

Connecteur UXP 2, version 1.1

Guide d'installation et de configuration

UXP-IG-CONNECTOR2

Table des matières

Notes de mise à jour de la dernière version de Connecteur UXP 2	1
1. Introduction	2
1.1. Public cible	2
1.2. Connecteur UXP 2	2
1.3. Références	3
2. Installation	4
2.1. Configuration requise	4
2.1.1. Exigences spécifiques aux composants	4
2.2. Configuration des ports	5
2.2.1. Ports externes	5
2.2.2. Routage interne des services	5
2.2.3. Notes complémentaires	6
2.3. Informations requises	6
2.4. Déploiement	7
3. Configuration après l'installation	9
3.1. Configuration et gestion des utilisateurs Casdoor	9
3.2. Remplacer les certificats TLS auto-signés	9
3.3. Ajouter ou modifier une licence dans Connecteur UXP 2	10
3.3.1. Ajouter ou mettre à jour la licence via l'interface utilisateur	10
3.3.2. Ajouter ou mettre à jour la licence via le CLI	10
3.4. Connexion aux bases de données SQL	11
3.4.1. Configurer la taille du pool de connexions à la base de données	12
3.5. Mettre à jour les informations d'identification du client OAuth	12
3.6. Seuil d'alerte pour l'expiration de la licence	13
4. Maintenance	15
4.1. Accéder aux journaux Docker	15
4.1.1. Suivre les journaux en temps réel	15
4.1.2. Rotation et gestion des journaux Docker	15
4.2. Fichiers journaux Connecteur UXP 2	16
4.3. Journaux de l'interface utilisateur Casdoor	16
4.4. Gérer les conteneurs Docker	16

4.5. Surveillance des performances des applications	18
4.6. Procédures de sauvegarde et de restauration	18
4.6.1. Vérifier la structure de la base de données (facultatif)	18
4.6.2. Créer une sauvegarde de base de données	18
4.6.3. Restaurer une sauvegarde de base de données	19
4.6.4. Meilleures pratiques pour les sauvegardes	20
Annexe A: Notes de mise à jour Connecteur UXP 2	21

Notes de mise à jour de la dernière version de Connecteur UXP 2

1.1.0 (12.2025)

- Prise en charge d'Ubuntu 24, optimisations des performances, prise en charge de la base de données Oracle.

Pour toutes les notes de mise à jour de Connecteur UXP 2, voir l'[Annexe](#).

1. Introduction

1.1. Public cible

Ce guide s'adresse aux administrateurs responsables de l'installation du logiciel Connecteur UXP 2. La fonctionnalité du logiciel est couverte par le Guide de l'utilisateur Connecteur UXP 2 disponible dans l'interface Web Connecteur UXP 2.

Ce document s'adresse à des lecteurs ayant une bonne connaissance de la gestion des serveurs Linux, des réseaux informatiques et des principes de fonctionnement d'UXP.

1.2. Connecteur UXP 2

Connecteur UXP 2 est un outil permettant de mettre en œuvre des services Web compatibles avec le protocole UXP.

Connecteur UXP 2 :

- convertit les demandes REST entrantes en instructions SQL ;
- exécute les instructions sur les bases de données SQL spécifiées ;
- et convertit les ensembles de résultats de la base de données en réponses REST sortantes.

Avec Connecteur UXP 2, le développeur peut rapidement mettre en œuvre les services UXP en écrivant des demandes de base de données ou des instructions d'insertion/mise à jour. Pour les cas plus complexes, il est possible d'implémenter la logique applicative sous forme de procédure stockée pouvant être invoquée par Connecteur 2.

Connecteur UXP 2 (ci-après dénommé Connecteur) se compose des éléments suivants, installés dans des conteneurs Docker distincts :

- API de configuration du Connecteur ;
- Interface utilisateur React (UI) ;
- PostgreSQL ;
- API Rest-SQL ;
- Redis ;
- Casdoor.

