Déployer Key Connector

Déployer Key Connector

Cet article vous guidera à travers la procédure pour activer et configurer Key Connector dans un environnement existant auto-hébergé. **Avant de continuer** , veuillez lire attentivement l'article à propos de Key Connector pour garantir une compréhension complète de ce qu'est Key Connector, de son fonctionnement et des impacts de sa mise en œuvre.

Bitwarden prend en charge le déploiement d'un Key Connector pour une utilisation par une organisation pour une instance auto-hébergée.

Exigences

> ⚠ **Warning**
>
> Management of cryptographic keys is incredibly sensitive and is **only recommended for enterprises with a team and infrastructure** that can securely support deploying and managing a key server.

Pour utiliser Key Connector, vous devez :

- Avoir une organisation d'entreprise .
- Avoir un serveur Bitwarden auto-hébergé .
- Avoir une implémentation active du SSO .
- Activez les stratégies Organisation unique et Exiger une authentification unique .

Si votre organisation répond ou peut répondre à ces exigences, y compris une équipe et une infrastructure qui peuvent gérer un serveur clé, contactez-nous et nous activerons Key Connector.

Configurer & déployer Key Connector

Une fois que vous nous aurez contacté concernant Key Connector , nous vous contacterons pour lancer une discussion sur Key Connector. Les étapes qui suivent dans cet article doivent être réalisées en collaboration avec les spécialistes de la réussite client et de la mise en œuvre de Bitwarden.

Obtenir un nouveau fichier de licence

Une fois que vous nous avez contacté concernant Key Connector, un membre de l'équipe de réussite client & mise en œuvre générera un fichier de licence activé par Key Connector pour votre organisation. Lorsque votre collaborateur Bitwarden vous indique qu'il est prêt, suivez les étapes suivantes pour obtenir la nouvelle licence :

1. Ouvrez l'application web cloud Bitwarden et naviguez vers l'écran **Facturation** → **Abonnement** de votre organisation dans la console Admin.
2. Faites défiler vers le bas et sélectionnez le bouton **Télécharger la Licence**.
3. Lorsqu'on vous le demande, entrez l'ID d'installation qui a été utilisé pour installer votre serveur auto-hébergé et sélectionnez **Soumettre**. Si vous ne connaissez pas votre ID d'installation par cœur, vous pouvez le récupérer à partir de `./bwdata/env/global.override.env`.

Vous n'aurez pas besoin de votre fichier de licence immédiatement, mais vous devrez le téléverser sur votre serveur auto-hébergé dans une étape ultérieure.

Initialisez Key Connector

Pour préparer votre serveur Bitwarden pour Key Connector :

1. Enregistrer une sauvegarde de, au minimum, `.bwdata/mssql`. Une fois que Key Connector est en utilisation, il est recommandé que vous ayez accès à une image de sauvegarde pré-Key Connector en cas de problème.

   > **Note**
   >
   > Si vous utilisez une base de données MSSQL externe, faites une sauvegarde de votre base de données de la manière qui convient à votre mise en œuvre.

2. Mettez à jour votre installation de Bitwarden auto-hébergée afin de récupérer les dernières modifications :

   ```
   Bash

   ./bitwarden.sh update
   ```

3. Éditez le fichier `.bwdata/config.yml` et activez Key Connector en basculant `enable_key_connector` à `true`.

   ```
   Bash

   nano bwdata/config.yml
   ```

4. Reconstruisez votre installation de Bitwarden auto-hébergée :

   ```
   Bash

   ./bitwarden.sh rebuild
   ```

5. Mettez à jour votre installation de Bitwarden auto-hébergée à nouveau afin d'appliquer les modifications :

   ```
   Bash

   ./bitwarden.sh update
   ```

Configurer Key Connector

Pour configurer Key Connector :

1. Éditez le fichier `.bwdata/env/key-connector.override.env` qui aura été téléchargé avec le `./bitwarden.sh mettre à jour`.

   ```
   Bash

   nano bwdata/env/key-connector.override.env
   ```

> ⚠ **Warning**
>
> This file will be pre-populated with default values that will spin up a functional local Key Connector setup, however the **default values are not recommended for production environments**.

2. Dans `key-connector.override.env`, vous devrez spécifier des valeurs pour les éléments suivants :

   - Points d'extrémité : Avec quels points d'extrémité Bitwarden le Key Connector peut communiquer.
   - Base de données: Où Key Connector stockera et récupérera les clés des utilisateurs.
   - Paire de clés RSA: Comment Key Connector accédera à une paire de clés RSA pour protéger les clés des utilisateurs au repos.

