bitwarden | ヘルプセンターの記事

シークレットマネージャー > 始めましょう

開発者クイックスタート

ヘルプセンターで表示:
https://bitwarden.com/help/developer-quick-start/

開発者クイックスタート

Bitwardenシークレットマネージャーは、開発者、DevOps、およびサイバーセキュリティチームが、シークレットを中央で保存、管理、および大規模に展開することを可能にします。シークレットマネージャーCLIは、
認証されたサービスアカウントを通じてアプリケーションやインフラストラクチャにシークレットを注入するための主要な手段です。

この記事では、シークレットマネージャーCLIの使用方法を示します。具体的には、保管庫に保存されたデータベースの資格情報を取得し、それをBitwarden Unified Dockerイメージのコンテナランタイムで注入するいくつかの方法を見ていきます。

> **Tip**
> シークレットマネージャーの機能に関するSDK情報と言語ラッパーをお探しの場合は、この記事を参照してください。

まだシークレットマネージャークイックスタートの記事を読んでいない場合は、先にそれを読むことをお勧めします。

基本的なチュートリアル

この最もシンプルな例では、保管庫に保存されたデータベースの資格情報を取得し、
それらをLinuxシステムの一時的な環境変数として保存します。保存したら、`docker run`コマンド内でランタイムにそれらを注入します。
これを行うには、以下をインストールする必要があります：

- Bitwarden シークレットマネージャー CLI
- ドッカー
- jqのようなコマンドラインJSONプロセッサ

認証する

シークレットマネージャーのCLIは、特定のサービスアカウントに対して生成されたアクセストークンを使用してログインすることができます。これは、**サービスアカウントがアクセスできるシークレットとプロジェクトのみが**CLIを使用して操作できることを意味します（サービスアカウントの権限について詳しく学びましょう）。CLIセッションを認証する方法は数値ありますが、
最も簡単なオプションは環境変数`BWS_ACCESS_TOKEN`をアクセストークンの値で保存することです。例えば：

```bash
export BWS_ACCESS_TOKEN=0.48c78342-1635-48a6-accd-afbe01336365.C0tMmQqHnAp1h0gL8bngprlPOYutt0:B3h5D+YgLvFiQhWkIq6Bow==
```

秘密を取り戻せ

次に、以下のコマンドを使用してデータベースのユーザー名を取得し、一時的な環境変数として保存します。この例では、`fc3a93f4-2a16-445b-b0c4-aeaf0102f0ff`はデータベースのユーザー名の秘密の特定の識別子を表しています。

```bash
export SECRET_1=$(bws secret get fc3a93f4-2a16-445b-b0c4-aeaf0102f0ff | jq '.value')
```

このコマンドはあなたの秘密の値を一時的な環境変数に保存します。これはシステムの再起動、ユーザーのログアウト、または新しいシェルでクリアされます。さて、データベースのパスワードに対して同じコマンドを実行してください。

```Bash
export SECRET_2=$(bws secret get 80b55c29-5cc8-42eb-a898-acfd01232bbb | jq '.value')
```

秘密を注入してください

あなたのデータベースの資格情報が一時的な環境変数として保存されているので、それらは`docker run`コマンドの中に注入することができます。この例では、注入された秘密を強調するために、Bitwarden Unifiedに必要な多くの変数を省略しました。

```Bash
docker run -d --name bitwarden .... -env BW_DB_USERNAME=$SECRET_1 BW_BD_PASSWORD=$SECRET_2 .... bitwarden/self-host:beta
```

このコマンドが実行されると、Dockerコンテナが起動し、一時的に保存された環境変数からデータベースの資格情報を注入します。
これにより、プレーンテキストとしての機密情報を渡すことなく、安全にBitwarden Unifiedを起動することができます。

高度なチュートリアル

この例では、Dockerイメージ内のシークレットマネージャーCLIを使用して、
ランタイムで保管庫に保存されたデータベースの資格情報を注入します。これは、
Dockerfileを操作してCLIをホストではなくイメージにインストールし、
コンテナのランタイムでデータベースの資格情報を取得することで達成します。次に、環境変数ファイルを注入のために準備し、
コンテナを実行することで全てを一緒にします。

あなたのDockerfileを設定してください。

あなたのDockerイメージにシークレットマネージャーCLIをインストールするには、以下をあなたのDockerfileに追加する必要があります:

```Bash
RUN curl -O https://github.com/bitwarden/sdk/releases/download/bws-v1.0.0/bws-x86_64-unknown-linux-gnu-1.0.0.zip && unzip bws-x86_64-unknown-linux-gnu-1.0.0.zip && export PATH=/this/directory:$PATH
```

次に、各資格情報を取得して注入可能にするために、`RUN`ステートメントを構築する必要があります。
これらのステートメントにはインライン認証が含まれますが、これが実装できる唯一の方法ではありません：

```Bash
RUN SECRET_1=$(bws secret get fc3a93f4-2a16-445b-b0c4-aeaf0102f0ff --access-token $BWS_ACCESS_TOKEN | jq '.value')
```

```
Bash

RUN SECRET_2=$(bws secret get 80b55c29-5cc8-42eb-a898-acfd01232bbb --access-token $BWS_ACCESS_TOKEN
| jq '.value')
```

これらのRUNステートメントは、Dockerfileに指定されたシークレットを取得するように促します。ここで、fc3a93f4-2a16-445b-b0c4-aeaf0102f0ffはシークレットの特定の識別子を表しています。

あなたのenvファイルを準備してください

これからあなたのデータベースの資格情報がインジェクションのために利用可能になるので、
これらの値を受け取ることができるようにあなたのsettings.envファイルを調整してください。これを行うには、
ファイル内の関連するハードコードされた値を指定された変数名（この場合、SECRET_1とSECRET_2）に置き換えてください：

```
Bash

# Database
# Available providers are sqlserver, postgresql, mysql/mariadb, or sqlite
BW_DB_PROVIDER=mysql
BW_DB_SERVER=db
BW_DB_DATABASE=bitwarden_vault
BW_DB_USERNAME=$SECRET_1
BW_DB_PASSWORD=$SECRET_2
```

コンテナを実行します

あなたのデータベースの資格情報が準備完了し、注入の準備ができたので、コンテナを起動し、環境変数として使用するアクセストークンをbws ログインで指定してください。

```
Bash

docker run --rm -it -e BWS_ACCESS_TOKEN=<your-access-token> image-name
```

このコマンドが実行されると、Dockerコンテナが起動し、コンテナから取得した値からデータベースの資格情報を注入します。これにより、平文としての機密値を渡すことなく、安全にBitwarden Unifiedを起動することができます。