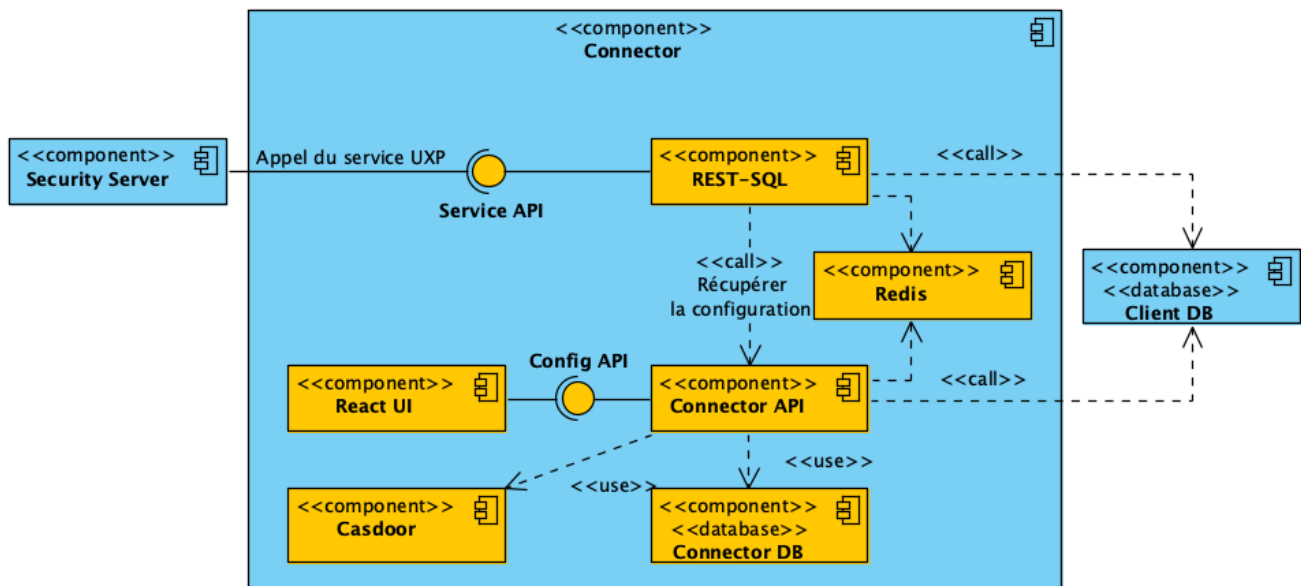


Figure 1. Diagramme des composants de Connecteur UXP 2

1.3. Références

- [CASDOOR] <https://casdoor.org/>
- [UXP-UG-CONNECTOR2] Cybernetica AS. Connecteur UXP 2 : Guide de l'utilisateur. Identifiant du document : UXP-UG-CONNECTOR2

2. Installation

2.1. Configuration requise

Plates-formes prises en charge

Le système d'exploitation recommandé est Ubuntu Server 24.04 Long-Term Support (LTS) sur une plate-forme **64 bits**.

Configuration minimale requise

CPU 2 cœurs 64-bit x86

Installation du serveur Ubuntu 24.04 64 bits

Au moins 2 Go de RAM

Au moins 20 Go d'espace de stockage

Interface réseau 10/100 Mbps ou mieux

Configuration système recommandée

CPU 4 cœurs 64-bit x86

Installation du serveur Ubuntu 24.04 64 bits

4 Go de RAM ;

Au moins 30 Go d'espace de stockage

Interface réseau 10/100 Mbps ou mieux

2.1.1. Exigences spécifiques aux composants

Les composants suivants ont été développés spécifiquement pour Connecteur UXP 2 ou sont intégrés à celui-ci. Veillez à ce que les exigences en matière de mémoire pour chaque composant soient respectées pour une performance optimale :

- **API de configuration du Connecteur**

- Mémoire : *Au minimum 500 Mo de RAM sont recommandés pour des performances optimales*

- **React UI**

- Mémoire : *Au minimum 10 Mo de RAM sont recommandés pour des performances optimales*

- **Rest-SQL API**

- Mémoire : *Au minimum 300 Mo de RAM sont recommandés pour des performances optimales*

- **PostgreSQL**

- Mémoire : *Au minimum 100 Mo de RAM sont recommandés pour des performances optimales*

- **Redis**

- Mémoire : *Au minimum 10 Mo de RAM sont recommandés pour des performances optimales*

Les outils tiers suivants sont fournis avec Connecteur et sont inclus ici en tant qu'exigences distinctes :

- **Casdoor**

- Mémoire : *Au minimum 20 Mo de RAM sont recommandés pour des performances optimales*



Veillez à ce que chacun de ces composants dispose de suffisamment de mémoire pour éviter les problèmes de performance.

2.2. Configuration des ports

Connecteur expose deux ports externes. Tous les autres services sont acheminés en interne via un proxy inverse NGINX, ce qui permet d'avoir une interface propre et une sécurité accrue.

2.2.1. Ports externes

Port	Protocole	Description
22	TCP	Accès SSH au système hôte (pour les administrateurs)
4400	TCP	React UI (interface principale pour les utilisateurs)
4401	TCP	Service d'authentification Casdoor (gère la connexion, l'enregistrement et l'authentification unique des utilisateurs)

2.2.2. Routage interne des services

Tous les services backend sont accessibles via des chemins spécifiques sous le port principal de l'interface utilisateur (4400) ou le port d'authentification (4401). Ceux-ci ne sont pas exposés en tant que ports autonomes mais routés à travers NGINX en utilisant les noms d'hôtes internes des conteneurs Docker.

Port 4400 (interface Web principale)

- Le port de base 4400 dessert l'interface React et transmet les appels API aux services backend.