Points finaux

La configuration automatique remplira les valeurs de point de terminaison en fonction de votre configuration d'installation, cependant, il est recommandé de confirmer que les valeurs suivantes dans `key-connector.override.env` sont précises pour votre configuration :

```bash
keyConnectorSettings__webVaultUri=https://your.bitwarden.domain.com
keyConnectorSettings__identityServerUri=http://identity:5000
```

Base de données

Key Connector doit accéder à une base de données qui stocke les clés d'utilisateur cryptées pour les membres de votre organisation. Créez une base de données sécurisée pour stocker les clés des utilisateurs cryptées et remplacez les valeurs par défaut de `keyConnectorSettings__database__` dans `key-connector.override.env` par les valeurs indiquées dans la colonne **Valeurs Requises** pour la base de données choisie :

> ⚠ **Warning**
>
> Migration from one database to another is **not supported** at this time. Regardless of which provider you choose, **implement a frequent automated backup schedule** for the database.

Base de données	Valeurs requises

JSON local (**par défaut**)	**Non recommandé en dehors des tests.**
	`keyConnectorSettings__database__provider=json`
	`keyConnectorSettings__database__jsonFilePath={File_Path}`

Base de données	Valeurs requises

Microsoft SQL Server	`keyConnectorSettings__database__provider=sqlserver`
	`keyConnectorSettings__database__sqlServerConnectionString={Connection_String}`
	Apprenez à formater les chaînes de connexion MSSQL

PostgreSQL	`keyConnectorSettings__database__provider=postgresql`
	`keyConnectorSettings__database__postgreSqlConnectionString={Connection_String}`
	Apprenez à formater les chaînes de connexion PostgreSQL

MySQL/MariaDB	`keyConnectorSettings__database__provider=mysql`
	`keyConnectorSettings__database__mySqlConnectionString={Connection_String}`
	Apprenez à formater les chaînes de connexion MySQL

MongoDB	`keyConnectorSettings__database__provider=mongo`
	`keyConnectorSettings__database__mongoConnectionString={Connection_String}`
	`keyConnectorSettings__database__mongoDatabaseName={DatabaseName}`
	Apprenez à formater les chaînes de connexion MongoDB

Paire de clés RSA

Key Connector utilise une paire de clés RSA pour protéger les clés utilisateur au repos. Créez une paire de clés et remplacez les valeurs par défaut `keyConnectorSettings__rsaKey__` et `keyConnectorSettings__certificate__` dans `key-connector.override.env` par les valeurs requises pour votre implémentation choisie.

> **Tip**
> The RSA key pair must be **at a minimum** 2048 bits in length.

En général, vos options incluent l'octroi d'un accès Key Connector à un **Certificat** X509 qui contient la paire de clés ou l'octroi d'un accès Key Connector directement à la **Paire de Clés** :

⇒Certificat

Pour utiliser un certificat X509 qui contient une paire de clés RSA, spécifiez les valeurs requises en fonction de l'emplacement où votre certificat est stocké (voir **Système de fichiers**, **Magasin de certificats OS**, et ainsi de suite):

Tip

The certificate **must** be made available as a PKCS12 `.pfx` file, for example:

```
Bash

openssl req -x509 -newkey rsa:4096 -sha256 -nodes -keyout bwkc.key -out bwkc.crt -subj "/CN=Bi
twarden Key Connector" -days 36500

openssl pkcs12 -export -out ./bwkc.pfx -inkey bwkc.key -in bwkc.crt -passout pass:{Password}
```

In all certificate implementations, you'll need the `CN` value shown in this example.

Système de fichiers (par défaut)

Si le certificat est stocké sur le système de fichiers de la machine exécutant Key Connector, spécifiez les valeurs suivantes :

Note

By default, Key Connector will be configured to create a `.pfx` file located at `etc/bitwarden/key-connector/bwkc.pfx` with a generated password. **It is not recommended** for enterprise implementations to use these defaults.

```
Bash

keyConnectorSettings__rsaKey__provider=certificate
keyConnectorSettings__certificate__provider=filesystem
keyConnectorSettings__certificate__filesystemPath={Certificate_Path}
keyConnectorSettings__certificate__filesystemPassword={Certificate_Password}
```

