de MicroStrategy : Comment générer les fichiers de configuration, mais notez les points suivants :
   - Lorsque les instructions du site Web de MicroStrategy demandent une *URL de base d'entité*, utilisez : `http://<server>:8080/MicroStrategy/saml/config/open`.
   - Vous devez vous connecter à `http://<server>:8080/MicroStrategy/saml/config/open` en utilisant les informations d'identification de l'administrateur Tomcat. Vous pouvez changer le mot de passe de l'administrateur Tomcat en définissant la variable TOMCAT_ADMINPWD (reportez-vous à la Procédure : Configuration de secrets Kubernetes) ou contactez le service client pour obtenir de l'aide.

4. Exécutez la commande suivante et examinez les résultats pour vous assurer que les fichiers liés à SAML apparaissent dans le pod en cours d'exécution :

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

5. À un emplacement approprié, créez un répertoire vide appelé **saml_folder**. (Vous pouvez utiliser un autre nom, mais cette procédure suppose que vous avez utilisé le nom **saml_folder**).

6. Copiez les fichiers de configuration SAML du pod dans le répertoire saml_folder :

   ```
   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
   ```

   Vérifiez que les fichiers sont correctement copiés dans saml_folder et qu'aucun autre fichier ne figure dans le répertoire.

7. À l'aide du fichier **saml_folder/SPMetadata.xml**, enregistrez-vous auprès de l'IdP — les étapes

exactes varient suivant votre IdP.

8. Copiez le fichier **IDPMetadata.xml** fourni par votre IdP dans saml_folder.
9. Vérifiez que le répertoire saml_folder contient les fichiers suivants :

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

   Genesys CX Insights n'utilise pas le dossier personnalisé.

10. Exécutez la commande suivante pour créer une carte de configuration - gcxi-saml :

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

    Une fois cette opération terminée, supprimez saml_folder, car il n'est plus nécessaire.

11. Modifiez le fichier **gcxi.properties** en ajoutant les variables suivantes pour activer SAML et le définir comme mode de connexion par défaut :

    ```
    MSTR_WEB_SAML_ON=true
    MSTR_WEB_DEFAULT_LOGIN_MODE=64
    ```

12. Exécutez les commandes suivantes pour recréer la carte de configuration **gcxi-config** :

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

13. Modifiez le fichier **gcxi.yaml** et ajoutez le volume gcxi-saml et volumeMount à la définition des déploiements gcxi-secondary et gcxi-primary :

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

14. Exécutez les commandes suivantes pour recréer les pods gcxi et appliquer ainsi les changements **gcxi.yaml** :

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

    Attendez que GCXI soit installé.

15. Ouvrez MicroStrategy Web (http://<server>:8080/MicroStrategy/servlet/mstrWeb) ; vous serez automatiquement redirigé vers la page de connexion de l'IdP SAML.

Gestion de SAML

Si vous avez activé SAML, utilisez les informations de cette section pour le gérer.

- Désactiver SAML dans les déploiements qui utilisent Helm
- Désactiver SAML dans les déploiements qui utilisent des descripteurs Kubernetes

Procedure: Désactiver SAML dans les déploiements qui utilisent Helm

Purpose: Désactivez SAML lorsque GCXI est déployé via Helm.

Steps

1. Exécutez les commandes suivantes pour arrêter GCXI :

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

   où :

   <values_file> — nom de votre fichier de valeurs YAML. Il s'agit généralement du même fichier que celui que vous avez utilisé lors du déploiement à l'aide des instructions figurant sur : Genesys CX Insights - Kubernetes via Helm.

   <release_name> — nom de la version Helm GCXI.

   <path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

   Par exemple :

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

2. Exécutez les commandes suivantes pour mettre à niveau GCXI (en utilisant les valeurs de **gcxi.properties** pour définir les variables de conteneur) et redémarrer les pods :

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

   où :

   <values_file> — nom de votre fichier de valeurs YAML. Il s'agit généralement du même fichier que celui que vous avez utilisé lors du déploiement à l'aide des instructions figurant sur : Genesys CX Insights - Kubernetes via Helm.

   <release_name> — nom de la version Helm GCXI.

   <path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

Par exemple :

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

Attendez que GCXI soit mis à niveau.