Chemin	Transmis à	Description
/	Fichiers statiques servis par <code>/usr/share/nginx/html</code>	Interface utilisateur principale basée sur React. Routes React gérées via <code>index.html</code> .
/docs/	Fichiers de <code>/usr/share/nginx/html/docs/</code>	Documentation utilisateur statique, par défaut <code>uxp-ug-connector2_uxp_connector2_user_guide.html</code> .
/api/	http://uxp-connector:8082/api/	Achemine les demandes d'API du frontend vers l'API interne de configuration du Connecteur.
/rest/	http://rest-sql:8081/	Transmet les requêtes SQL RESTful à l'API REST-SQL interne.

Port 4401 (authentification Casdoor)

- Le port 4401 est dédié à l'interface d'identité et d'authentification Casdoor.

Chemin	Transmis à	Description
/	http://casdoor:8000/	Interface Web Casdoor et API pour l'authentification et la gestion des utilisateurs.

2.2.3. Notes complémentaires

- Toutes les communications entre les services internes se font par l'intermédiaire de noms de conteneurs Docker (par exemple, `uxp-connector`, `rest-sql`, `casdoor`).
- Aucun autre port (par exemple 8081, 8082, etc.) n'est exposé au système hôte ou au public.
- Cette conception réduit la surface d'attaque externe et simplifie la configuration du pare-feu.



En outre, les connexions sortantes requises pour se connecter aux bases de données SQL à l'aide de services supplémentaires (DNS, NTP, SSH) et pour installer des mises à jour logicielles (port 80/TCP) doivent être autorisées.

2.3. Informations requises

Informations d'identification pour le téléchargement de logiciels et de scripts

Pour télécharger le logiciel Connecteur UXP 2 depuis le registre Docker UXP ainsi que le script d'installation depuis le dépôt sécurisé, vous aurez besoin d'informations d'identification valides.

Licence Connecteur UXP 2

Connecteur UXP 2 nécessite une licence valide pour fonctionner.

Informations spécifiques à Connecteur UXP 2

Au cours de l'installation, l'administrateur du serveur sera invité à attribuer un nom d'utilisateur et un mot de passe au compte administrateur principal.

(Ces informations d'identification sont requises pour la connexion initiale à l'interface Connecteur UXP 2 et à l'interface Casdoor. Des comptes d'utilisateurs supplémentaires peuvent être créés ultérieurement si nécessaire.)

2.4. Déploiement

Pour déployer Connecteur UXP 2, téléchargez et exécutez le script d'installation. Ce script automatise l'ensemble du processus d'installation, y compris l'extraction des images Docker, la configuration des conteneurs, la génération des certificats TLS et l'assurance que tous les services sont en cours d'exécution.



Assurez-vous que le paquet `ca-certificates` est installé sur votre système. Sans celui-ci, les téléchargements sécurisés (tels que la récupération du script d'installation à l'aide de `wget`) risquent d'échouer.

Si nécessaire, vous pouvez l'installer ou le corriger à l'aide des commandes suivantes :

```
sudo apt install ca-certificates
sudo update-ca-certificates
```

Procédez aux étapes suivantes pour exécuter le script d'installation :

1. Ouvrez votre terminal

Commencez par ouvrir l'interface de ligne de commande (CLI) ou le terminal de votre système.

2. Téléchargez le script d'installation

Utilisez la commande suivante pour télécharger le script à partir du dépôt sécurisé (il sera enregistré dans votre répertoire de travail actuel) :

```
wget --user=<your-username> --password='<your-password>' https://docker-registry-ext.cyber.ee/scripts/c2_installation_script.sh
```



Remplacez `<your-username>` et `<your-password>` par les informations d'identification qui vous ont été fournies.

3. Rendez le script exécutable

Assurez-vous que le script dispose des autorisations nécessaires :

```
chmod +x c2_installation_script.sh
```

4. Exécutez le script

Exécutez le script pour lancer le déploiement automatisé :

```
./c2_installation_script.sh
```

Pendant l'exécution du script, le script d'installation :

- Vérifie la présence de paquets debian `uxp-*` installés sur le serveur.
- Vérifie l'état de préparation du système (installation de Docker, version et autorisations).
- S'authentifie auprès du registre Docker et extrait les images de conteneurs nécessaires.
- Applique les paramètres de configuration nécessaires, y compris le nom d'hôte et les variables d'environnement.
- Génère des certificats TLS auto-signés pour une communication sécurisée.
- Permet la personnalisation du nom de l'organisation, des identifiants utilisateur et de l'accès administratif.
- Déploie les conteneurs et s'assure qu'ils fonctionnent correctement.
- Résume le processus d'installation et fournit une vue d'ensemble de l'état du système.



