SELF-HOST > KEY CONNECTOR

Key Connector einsetzen

Key Connector einsetzen

Dieser Artikel führt Sie durch das Verfahren zur Aktivierung und Konfiguration des Key Connectors in einer bestehenden selbst gehosteten Umgebung. **Bevor Sie fortfahren** , lesen Sie bitte den Artikel über Key Connector sorgfältig durch, um sicherzustellen, dass Sie vollständig verstehen, was Key Connector ist, wie er funktioniert und welche Auswirkungen die Implementierung hat.

Bitwarden unterstützt die Bereitstellung eines Key Connectors zur Nutzung durch eine Organisation für eine selbst gehostete Instanz.

Anforderungen

> **⚠ 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.

Um Key Connector zu verwenden, müssen Sie:

- Haben Sie eine Unternehmensorganisation .
- Verfügen Sie über einen selbst gehosteten Bitwarden-Server .
- Verfügen Sie über eine aktive SSO-Implementierung .
- Aktivieren Sie die Richtlinien „Einzelne Organisation“ und „Einmaliges Anmelden erforderlich“ .

Wenn Ihre Organisation diese Anforderungen erfüllt oder erfüllen kann, einschließlich eines Teams und einer Infrastruktur, die die Verwaltung eines Schlüsselservers unterstützen kann, kontaktieren Sie uns und wir werden den Key Connector aktivieren.

Einrichten & Bereitstellen von Key Connector

Sobald Sie uns bezüglich Key Connector kontaktiert haben , werden wir uns mit Ihnen in Verbindung setzen, um eine Key Connector-Diskussion zu starten. Die folgenden Schritte in diesem Artikel müssen in Zusammenarbeit mit Bitwarden Kundenerfolgs- und Implementierungsspezialisten abgeschlossen werden.

Erhalten Sie eine neue Lizenzdatei

Sobald Sie uns bezüglich des Key Connectors kontaktiert haben, wird ein Mitglied des Kundenerfolgs- und Implementierungsteams eine Key Connector-aktivierte Lizenzdatei für Ihre Organisation generieren. Wenn Ihr Bitwarden-Mitarbeiter Ihnen mitteilt, dass er bereit ist, führen Sie die folgenden Schritte aus, um die neue Lizenz zu erhalten:

1. Öffnen Sie die Bitwarden Cloud-Web-App und navigieren Sie zu dem **Rechnung → Abonnement** Bildschirm Ihrer Organisation im Administrator-Konsole.
2. Scrollen Sie nach unten und wählen Sie die Schaltfläche **Lizenz herunterladen**.

3. Wenn Sie dazu aufgefordert werden, geben Sie die Installations-ID ein, die verwendet wurde, um Ihren selbst gehosteten Server zu installieren, und wählen Sie **Absenden**. Wenn Sie Ihre Installations-ID nicht auswendig kennen, können Sie diese von `./bwdata/env/global.override.env` abrufen.

Sie benötigen Ihre Lizenzdatei nicht sofort, aber Sie müssen sie auf Ihren selbst gehosteten Server in einem späteren Schritt hochladen.

Initialisiere Key Connector

Um Ihren Bitwarden-Server für den Key Connector vorzubereiten:

1. Speichern Sie mindestens eine Sicherungskopie von `.bwdata/mssql`. Sobald der Key Connector verwendet wird, wird empfohlen, dass Sie Zugang zu einem Backup-Image vor der Verwendung des Key Connectors haben, falls ein Problem auftritt.

> **ⓘ Note**
>
> Wenn Sie eine externe MSSQL-Datenbank verwenden, erstellen Sie eine Sicherungskopie Ihrer Datenbank auf die Weise, die am besten zu Ihrer Implementierung passt.

2. Aktualisieren Sie Ihre selbst gehostete Bitwarden-Installation, um die neuesten Änderungen abzurufen:

```
Bash

./bitwarden.sh update
```

3. Bearbeiten Sie die `.bwdata/config.yml` Datei und aktivieren Sie Key Connector, indem Sie `enable_key_connector` auf `true` umschalten.

```
Bash

nano bwdata/config.yml
```

4. Bauen Sie Ihre selbst gehostete Bitwarden-Installation neu auf:

```
Bash

./bitwarden.sh rebuild
```

5. Aktualisieren Sie Ihre selbst gehostete Bitwarden-Installation erneut, um die Änderungen anzuwenden:

```
Bash

./bitwarden.sh update
```

