bitwarden | Hilfezentrum Artikel

SECRETS MANAGER > LOS GEHT'S

Entwickler Schnellstart

Ansicht im Hilfezentrum:
https://bitwarden.com/help/developer-quick-start/

Entwickler Schnellstart

Bitwarden Secrets Manager ermöglicht Entwicklern, DevOps und Cybersicherheitsteams, Geheimnisse zentral zu speichern, zu verwalten und im großen Maßstab bereitzustellen. Die Secrets Manager CLI ist Ihr Hauptinstrument, um Geheimnisse in Ihre Anwendungen und Infrastruktur über ein authentifiziertes Service-Konto einzuspeisen.

In diesem Artikel demonstrieren wir die Verwendung der Secrets Manager-CLI, indem wir uns einige Möglichkeiten ansehen, um in Ihrem Tresor gespeicherte Datenbankanmeldeinformationen abzurufen, um sie zur Containerlaufzeit für ein Bitwarden Unified Docker-Image einzufügen.

> **Tip**
>
> Wenn Sie nach SDK-Informationen und Sprachwrappern für die Funktionalität des Secrets Manager suchen, verweisen Sie auf diesen Artikel.

Wenn Sie den Secrets Manager Quick Start Artikel noch nicht durchgegangen sind, empfehlen wir, dies zu tun, bevor Sie weiterlesen.

Grundanleitung

In diesem einfachsten Beispiel holen Sie Datenbank-Anmeldeinformationen aus Ihrem Tresor und speichern sie als temporäre Umgebungsvariablen auf einem Linux-System. Einmal gespeichert, injizieren Sie sie zur Laufzeit in einem `docker run` Befehl. Um dies zu tun, müssen Sie folgendes installiert haben:

- Bitwarden Secrets Manager CLI
- Docker
- Ein Befehlszeilen-JSON-Prozessor wie jq

Authentifizieren

Der Secrets Manager CLI kann mit einem für ein bestimmtes Service-Konto generierten Zugriffs-Token angemeldet werden. Das bedeutet, dass **nur Geheimnisse und Projekte, auf die das Dienstkonto Zugriff hat**, mit der CLI interagiert werden dürfen (erfahren Sie mehr über Berechtigungen für Dienstkonten). Es gibt eine Reihe von Möglichkeiten, eine CLI-Sitzung zu authentifizieren, aber für die einfachste Option tun Sie dies, indem Sie eine Umgebungsvariable `BWS_ACCESS_TOKEN` mit dem Wert Ihres Zugriffstokens speichern, zum Beispiel:

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

Holen Sie das Geheimnis

Verwenden Sie als Nächstes den folgenden Befehl, um Ihren Datenbank-Benutzernamen abzurufen und ihn als temporäre Umgebungsvariable zu speichern. In diesem Beispiel stellt `fc3a93f4-2a16-445b-b0c4-aeaf0102f0ff` den spezifischen Identifikator für das Geheimnis des Datenbank-Benutzernamens dar:

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

Dieser Befehl wird den `Wert` Ihres Geheimnisses in einer temporären Umgebungsvariable speichern, die bei einem Systemneustart, Benutzerabmeldung oder in einer neuen Shell gelöscht wird. Führen Sie jetzt denselben Befehl für das Datenbank-Passwort aus:

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

Injiziere das Geheimnis

Jetzt, da Ihre Datenbank-Anmeldeinformationen als temporäre Umgebungsvariablen gespeichert sind, können sie in einen `docker run` Befehl eingefügt werden. In diesem Beispiel haben wir viele der von Bitwarden Unified benötigten Variablen weggelassen, um die injizierten Geheimnisse zu betonen:

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

Wenn dieser Befehl ausgeführt wird, startet Ihr Docker-Container und injiziert Ihre Datenbank-Anmeldeinformationen aus den temporär gespeicherten Umgebungsvariablen, was Ihnen ermöglicht, Bitwarden Unified sicher zu starten, ohne sensible Werte als Klartext zu übergeben.

Fortgeschrittenes Tutorial

In diesem Beispiel verwenden Sie die Secrets Manager CLI in Ihrem Docker-Image, um Datenbankanmeldedaten, die in Ihrem Tresor gespeichert sind, zur Laufzeit zu injizieren. Sie erreichen dies, indem Sie Ihre Dockerfile manipulieren, um die CLI auf dem Image zu installieren, anstatt auf dem Host, und um die Datenbank-Anmeldeinformationen zur Laufzeit des Containers abzurufen. Sie werden dann Ihre Umgebungsvariablen-Datei zur Injektion vorbereiten und alles zusammenfügen, indem Sie einen Container ausführen.

Richten Sie Ihre Dockerfile ein

Um die Secrets Manager CLI in Ihrem Docker-Image zu installieren, müssen Sie Folgendes zu Ihrer Dockerfile hinzufügen:

```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
```

Als nächstes müssen Sie `RUN`-Anweisungen erstellen, um jedes Anmeldeinformation zu abrufen, damit sie zur Injektion verfügbar sind. Diese Aussagen werden eine Inline-Authentifizierung enthalten, jedoch ist dies nicht die einzige Methode, die Sie implementieren könnten:

```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')
```

Diese RUN-Anweisungen fordern Ihre Dockerfile auf, die angegebenen Geheimnisse abzurufen, wobei `fc3a93f4-2a16-445b-b0c4-aeaf0102f0ff` den spezifischen Kennzeichner des Geheimnisses darstellt.

Bereiten Sie Ihre env-Datei vor

Nun, da Ihre Datenbank-Anmeldeinformationen für die Einspeisung verfügbar gemacht werden, konditionieren Sie Ihre `settings.env` Datei, um diese Werte empfangen zu können. Um dies zu tun, ersetzen Sie relevante fest codierte Werte in der Datei durch die zugewiesenen Variablennamen (in diesem Fall, `SECRET_1` und `SECRET_2`):

```
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
```

Starten Sie den Container

Jetzt, da Ihre Datenbank-Zugangsdaten bereit und für die Einspeisung vorbereitet sind, starten Sie Ihren Container und geben Sie das zu verwendende Token mit `bws Zugangsdaten` als Umgebungsvariable an:

```
Bash

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

Wenn dieser Befehl ausgeführt wird, startet Ihr Docker-Container und injiziert Ihre Datenbank-Anmeldeinformationen aus den vom Container abgerufenen Werten, was Ihnen ermöglicht, Bitwarden Unified sicher zu starten, ohne sensible Werte als Klartext zu übergeben.