Procedure: Désactiver SAML dans les déploiements qui utilisent des descripteurs Kubernetes

Purpose: Désactivez SAML lorsque GCXI est déployé à l'aide de descripteurs Kubernetes.

Prerequisites

Cette procédure suppose que l'espace de noms par défaut est **genesys**. (Si le vôtre ne l'est pas, vous devez ajouter **-n genesys** à toutes les commandes kubectl).

Steps

1. Modifiez le fichier **gcxi.properties** et supprimez les variables suivantes :

   ```
   MSTR_WEB_SAML_ON=true
   MSTR_WEB_DEFAULT_LOGIN_MODE=64
   ```

2. Exécutez les commandes suivantes pour recréer la carte de configuration gcxi-config :

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

3. Exécutez les commandes suivantes pour recréer les pods gcxi :

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

Activer et configurer LDAP

Pour activer LDAP, suivez l'une des procédures de cette section :

- Activer LDAP dans les déploiements qui utilisent Helm
- Activer LDAP dans les déploiements qui utilisent des descripteurs Kubernetes

Dans les deux cas, vous devez ensuite suivre la section Configurer LDAP.

Procedure: Activer LDAP dans les déploiements qui utilisent Helm

Purpose: Activez LDAP lorsque GCXI est déployé via Helm.

Prerequisites

Assurez-vous que la version 9.0.014.00 ou une version ultérieure de GCXI est déployée dans votre environnement.

Steps

Avant de commencer, procédez comme suit pour configurer GCXI :

1. Localisez la bibliothèque OpenLDAP à l'intérieur du conteneur :
    1. Exécutez la commande suivante pour vous assurer que la bibliothèque OpenLDAP est installée, et localisez le chemin d'accès à la bibliothèque :

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

    2. Copiez ou notez le chemin d'accès, par exemple, **/usr/lib64/libldap-2.4.so.2.**
2. Exécutez la commande suivante pour effectuer une sauvegarde de la base de données méta GCXI :

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

    où :

    <release_name> — nom de la version Helm GCXI.

    <path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

    Par exemple :

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

    Attendez que la tâche gcxi-backup soit terminée.
3. Exécutez la commande suivante pour empêcher la tâche de sauvegarde de s'exécuter à chaque mise à niveau de la version :

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

    où :

<release_name> — nom de la version Helm GCXI.

<path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

Par exemple :

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

4. Exécutez la commande suivante pour arrêter un conteneur :

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

   où :

   <release_name> — nom de la version Helm GCXI.

   <path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

   Par exemple :

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

   Attendez que les pods soient arrêtés avant de poursuivre.

5. Modifiez le fichier **gcxi.properties** et ajoutez les variables suivantes pour activer LDAP et le définir comme mode de connexion par défaut :

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

   N'ajoutez pas de commentaires, d'espaces supplémentaires ni de lignes vides.

6. Exécutez la commande suivante pour mettre à niveau et redémarrer GCXI (en utilisant les valeurs de **gcxi.properties** pour définir les variables de conteneur) et redémarrer les pods :

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

   où :

   <release_name> — nom de la version Helm GCXI.

   <path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

   Par exemple :

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

   Attendez que GCXI soit mis à niveau.

LDAP est maintenant activé dans GCXI et reste activé après le redémarrage du conteneur.

Procedure: Activer LDAP dans les déploiements qui utilisent des descripteurs Kubernetes

Purpose: Rassemblez les informations dont vous aurez besoin pour configurer LDAP et activer LDAP dans GCXI.

Prerequisites

- La version 9.0.014.00 ou une version ultérieure de GCXI est déployée dans votre environnement.
- Cette procédure suppose que l'espace de noms par défaut est **genesys**. (Si le vôtre ne l'est pas, vous devez ajouter **-n genesys** à toutes les commandes kubectl).