Konfigurieren Sie den Key Connector

Um den Key Connector zu konfigurieren:

1. Bearbeiten Sie die `.bwdata/env/key-connector.override.env` Datei, die mit der `./bitwarden.sh Aktualisierung` heruntergeladen worden sein wird.

```
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. In `key-connector.override.env`, müssen Sie Werte für Folgendes angeben:
   - Endpunkte: Mit welchen Bitwarden Endpunkten der Key Connector kommunizieren kann.
   - Datenbank: Wo Key Connector Benutzerschlüssel speichern und abrufen wird.
   - RSA-Schlüsselpaar: Wie Key Connector auf ein RSA-Schlüsselpaar zugreifen wird, um Benutzerschlüssel im Ruhezustand zu schützen.

Endpunkte

Die automatisierte Einrichtung wird Endpunkt-Werte basierend auf Ihrer Installationskonfiguration füllen, jedoch wird empfohlen, dass Sie die folgenden Werte in `key-connector.override.env` bestätigen, die für Ihre Einrichtung genau sind:

```
Bash

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

Datenbank

Der Key Connector muss auf eine Datenbank zugreifen, die verschlüsselte Benutzerschlüssel für die Mitglieder Ihrer Organisation speichert. Erstellen Sie eine sichere Datenbank zur Speicherung verschlüsselter Benutzerschlüssel und ersetzen Sie die Standardwerte von

`keyConnectorSettings__database__` in `key-connector.override.env` durch die Werte, die in der Spalte **Erforderliche Werte** für die ausgewählte Datenbank angegeben sind:

> **⚠ 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.

Datenbank	Erforderliche Werte

Lokales JSON (**Standard**)	**Nicht empfohlen außerhalb von Tests.**
	`keyConnectorSettings__database__provider=json`
	`keyConnectorSettings__database__jsonFilePath={File_Path}`

Microsoft SQL Server	`keyConnectorSettings__database__provider=sqlserver`
	`keyConnectorSettings__database__sqlServerConnectionString={Connection_String}`
	Lernen Sie, wie Sie MSSQL-Verbindungszeichenfolgen formatieren

PostgreSQL	`keyConnectorSettings__database__provider=postgresql`
	`keyConnectorSettings__database__postgreSqlConnectionString={Connection_String}`
	Lernen Sie, wie Sie Verbindungszeichenfolgen für PostgreSQL formatieren

MySQL/MariaDB	`keyConnectorSettings__database__provider=mysql`
	`keyConnectorSettings__database__mySqlConnectionString={Connection_String}`
	Lernen Sie, wie Sie MySQL-Verbindungsstrings formatieren

MongoDB	`keyConnectorSettings__database__provider=mongo` `keyConnectorSettings__database__mongoConnectionString={Connection_String}`

Datenbank	Erforderliche Werte
	`keyConnectorSettings__database__mongoDatabaseName={DatabaseName}` Lernen Sie, wie Sie Verbindungsstrings in MongoDB formatieren

RSA-Schlüsselpaar

Key Connector verwendet ein RSA-Schlüsselpaar, um Benutzerschlüssel im Ruhezustand zu schützen. Erstellen Sie ein Schlüsselpaar und ersetzen Sie die Standardwerte `keyConnectorSettings__rsaKey__` und `keyConnectorSettings__certificate__` in `key-connector.override.env` durch die für Ihre gewählte Implementierung erforderlichen Werte.

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

Im Allgemeinen beinhalten Ihre Optionen, dem Key Connector Zugriff auf ein X509 **Zertifikat** zu gewähren, das das Schlüsselpaar enthält, oder dem Key Connector direkt Zugriff auf das **Schlüsselpaar** zu gewähren:

⇒Zertifikat

Um ein X509-Zertifikat zu verwenden, das ein RSA-Schlüsselpaar enthält, geben Sie die erforderlichen Werte an, abhängig davon, wo Ihr Zertifikat gespeichert ist (siehe **Dateisystem**, **OS-Zertifikatspeicher** und so weiter):

> **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=
> Bitwarden 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.