Lorsque vous accédez pour la première fois à l'interface utilisateur de Connecteur UXP 2 à l'adresse <https://<PUBLIC-IP-OR-HOSTNAME>:4400>, vous pouvez rencontrer des avertissements de sécurité liés aux certificats auto-signés. C'est tout à fait normal. Les utilisateurs doivent valider manuellement le certificat dans leur navigateur. Si vous utilisez des certificats auto-signés, rendez-vous dans le panneau d'administration Casdoor <https://<PUBLIC-IP-OR-HOSTNAME>:4401>, acceptez le certificat, puis réessayez.



Le script d'installation génère des certificats TLS auto-signés adaptés aux environnements de développement ou de test. Pour les environnements de production, il est fortement recommandé de remplacer les certificats auto-signés par des certificats valides et fiables. Cette étape est toutefois facultative si vous utilisez l'application à des fins de développement ou de test.

Voir [Remplacer les certificats TLS auto-signés](#) pour des instructions détaillées.



La modification des fichiers de configuration peut nécessiter des privilèges élevés. Utilisez `sudo` si vous n'êtes pas connecté en tant qu'utilisateur `root`. Pour les commandes Docker, `sudo` n'est pas nécessaire si votre utilisateur fait partie du groupe `docker`.

3. Configuration après l'installation

3.1. Configuration et gestion des utilisateurs Casdoor

Après l'installation, il est recommandé de terminer la configuration de Casdoor, en particulier pour les environnements de production. Cela comprend :

- **Définir les informations complémentaires sur l'organisation** : Vous pouvez mettre à jour l'organisation créée lors de l'installation afin d'y inclure tous les détails nécessaires.
- **Gérer les comptes d'utilisateurs** : Ajoutez ou modifiez des comptes d'utilisateurs, changez les noms d'utilisateurs, définissez des e-mails ou attribuez des rôles en fonction de vos besoins organisationnels.
- **Activer la récupération du mot de passe** : Pour permettre aux utilisateurs de réinitialiser leur mot de passe par e-mail, configurez un fournisseur de messagerie électronique et associez-le aux applications concernées.

Casdoor gère l'authentification et la gestion des utilisateurs pour le Connecteur. Pour des conseils spécifiques au Connecteur, voir la section « 8. Gérer les comptes d'utilisateurs » dans [\[UXP-UG-CONNECTOR2\]](#). Pour obtenir des informations générales sur Casdoor, consultez la documentation officielle : [\[CASDOOR\]](#).

3.2. Remplacer les certificats TLS auto-signés

Par défaut, le script d'installation génère des certificats TLS auto-signés adaptés aux environnements de développement ou de test.

Si vous effectuez un déploiement dans un environnement de production, nous vous recommandons vivement de remplacer ces certificats auto-signés par des certificats valides et fiables.

Suivez les étapes ci-dessous si vous choisissez de remplacer les certificats de développement par défaut par vos certificats de production :

1. Obtenez un certificat SSL/TLS valide auprès d'une autorité de certification (AC) de confiance.
2. Placez les fichiers de certificat et de clé dans le répertoire suivant :

```
/etc/uxp/uxp-connector/nginx/ssl/
```

Veillez à ce que les fichiers soient nommés exactement comme suit :

- `cert.pem` – votre certificat public
- `key.pem` – votre clé privée

Le fait de conserver ces noms de fichiers garantit qu'aucune modification supplémentaire n'est nécessaire dans la configuration Nginx.

3. Redémarrez les conteneurs Docker pour appliquer les nouveaux certificats. Exécutez la commande suivante à partir du répertoire `/etc/uxp/connector-v2/`, qui contient votre fichier `docker-compose.yml` :

```
docker compose restart
```

Cette commande redémarre tous les services définis dans la configuration de Docker Compose, en rechargeant les nouveaux fichiers de certificats sans qu'il soit nécessaire d'arrêter complètement les conteneurs.

3.3. Ajouter ou modifier une licence dans Connecteur UXP 2

Cette section couvre les étapes pour ajouter ou changer la licence dans Connecteur UXP 2 en utilisant deux méthodes : l'interface utilisateur de Connecteur UXP 2 et l'interface de ligne de commande (CLI).

3.3.1. Ajouter ou mettre à jour la licence via l'interface utilisateur

1. **Accédez à l'interface utilisateur de Connecteur UXP 2** : Ouvrez votre navigateur Web et accédez à l'interface utilisateur de Connecteur UXP 2 à l'adresse <https://<PUBLIC-IP-OR-HOSTNAME>:4400>.
2. **Connexion** : Entrez vos informations d'identification pour vous connecter.
3. **Accédez à Gestion des licences** : Sous **Configuration**, cliquez sur **Licence**.
4. **Téléchargez ou modifiez la licence** :
 - Si vous ajoutez une nouvelle licence, cliquez sur **Télécharger la licence**. Parcourez votre ordinateur local pour sélectionner le nouveau fichier de licence, puis cliquez sur **Télécharger**.
 - Si vous modifiez une licence existante, la procédure est la même que pour l'ajout d'une licence. Vous pouvez télécharger un nouveau fichier de licence qui remplacera le fichier existant.
5. **Vérification** : Si tout se passe bien, un message apparaîtra indiquant : « *La licence a été vérifiée avec succès.* »
6. **Enregistrez les modifications** : Après vérification, cliquez sur **Enregistrer** pour appliquer et stocker la nouvelle configuration de licence. Un message apparaîtra indiquant : « *La licence a été téléchargée avec succès.* »



La licence peut limiter les membres UXP autorisés à utiliser Connecteur UXP 2. Dans ce cas, Connecteur UXP 2 n'accepte que les demandes envoyées aux membres UXP dont les identifiants répondent aux critères fixés pour les fournisseurs de services agréés.