Steps

Avant de commencer, procédez comme suit pour configurer GCXI :

1. Localisez la bibliothèque OpenLDAP à l'intérieur du conteneur :
    1. Exécutez la commande suivante pour vous assurer que la bibliothèque OpenLDAP est installée, et localisez le chemin d'accès à la bibliothèque :

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

    2. Copiez ou notez le chemin d'accès, par exemple, **/usr/lib64/libldap-2.4.so.2.**
2. Activez LDAP :
    1. Exécutez la commande suivante pour effectuer une sauvegarde de la base de données méta GCXI :

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

    2. Exécutez les commandes suivantes pour arrêter les conteneurs en cours d'exécution :

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

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

    3. Modifiez le fichier **gcxi.properties** et définissez **MSTR_WEB_LDAP_ON=true**.
    4. Pour que LDAP soit le mode de connexion par défaut, modifiez le fichier **gcxi.properties** et définissez **MSTR_WEB_DEFAULT_LOGIN_MODE=16**.
    5. Exécutez les commandes suivantes pour charger gcxi.properties dans Kubernetes :

        ```
        kubectl delete configmap gcxi-config

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

    6. Exécutez la commande suivante pour démarrer le conteneur PRIMARY :

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

Attendez que PRIMARY soit terminé (attendez l'exécution de Tomcat et que la page MicroStrategyWeb soit disponible).

7. Exécutez la commande suivante pour démarrer le conteneur SECONDARY :

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

LDAP est maintenant activé dans GCXI et reste activé après le redémarrage du conteneur.

Si vous prévoyez d'utiliser LDAPS, vous devez également monter le dossier dans lequel résident vos certificats. Reportez-vous à la section LDAP avec SSL pour plus d'informations.

Configuration de LDAP

Pour les déploiements Helm et Kubernetes, utilisez les instructions de cette section pour configurer LDAP.

Procedure: Configurer les paramètres de connexion LDAP

Purpose: Pour établir une connexion au serveur LDAP.

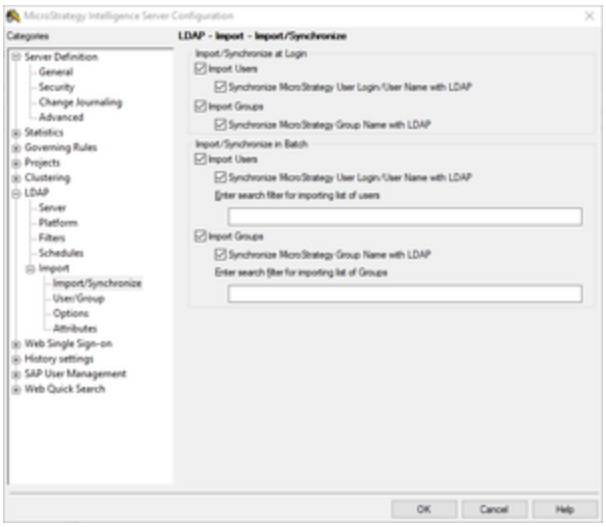
Steps

1. Connectez-vous à MicroStrategy Developer.
2. Cliquez sur **Administration** > **Serveur** > **Configurer MicroStrategy Intelligence Server**.
3. Cliquez sur **LDAP**, et, dans la section **Serveur**, fournissez les paramètres suivants :
   - **Hôte** : `<hostname_or_ip_of_ldap_server>`
   - **Port** : `<port_of_ldap_server>`
   - **Connexion de sécurité** : `Cleartext`
   - **Méthode d'authentification** : `Binding` (Obligatoire)
   - **Utilisateur de l'authentification** (Ce compte d'utilisateur est utilisé lorsque GCXI doit accéder à Active Directory, effectuer une recherche, répertorier tous les utilisateurs, etc. Il s'agit d'un « utilisateur de services »).
   - : Par exemple :
     - `DN : CN=<user_cn>,OU=<ou>,…, DC=<dc>,…`