Dateisystem (Standard)

Wenn das Zertifikat auf dem Dateisystem der Maschine gespeichert ist, auf der Key Connector läuft, geben Sie die folgenden Werte an:

> **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}
```

Azure Blob Speicher

Wenn das Zertifikat auf Azure Blob Storage hochgeladen wird, geben Sie die folgenden Werte an:

```
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}
```

Setzen Sie `azureStorageConnectionString` auf eine **Verbindungszeichenfolge**, die Sie in Ihrem Azure-Portal auf der **Shared Access Signature** (SAS) Seite Ihres Speicherkontos generieren können. Die SAS muss haben:

- Erlaubte Dienste: Blob und Datei
- Erlaubte Ressourcen-Typen: Service, Container und Objekt
- Erlaubte Berechtigungen: Lesen, Schreiben und Auflisten
- Erlaubte Blob-Index-Berechtigungen: Lesen/Schreiben und Filtern

Azure Schlüsseltresor

Wenn das Zertifikat im Azure Key Vault gespeichert ist, geben Sie die folgenden Werte an:

> **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 Tresor

Wenn das Zertifikat im Hashicorp Tresor gespeichert ist, geben Sie die folgenden Werte an:

> **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}
```

⇒Schlüsselpaar

Um einen Cloud-Anbieter oder ein physisches Gerät zur Speicherung eines RSA 2048 Schlüsselpaars zu verwenden, geben Sie die erforderlichen Werte an, abhängig von Ihrer gewählten Implementierung (siehe **Azure Key Tresor**, **Google Cloud Schlüsselverwaltung** und so weiter):

Azure Schlüsseltresor

Wenn Sie Azure Key Vault verwenden, um ein RSA 2048 Schlüsselpaar zu speichern, geben Sie die folgenden Werte an:

> **ⓘ 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}
```

Lernen Sie, wie Sie Azure Key Vault verwenden, um ein Schlüsselpaar zu erstellen.

Google Cloud Schlüsselverwaltung

Wenn Sie Google Cloud Key Management verwenden, um ein RSA 2048 Schlüsselpaar zu speichern, geben Sie die folgenden Werte an:

```
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}
```

Lernen Sie, wie Sie den Google Cloud Key Management Service verwenden, um Schlüsselringe und asymmetrische Schlüssel zu erstellen.

AWS Schlüsselverwaltungsdienst

Wenn Sie den AWS Key Management Service (KMS) verwenden, um ein RSA 2048 Schlüsselpaar zu speichern, geben Sie die folgenden Werte an:

```
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}
```

Lernen Sie, wie Sie AWS KMS zur Erstellung asymmetrischer Schlüssel verwenden können.

PKCS11 Physischer HSM

Wenn Sie ein physisches HSM-Gerät mit dem PKCS11-Anbieter verwenden, geben Sie die folgenden Werte an:

```
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__pkcs11LibraryPath={path/to/library/file}
```

Wo:

- `{Provider}` kann `yubihsm` oder `opensc` sein
- `{Login_UserType}` kann `Benutzer`, `so`, oder `kontextspezifisch` sein

> **ⓘ 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.

Aktivieren Sie den Key Connector

Jetzt, da Key Connector vollständig konfiguriert ist und Sie eine Key Connector-fähige Lizenz haben, führen Sie die folgenden Schritte aus:

1. Starten Sie Ihre selbst gehostete Bitwarden-Installation neu, um die Konfigurationsänderungen anzuwenden:

   ```
   Bash

   ./bitwarden.sh restart
   ```

2. Melden Sie sich als **Organisationsinhaber** bei Ihrem selbst gehosteten Bitwarden an und navigieren Sie zum Bildschirm **„Abrechnung“** → **„Abonnement“** der Admin-Konsole.

3. Wählen Sie die Schaltfläche **Lizenz aktualisieren** und laden Sie die Key Connector-aktivierte Lizenz hoch, die in einem früheren Schritt

abgerufen wurde.

4. Wenn Sie es noch nicht getan haben, navigieren Sie zu dem **Einstellungen → Richtlinien** Bildschirm und aktivieren Sie die Einzelne Organisation und Einfache Anmeldungsauthentifizierung erforderlich Richtlinien. **Beide sind erforderlich, um Key Connector zu verwenden** .

5. Navigieren Sie zu dem **Einstellungen → Einmaliges Anmelden** Bildschirm.

> **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. Im Abschnitt **Entschlüsselungsoptionen für Mitglieder**, wählen Sie **Key Connector**.

7. Geben Sie in der Eingabe für die **Key Connector URL** die Adresse ein, unter der Key Connector läuft (standardmäßig `https://your.domain/key-connector`) und wählen Sie die **Test** Schaltfläche aus, um sicherzustellen, dass Sie Key Connector erreichen können.

8. Scrollen Sie zum unteren Rand des Bildschirms und wählen Sie **Speichern**.