3.3.2. Ajouter ou mettre à jour la licence via le CLI

1. **Copiez le fichier de licence sur le serveur** : Utilisez `scp` ou toute autre méthode pour transférer le fichier de licence vers le serveur.

2. **Copiez la licence dans le conteneur UXP Connector 2 Configuration API** : Une fois que les conteneurs fonctionnent, copiez le fichier de licence dans le conteneur `uxp-connector` à l'aide de la commande suivante :

```
docker cp </path/to/your/license.lic> connector-v2-uxp-connector-1:/etc/uxp/connector-v2/license.lic
```

Remplacez `</path/to/your/license.lic>` par l'emplacement réel de votre fichier de licence.

3. **Redémarrez le conteneur** : Redémarrez le conteneur pour appliquer la licence. Pour ce faire, exécutez la commande suivante à partir du répertoire `/etc/uxp/connector-v2/`, qui contient votre fichier `docker-compose.yml` :

```
docker compose restart uxp-connector
```

4. **Confirmez la mise à jour de la licence** : Après le redémarrage du conteneur, suivez les mêmes étapes que celles décrites aux points 1 à 3 de la section [Ajouter ou mettre à jour la licence via l'interface utilisateur](#). Ensuite, consultez le panneau **Détails de la licence** dans l'interface utilisateur pour vérifier que la nouvelle licence a été appliquée avec succès.



Lorsque vous ajoutez ou remplacez le fichier de licence dans le répertoire du conteneur `uxp-connector /etc/uxp/connector-v2/`, veillez à ce que le fichier soit nommé **license.lic**. Cette convention de nom est nécessaire pour que l'application reconnaisse le fichier de licence.

3.4. Connexion aux bases de données SQL

Connecteur UXP 2 nécessite des pilotes JDBC pour établir des connexions aux bases de données SQL.

Les pilotes pour **PostgreSQL**, **Microsoft SQL Server**, **MariaDB** et **OracleDB** sont inclus dans Connecteur UXP 2.

Les pilotes JDBC inclus sont des logiciels tiers distribués sous différentes licences.

Si vous avez besoin d'aide pour d'autres bases de données, veuillez contacter le service clientèle.

Tableau 1. Pilotes JDBC inclus dans Connecteur

Base de données	Pilote JDBC	Licence
PostgreSQL	PostgreSQL JDBC v42.7.5	Licence BSD à deux clauses
Microsoft SQL Server	Pilote Microsoft JDBC pour SQL Server v12.8.1	Licence MIT
MariaDB	MariaDB Connector/J v3.5.2	GNU LGPL

Base de données	Pilote JDBC	Licence
OracleDB	Pilote Oracle JDBC v23.26.0.0.0	Oracle FUTC

3.4.1. Configurer la taille du pool de connexions à la base de données

Par défaut, chaque connexion à une base de données SQL utilise un pool de connexion d'une taille maximale de 20.

Pour modifier cette valeur :

1. Ouvrez le fichier de configuration suivant sur le serveur :

```
/etc/uxp/connector-v2/restsql-config/application.properties
```

2. Ajoutez ou mettez à jour la propriété suivante :

```
hikari.db.maxPoolSize=20
```

3. Enregistrez les modifications
4. Redémarrez le conteneur REST-SQL API pour appliquer les modifications. Pour ce faire, exécutez la commande suivante à partir du répertoire `/etc/uxp/connector-v2/`, qui contient votre fichier `docker-compose.yml` :

```
docker compose restart rest-sql
```

3.5. Mettre à jour les informations d'identification du client OAuth

Par défaut, le script d'installation configure les informations d'identification du client OAuth utilisées par Connecteur UXP 2 dans le système d'authentification. Si vous devez modifier les adresses `Client ID` et `Client secret`, vous devez mettre à jour l'interface utilisateur de Casdoor et les fichiers de configuration de Connecteur pour garantir un comportement cohérent.

Mise à jour via l'interface utilisateur Casdoor :

1. Accédez au panneau d'administration Casdoor : <https://<PUBLIC-IP-OR-HOSTNAME>:4401>
2. Connectez-vous avec les identifiants administrateur.
3. Naviguez vers Identity → Applications.
4. Sélectionnez l'application utilisée par Connecteur UXP 2.
5. Mettez à jour les champs `Client ID` et `Client secret` avec les valeurs souhaitées.
6. Enregistrez les modifications.