Stockage Blob Azure

Si le certificat est téléversé vers Azure Blob Storage, spécifiez les valeurs suivantes :

```
Bash

keyConnectorSettings__rsaKey__provider=certificate
keyConnectorSettings__certificate__provider=azurestorage
keyConnectorSettings__certificate__azureStorageConnectionString={Connection_String}
keyConnectorSettings__certificate__azureStorageContainer={Container_Name}
keyConnectorSettings__certificate__azureStorageFileName={File_Name}
keyConnectorSettings__certificate__azureStorageFilePassword={File_Password}
```

Définissez `azureStorageConnectionString` sur une **chaîne de connexion** que vous pouvez générer dans votre portail Azure à partir de la page **Signature d'accès partagé** (SAS) de votre compte de stockage. Le SAS doit avoir :

- Services autorisés : Blob et Fichier
- Types de ressources autorisés : Service, Conteneur et Objet
- Autorisations autorisées : Lire, Écrire et Lister
- Autorisations d'index de blob autorisées : Lire/Écrire et Filtrer

Clé Azure Coffre

Si le certificat est stocké dans le coffre de clés Azure, spécifiez les valeurs suivantes :

> **Note**
>
> To use Azure Key Vault to store your `.pfx` certificate, you'll need to create an Active Directory **App Registration**. This App Registration must:
>
> - Give delegated API permissions to access Azure Key Vault
> - Have a client secret generated to allow access by Key Connector

```bash
keyConnectorSettings__certificate__provider=azurekv
keyConnectorSettings__certificate__azureKeyvaultUri={Vault_URI}
keyConnectorSettings__certificate__azureKeyvaultCertificateName={Certificate_Name}
keyConnectorSettings__certificate__azureKeyvaultAdTenantId={ActiveDirectory_TenantId}
keyConnectorSettings__certificate__azureKeyvaultAdAppId={AppRegistration_ApplicationId}
keyConnectorSettings__certificate__azureKeyvaultAdSecret={AppRegistration_ClientSecretValue}
```

Hashicorp Coffre

Si le certificat est stocké dans le coffre Hashicorp, spécifiez les valeurs suivantes :

> **Note**
>
> Key Connector integrates with the Hashicorp Vault KV2 Storage Engine. As per the top of this tab, the certificate file should be in PKCS12 format and stored base64-encoded as the value to a named key in your Vault. If following a Vault tutorial for the KV2 Storage Engine, the key name may be `file` unless otherwise specified.

```Bash
keyConnectorSettings__rsaKey__provider=certificate
keyConnectorSettings__certificate__provider=vault
keyConnectorSettings__certificate__vaultServerUri={Server_URI}
keyConnectorSettings__certificate__vaultToken={Token}
keyConnectorSettings__certificate__vaultSecretMountPoint={Secret_MountPoint}
keyConnectorSettings__certificate__vaultSecretPath={Secret_Path}
keyConnectorSettings__certificate__vaultSecretDataKey={Secret_DataKey}
keyConnectorSettings__certificate__vaultSecretFilePassword={Secret_FilePassword}
```

⇒Paire de clés

Pour utiliser un fournisseur de cloud ou un appareil physique pour stocker une paire de clés RSA 2048, spécifiez les valeurs requises en fonction de votre mise en œuvre choisie (voir **Azure Key Vault**, **Google Cloud Key Management**, et ainsi de suite):

Clé Azure Coffre

Si vous utilisez Azure Key Vault pour stocker une paire de clés RSA 2048, spécifiez les valeurs suivantes :

> **Note**
>
> To use Azure Key Vault to store your RSA 2048 key, you'll need to create an Active Directory **App Registration**. This App Registration must:
>
> - Give delegated API permissions to access Azure Key Vault
> - Have a client secret generated to allow access by Key Connector

```Bash
keyConnectorSettings__rsaKey__provider=azurekv
keyConnectorSettings__rsaKey__azureKeyvaultUri={Vault_URI}
keyConnectorSettings__rsaKey__azureKeyvaultKeyName={Key_Name}
keyConnectorSettings__rsaKey__azureKeyvaultAdTenantId={ActiveDirectory_TenantId}
keyConnectorSettings__rsaKey__azureKeyvaultAdAppId={AppRegistration_ApplicationId}
keyConnectorSettings__rsaKey__azureKeyvaultAdSecret={AppRegistration_ClientSecretValue}
```