    - Mot de passe : <mot de passe du domaine>

4. Cliquez sur **LDAP** > **Plate-forme**, et fournissez les paramètres suivants :
    - **Nom du fournisseur du serveur LDAP** : <votre fournisseur> (par exemple, Microsoft Active Directory)
    - **Pilote de connectivité LDAP** : <votre pilote de connectivité> (par exemple, Open LDAP)
    - **Plate-forme Intelligence Server** : Linux
    - **Noms des fichiers de connectivité LDAP** : <path_to_openldap_lib> (Il s'agit du chemin d'accès à l'intérieur du conteneur gcxi, que vous avez découvert à l'étape 1. Recueillez les informations et activez LDAP.)
5. Cliquez sur **LDAP** > **Filtres**, et fournissez les paramètres suivants :
    - **DN racine** : DC=<dc>,DC=<dc>… (Il s'agit du point de départ de la recherche LDAP)
    - **Filtre de recherche utilisateur** : <user_search> (La valeur que vous entrez ici dépend des paramètres LDAP. Par exemple, (&(objectclass=person)(sAMAccountName= #LDAP_LOGIN#)).)
    - **Filtre de recherche de groupe** : <group_search> (La valeur que vous entrez ici dépend des paramètres LDAP. Par exemple, (&(objectclass=group) (member=#LDAP_DN#)).)
    - **Nombre de niveaux de groupe supérieurs à importer** : 1
6. Pour vérifier que la connexion entre le serveur LDAP et Intelligence Server peut être établie, essayez de vous connecter à MicroStrategy Web (<IP>/MicroStrategy/servlet/mstrWeb) en utilisant les informations d'identification LDAP. Notez que même lorsque vous parvenez à vous connecter, les privilèges utilisateur ne sont pas encore définis. Les privilèges peuvent être attribués manuellement à chaque utilisateur dans mstrServerAdmin ou importés et configurés comme décrit à la section Importation de LDAP, synchronisation et liaison.

Importation de LDAP, synchronisation et liaison



LDAP - Importation

Vous devez configurer les privilèges des utilisateurs qui se connectent à l'aide des informations d'identification LDAP. Des informations détaillées sur cette étape sont disponibles dans le document MicroStrategy suivant : KB18506 : Importation et liaison des utilisateurs à l'aide de l'intégration LDAP avec MicroStrategy Intelligence Server 9.x et les versions plus récentes.

LDAP avec SSL

Pour prendre en charge SSL avec LDAP, vous devez exécuter le conteneur avec un dossier monté contenant les certificats LDAP et vous assurer que le dossier est monté sur tous les serveurs du cluster. Pour plus d'informations, reportez-vous à la section Comment configurer la connectivité LDAP via Texte en clair.

LDAP dans un conteneur

À partir de la version 9.0.014.00 de Genesys CX Insights, une option de configuration, **MSTR_WEB_LDAP_ON**, active LDAP pour MicroStrategy.

Gestion de LDAP

Si vous avez activé LDAP, utilisez les informations de cette section pour le gérer.

- Désactiver LDAP dans les déploiements qui utilisent Helm
- Désactiver LDAP dans les déploiements qui utilisent des descripteurs Kubernetes

Procedure: Désactiver LDAP dans les déploiements qui utilisent Helm

Purpose: Désactivez LDAP lorsque GCXI est déployé via Helm.

Steps

1. Exécutez les commandes suivantes pour arrêter GCXI :

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

   où :

   <values_file> — nom de votre fichier de valeurs YAML. Il s'agit généralement du même fichier que celui que vous avez utilisé lors du déploiement à l'aide des instructions figurant sur : Genesys CX Insights - Kubernetes via Helm.

   <release_name> — nom de la version Helm GCXI.

   <path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

   Par exemple :

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

2. Exécutez les commandes suivantes pour démarrer les pods GCXI :

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

   où :

   <values_file> — nom de votre fichier de valeurs YAML. Il s'agit généralement du même fichier que celui que vous avez utilisé lors du déploiement à l'aide des instructions figurant sur : Genesys CX Insights - Kubernetes via Helm.

   <release_name> — nom de la version Helm GCXI.

   <path_to_chart> — chemin d'accès aux fichiers de graphiques Helm.

   Par exemple :

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

Procedure: Désactiver LDAP dans les déploiements qui utilisent

des descripteurs Kubernetes

Purpose: Désactivez LDAP lorsque GCXI est déployé à l'aide de descripteurs Kubernetes.

Prerequisites

Cette procédure suppose que l'espace de noms par défaut est **genesys**. (Si le vôtre ne l'est pas, vous devez ajouter **-n genesys** à toutes les commandes kubectl).

Steps

1. Modifiez le fichier **gcxi.properties** et supprimez les variables suivantes :

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

2. Exécutez les commandes suivantes pour recréer la carte de configuration gcxi-config :

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

3. Exécutez les commandes suivantes pour recréer les pods gcxi :

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