Mettre à jour la configuration de Connecteur UXP 2 :

1. Ouvrez les fichiers de configuration suivants sur le serveur :

```
/etc/uxp/connector-v2/connector-config/application.properties
/etc/uxp/connector-v2/web-config/config.json
```

2. Mettez à jour les entrées suivantes pour qu'elles correspondent aux valeurs définies dans l'interface utilisateur Casdoor :

Dans `application.properties`, modifiez les lignes suivantes :

```
spring.security.oauth2.client.registration.casdoor.client-id=<your-new-client-id>
spring.security.oauth2.client.registration.casdoor.client-secret=<your-new-client-secret>
```

Dans `config.json`, modifiez les entrées suivantes :

```
{
  "OAUTH_CLIENT_ID": "<your-new-client-id>",
}
```

3. Redémarrer les conteneurs Connecteur UXP 2 avec les changements de configuration appliqués. Exécutez la commande suivante à partir du répertoire `/etc/uxp/connector-v2/`, qui contient votre fichier `docker-compose.yaml` :

```
docker compose down
docker compose up -d
```

Cette opération permet d'arrêter et de démarrer proprement tous les services, en veillant à ce que la nouvelle configuration soit appliquée.

3.6. Seuil d'alerte pour l'expiration de la licence

Le paramètre de configuration `DAYS_BEFORE_LICENSE_EXPIRY_WARNING` définit le nombre de jours avant le déclenchement de l'avertissement d'expiration de la licence. Ce paramètre vous permet de définir un seuil à partir duquel l'interface utilisateur de Connecteur UXP 2 informera les utilisateurs de l'expiration imminente de la licence.

Par défaut, le système émet un avertissement 32 jours avant l'expiration de la licence. Vous pouvez adapter cette valeur aux besoins de votre environnement.

Pour configurer ce paramètre :

1. Ouvrez le fichier de configuration suivant sur le serveur :

```
/etc/uxp/connector-v2/web-config/config.json
```

2. Modifiez la valeur `DAYS_BEFORE_LICENSE_EXPIRY_WARNING` :

```
{
```



```
"DAYS_BEFORE_LICENSE_EXPIRY_WARNING": <desired number of days>
}
```

3. Redémarrez les conteneurs Docker pour appliquer la nouvelle valeur. Exécutez la commande suivante à partir du répertoire `/etc/uxp/connector-v2/`, qui contient votre fichier `docker-compose.yaml` :

```
docker compose restart
```

Cette opération permet d'arrêter et de démarrer proprement tous les services, en veillant à ce que la nouvelle configuration soit appliquée.

4. Maintenance

Cette section traite des tâches de maintenance essentielles pour garantir le bon fonctionnement de l'application.

4.1. Accéder aux journaux Docker

Les informations de journalisation de vos conteneurs Docker sont accessibles à l'aide de la commande suivante :

```
docker logs <container-name>
```

Remplacez `<container-name>` par le nom réel du conteneur que vous souhaitez inspecter. Cette commande affiche les journaux du conteneur depuis son démarrage.

4.1.1. Suivre les journaux en temps réel

Pour suivre les journaux en temps réel et surveiller l'activité au fur et à mesure qu'elle se produit, utilisez l'indicateur `-f` :

```
docker logs -f <container-name>
```

Cette commande permet d'afficher un flux en direct des journaux au fur et à mesure qu'ils sont générés. Appuyez sur `Ctrl+C` pour arrêter de suivre les journaux.

4.1.2. Rotation et gestion des journaux Docker

Docker génère des journaux en continu, ce qui peut consommer de l'espace disque au fil du temps. Il est important de configurer la rotation des journaux pour gérer cela. Vous pouvez ajouter des options de journalisation à votre fichier `docker-compose.yml` dans la configuration du service afin de limiter la taille des fichiers journaux et de gérer la fréquence de rotation.

Exemple :

```
services:
  my-service:
    logging:
      driver: "json-file"
      options:
        max-size: "10m"
        max-file: "3"
```

Dans cet exemple :

- `max-size`: Limite chaque fichier journal à 10 Mo.
- `max-file`: Limite le nombre de fichiers journaux à 3. Une fois la limite atteinte, le fichier

journal le plus ancien est supprimé.

4.2. Fichiers journaux Connecteur UXP 2

Tableau 2. Fichiers journaux de Connecteur UXP 2.