Apprenez à utiliser Azure Key Vault pour créer une paire de clés

Gestion des clés Google Cloud

Si vous utilisez Google Cloud Key Management pour stocker une paire de clés RSA 2048, spécifiez les valeurs suivantes :

```Bash
keyConnectorSettings__rsaKey__provider=gcpkms
keyConnectorSettings__rsaKey__googleCloudProjectId={Project_Id}
keyConnectorSettings__rsaKey__googleCloudLocationId={Location_Id}
keyConnectorSettings__rsaKey__googleCloudKeyringId={Keyring_Id}
keyConnectorSettings__rsaKey__googleCloudKeyId={Key_Id}
keyConnectorSettings__rsaKey__googleCloudKeyVersionId={KeyVersionId}
```

Apprenez à utiliser le service de gestion de clés Cloud de Google pour créer des anneaux de clés et des clés asymétriques

Service de Gestion de Clés AWS

Si vous utilisez le Service de Gestion de Clés AWS (KMS) pour stocker une paire de clés RSA 2048, spécifiez les valeurs suivantes :

```Bash
keyConnectorSettings__rsaKey__provider=awskms
keyConnectorSettings__rsaKey__awsAccessKeyId={AccessKey_Id}
keyConnectorSettings__rsaKey__awsAccessKeySecret={AccessKey_Secret}
keyConnectorSettings__rsaKey__awsRegion={Region_Name}
keyConnectorSettings__rsaKey__awsKeyId={Key_Id}
```

Apprenez à utiliser AWS KMS pour créer des clés asymétriques

PKCS11 HSM Physique

Si vous utilisez un appareil HSM physique avec le fournisseur PKCS11, spécifiez les valeurs suivantes :

```Bash
keyConnectorSettings__rsaKey__provider=pkcs11
keyConnectorSettings__rsaKey__pkcs11Provider={Provider}
keyConnectorSettings__rsaKey__pkcs11SlotTokenSerialNumber={Token_SerialNumber}
keyConnectorSettings__rsaKey__pkcs11LoginUserType={Login_UserType}
keyConnectorSettings__rsaKey__pkcs11LoginPin={Login_PIN}

ONE OF THE FOLLOWING TWO:
keyConnectorSettings__rsaKey__pkcs11PrivateKeyLabel={PrivateKeyLabel}
keyConnectorSettings__rsaKey__pkcs11PrivateKeyId={PrivateKeyId}

OPTIONALLY:
keyConnectorSettings__rsaKey__pkcsLibraryPath={path/to/library/file}
```

Où :

- `{Provider}` peut être `yubihsm` ou `opensc`
- `{Login_UserType}` peut être `utilisateur`, `donc`, ou `spécifique_au_contexte`

> **Note**
> If you are using the PKCS11 provider to store your private key on an HSM device, the associated public key must be made available and configured as a certificate using any of the options found in the **Certificates** tab.

Activez Key Connector

Maintenant que Key Connector est entièrement configuré et que vous avez une licence activée par Key Connector, suivez les étapes suivantes :

1. Redémarrez votre installation de Bitwarden auto-hébergée afin d'appliquer les modifications de configuration :

   ```Bash
   ./bitwarden.sh restart
   ```

2. Connectez-vous à votre Bitwarden auto-hébergé en tant que **propriétaire** de l'organisation et accédez à l'écran **Facturation** → **Abonnement** de la console d'administration.

3. Sélectionnez le bouton **Mettre à jour la licence** et téléversez la licence activée par Key Connector récupérée lors d'une étape précédente.

4. Si vous ne l'avez pas déjà fait, naviguez vers l'écran **Paramètres** → **Politiques de sécurité** et activez les politiques Organisation unique et Exiger une authentification unique. **Les deux sont nécessaires pour utiliser Key Connector**.

5. Naviguez vers l'écran **Paramètres** → **Authentification unique**.

> **Tip**
>
> The next few steps assume that you already have an active login with SSO implementation using SAML 2.0 or OIDC. **If you don't**, please implement and test login with SSO before proceeding.

6. Dans la section **Options de déchiffrement des membres**, sélectionnez **Key Connector**.

7. Dans l'entrée **URL du Key Connector**, entrez l'adresse où le Key Connector est en cours d'exécution (par défaut, `https://votre.domaine/key-connector`) et sélectionnez le bouton **Test** pour vous assurer que vous pouvez atteindre le Key Connector.

8. Faites défiler jusqu'en bas de l'écran et sélectionnez **Enregistrer**.