ADMINISTRADOR DE SECRETOS > COMIENZA

Inicio Rápido para Desarrolladores

Inicio Rápido para Desarrolladores

El Administrador de secretos de Bitwarden permite a los desarrolladores, DevOps y Equipos de ciberseguridad almacenar, gestionar y desplegar secretos de manera centralizada y a gran escala. El ILC del Administrador de secretos es su vehículo principal para inyectar secretos en sus aplicaciones e infraestructura a través de una cuenta de servicio autenticada.

En este artículo, demostraremos el uso del ILC del Administrador de secretos al observar algunas formas de recuperar las credenciales de la base de datos almacenadas en su caja fuerte para ser inyectadas en tiempo de ejecución del contenedor para una imagen Docker de Bitwarden Unified.

> **Tip**
>
> Si está buscando información de SDK y envoltorios de lenguaje para la funcionalidad del Administrador de secretos, consulte este artículo.

Si aún no has revisado el artículo de Inicio Rápido del Administrador de Secretos, recomendamos hacerlo antes de continuar leyendo.

Tutorial básico

En este ejemplo más simple, recuperarás las credenciales de la base de datos almacenadas en tu caja fuerte y las almacenarás como variables de entorno temporales en un sistema Linux. Una vez almacenados, los inyectarás en tiempo de ejecución dentro de un comando `docker run`. Para hacer esto, necesitarás haber instalado:

- Bitwarden Administrador de secretos ILC
- Docker
- Un procesador JSON de línea de comandos como jq

Autenticar

El CLI del Administrador de secretos puede iniciar sesión utilizando un token de acceso generado para una cuenta de servicio en particular. Esto significa que **solo los secretos y proyectos a los que la cuenta de servicio tiene acceso** pueden interactuar utilizando la ILC (aprende más sobre los permisos de la cuenta de servicio). Hay un número de formas de autenticar una sesión de ILC, pero para la opción más simple, hazlo guardando una variable de entorno `BWS_ACCESS_TOKEN` con el valor de tu token de acceso, por ejemplo:

```
Bash

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

Recupera el secreto

A continuación, use el siguiente comando para recuperar su nombre de usuario de la base de datos y almacenarlo como una variable de entorno temporal. En este ejemplo, `fc3a93f4-2a16-445b-b0c4-aeaf0102f0ff` representa el identificador específico para el secreto del nombre de usuario de la base de datos:

```
Bash

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

Este comando guardará el `valor` de tu secreto en una variable de entorno temporal, que se borrará al reiniciar el sistema, cerrar sesión del usuario, o en cualquier nueva shell. Ahora, ejecuta el mismo comando para la contraseña de la base de datos:

```
Bash

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

Inyecta el secreto

Ahora que tus credenciales de base de datos están guardadas como variables de entorno temporales, pueden ser inyectadas dentro de un comando `docker run`. En este ejemplo, hemos omitido muchas de las variables requeridas por Bitwarden Unified para enfatizar los secretos inyectados:

```
Bash

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

Cuando se ejecuta este comando, su contenedor Docker se iniciará e inyectará sus credenciales de base de datos desde las variables de entorno almacenadas temporalmente, permitiéndole iniciar de manera segura Bitwarden Unified sin pasar valores sensibles como texto sin formato.

Tutorial avanzado

En este ejemplo, utilizarás el ILC del Administrador de secretos en tu imagen Docker para inyectar las credenciales de la base de datos almacenadas en tu caja fuerte en tiempo de ejecución. Lograrás esto manipulando tu Dockerfile para instalar el ILC en la imagen, en lugar de en el anfitrión, y para recuperar las credenciales de la base de datos en el tiempo de ejecución del contenedor. Luego prepararás tu archivo de variables de entorno para inyección y lo unirás todo ejecutando un contenedor.

Configura tu Dockerfile

Para instalar el ILC del Administrador de secretos en tu imagen de Docker, necesitarás agregar lo siguiente a tu Dockerfile:

```
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
```

A continuación, necesitarás construir declaraciones `RUN` para recuperar cada credencial con el fin de hacerlas disponibles para la inyección. Estas declaraciones incluirán autenticación en línea, sin embargo, este no es el único método que podrías implementar:

```
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')
```

Estas declaraciones `RUN` harán que tu Dockerfile recupere los secretos indicados, donde `fc3a93f4-2a16-445b-b0c4-aeaf0102f0ff` representa el identificador específico del secreto.

Prepara tu archivo env

Ahora que sus credenciales de base de datos estarán disponibles para inyección, condicione su archivo `settings.env` para poder recibir estos valores. Para hacer esto, reemplace los valores codificados relevantes en el archivo con los nombres de las variables designadas (en este caso, `SECRET_1` y `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
```

Ejecuta el contenedor

Ahora que tus credenciales de base de datos están preparadas y listas para la inyección, inicia tu contenedor y especifica el token de acceso a usar con `inicio de sesión bws` como una variable de entorno:

```
Bash

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

Cuando se ejecuta este comando, su contenedor Docker se iniciará e inyectará sus credenciales de base de datos a partir de los valores recuperados por el contenedor, permitiéndole iniciar de manera segura Bitwarden Unified sin pasar valores sensibles como texto sin formato.