Emplacement du journal	Description	Configuration du journal
/var/log/uxp-connector-v2/uxp-connector-audit.log	Enregistrement des actions réussies et échouées de l'utilisateur dans l'interface utilisateur Connecteur UXP 2	/etc/uxp/connector-v2/connector-config/connector-logback.xml
/var/log/uxp-connector-v2/uxp-connector-application.log	Enregistrements des demandes adressées au serveur d'application	/etc/uxp/connector-v2/connector-config/connector-logback.xml
/var/log/uxp-connector-v2/uxp-connector-rest.log	Enregistrements du traitement des transactions SQL2REST	/etc/uxp/connector-v2/restsql-config/restsql-logback.xml
/var/log/uxp-connector-v2/connector-v2-installation.log	Enregistrements des actions liées au processus d'installation	N/A



Notez que la lecture de /var/log/uxp-connector-v2/connector-v2-installation.log nécessite les privilèges de l'administrateur (root).

4.3. Journaux de l'interface utilisateur Casdoor

Casdoor propose des fonctions intégrées de journalisation et d'audit accessibles à partir de l'onglet **Logging & Auditing** dans l'interface utilisateur de Casdoor. Ces journaux suivent les événements d'authentification, l'activité des utilisateurs et les actions au niveau du système dans Casdoor.

Consultez ces journaux si vous rencontrez des problèmes liés à l'authentification des utilisateurs ou à la gestion des rôles.

4.4. Gérer les conteneurs Docker

Pour gérer efficacement vos conteneurs et images Docker, voici quelques commandes couramment utilisées :

1. Exécuter des conteneurs :

```
docker compose up -d
```

2. **Redémarrer les conteneurs :**

```
docker compose restart
```

3. **Arrêter des conteneurs :** Si vous avez besoin d'arrêter les conteneurs en cours d'exécution, vous pouvez utiliser :

```
docker compose down
```

4. **Voir les conteneurs en cours d'exécution :** Utilisez cette commande pour voir les conteneurs en cours d'exécution :

```
docker ps
```

5. **Voir tous les conteneurs :** Pour afficher tous les conteneurs, y compris ceux qui sont arrêtés :

```
docker ps -a
```

6. **Afficher les images Docker :** Pour lister toutes les images Docker sur votre machine locale :

```
docker images
```

7. **Supprimer les images inutilisées :** Pour supprimer les images flottantes et les images inutilisées :

```
docker image prune
```

ou supprimer toutes les images inutilisées :

```
docker image prune -a
```

8. **Inspecter un conteneur :** Pour obtenir des informations détaillées sur un conteneur :

```
docker inspect <container-name>
```

9. **Afficher les volumes Docker :** Pour voir les volumes disponibles dans Docker :

```
docker volume ls
```

10. **Supprimer un volume :** Pour supprimer un ou plusieurs volumes :

```
docker volume rm <volume-name>
```

11. **Voir les réseaux Docker :** Pour répertorier tous les réseaux Docker :

```
docker network ls
```

12. **Inspecter un réseau** : Pour obtenir des informations détaillées sur un réseau :

```
docker network inspect <network-name>
```

13. **Supprimer un réseau** : Pour supprimer un ou plusieurs réseaux :

```
docker network rm <network-name>
```

4.5. Surveillance des performances des applications

Vous pouvez surveiller l'utilisation des ressources de vos conteneurs à l'aide de la commande suivante :

```
docker stats
```

Cette commande permet de visualiser en temps réel l'utilisation des ressources de vos conteneurs, notamment le processeur, la mémoire, les entrées/sorties réseau et les entrées/sorties par bloc. Elle peut être particulièrement utile pour identifier les goulets d'étranglement en matière de performances et s'assurer que vos conteneurs fonctionnent efficacement.

4.6. Procédures de sauvegarde et de restauration

Il est essentiel de sauvegarder régulièrement vos bases de données et vos fichiers de configuration pour préserver l'intégrité des données et les options de récupération. Suivez ces étapes pour créer une sauvegarde et restaurer votre base de données en utilisant PostgreSQL dans un conteneur Docker.

4.6.1. Vérifier la structure de la base de données (facultatif)

Avant ou après avoir effectué une sauvegarde ou une restauration, il se peut que vous souhaitiez vérifier la structure de votre base de données (par exemple, en vérifiant la liste des tables). Vous pouvez le faire en exécutant la commande suivante :

```
docker exec -it <db-container-name> psql -U <db-user> -d <db-name> -c '\dt'
```

Cette commande répertorie toutes les tables de la base de données spécifiée. Vous pouvez l'exécuter **avant** de lancer le processus de sauvegarde pour vérifier l'état actuel de la base de données ou **après** la restauration pour vérifier que les données et les tables ont été restaurées correctement.



Cette étape est facultative. Elle est utile si vous souhaitez confirmer visuellement les tables de votre base de données à n'importe quel moment du processus.

4.6.2. Créer une sauvegarde de base de données

Pour créer une sauvegarde de votre base de données PostgreSQL, utilisez la commande `pg_dump`. Cette commande génère un fichier dump contenant les commandes SQL nécessaires pour recréer la base de données. Voici comment procéder :

1. **Identifiez votre conteneur de base de données :** Assurez-vous de connaître le nom du conteneur qui exécute votre base de données PostgreSQL. Vous pouvez le trouver en utilisant :

```
docker ps
```

2. **Créez la sauvegarde :** Utilisez la commande suivante pour créer une sauvegarde. Remplacez `<db-container-name>`, `<db-user>`, `<db-name>`, et `/path/to/backup/backup-file.dump` par le nom de votre conteneur de base de données, l'utilisateur de la base de données, le nom de la base de données et le chemin d'accès au fichier de sauvegarde souhaité, respectivement :

```
docker exec <db-container-name> pg_dump -U <db-user> -F c <db-name> > /path/to/backup/backup-file.dump
```

3. **Vérifiez la sauvegarde :** Assurez-vous que le fichier de sauvegarde a été créé avec succès en vérifiant le répertoire de sauvegarde spécifié :

```
ls -lh /chemin/vers/sauvegarde/
```

4.6.3. Restaurer une sauvegarde de base de données

Pour restaurer une base de données PostgreSQL à partir d'un fichier dump, procédez comme suit :

1. **Identifiez votre conteneur de base de données :** Comme pour le processus de sauvegarde, assurez-vous de connaître le nom du conteneur qui exécute votre base de données PostgreSQL.
2. **Copiez le fichier de sauvegarde dans le conteneur :**

Utilisez `docker cp` pour copier le fichier de sauvegarde dans le conteneur. Remplacez les espaces réservés en conséquence :

```
docker cp /path/to/backup/backup-file.dump <db-container-name>:/var/tmp/backup-file.dump
```

3. **Restaurez la sauvegarde :** Utilisez la commande `pg_restore` pour restaurer votre base de données à partir du fichier dump. Remplacez `<db-container-name>`, `<db-user>`, `<db-name>`, et `/var/tmp/backup-file.dump` par vos valeurs réelles :

```
docker exec <db-container-name> pg_restore --clean --if-exists -U <db-user> -d <db-name> /var/tmp/backup-file.dump
```

Explication des options :

- `docker exec`: Exécute une commande dans le conteneur spécifié.
- `-it`: Permet d'interagir avec le shell du conteneur (utile pour des commandes comme `psql`).
- `<db-container-name>`: Le nom de votre conteneur PostgreSQL.
- `psql`: Le client de ligne de commande PostgreSQL.
- `-U <db-user>`: Spécifie l'utilisateur PostgreSQL sous lequel se connecter. Cet utilisateur est défini à la fois dans la configuration de votre application et dans l'environnement du conteneur Docker PostgreSQL :
 - Dans `/etc/uxp/connector-v2/connector-config/application.properties`, le paramètre `spring.datasource.username=<db-user>` spécifie l'utilisateur au niveau de l'application.
 - Dans `/etc/uxp/connector-v2/docker-compose.yaml`, la variable d'environnement `POSTGRES_USER=<db-user>` est utilisée pour configurer l'utilisateur du conteneur PostgreSQL au démarrage du conteneur.
- `-F c`: Spécifie le format du fichier de sauvegarde en tant qu'archive personnalisée.
- `<db-name>`: Le nom de la base de données PostgreSQL que vous souhaitez sauvegarder ou restaurer. Ceci est défini à la fois dans la configuration de votre application et dans l'environnement du conteneur Docker PostgreSQL :
 - Dans `/etc/uxp/connector-v2/connector-config/application.properties`, le paramètre `spring.datasource.url=jdbc:postgresql://postgres:5432/<db-name>` spécifie le nom de la base de données pour l'application.
 - Dans `/etc/uxp/connector-v2/docker-compose.yaml`, la variable d'environnement `POSTGRES_DB=<db-name>` est utilisée pour configurer le nom de la base de données du conteneur PostgreSQL au démarrage.
- `--clean --if-exists`: Supprime les objets de la base de données avant de les recréer, uniquement s'ils existent.

4.6.4. Meilleures pratiques pour les sauvegardes

- **Planifiez des sauvegardes régulières** : Automatisez le processus de sauvegarde à l'aide d'une tâche cron ou d'un outil de planification similaire.
- **Stockez les sauvegardes en toute sécurité** : Conservez les sauvegardes dans un endroit sûr, idéalement hors site ou sur une solution de stockage en nuage.
- **Surveillez l'espace disque** : Veillez à disposer d'un espace disque suffisant pour stocker les sauvegardes et surveillez l'utilisation qui en est faite au fil du temps.

Annexe A: Notes de mise à jour

Connecteur UXP 2

1.1.0 (12.2025)

- Prise en charge d'Ubuntu 24, optimisations des performances, prise en charge de la base de données Oracle.

1.0.0 (04.2025)

- Première version de Connecteur UXP 2.

1.0.0-alpha (09.2024)

- Une version alpha de Connecteur UXP 2.