bitwarden | Article du Centre d'Aide

SECRETS MANAGER > INTÉGRATIONS

Ansible

Afficher dans le centre d'aide:
https://bitwarden.com/help/ansible-integration/

Ansible

Bitwarden offre une intégration avec Ansible pour récupérer les secrets de Secrets Manager et les injecter dans votre playbook Ansible. Le plugin de recherche injectera les secrets récupérés sous forme de variables d'environnement masquées à l'intérieur d'un playbook Ansible. Pour configurer la collection :

Exigences

- Nous recommandons d'installer les packages Python dans un environnement virtuel Python.
- Version actuelle d'Ansible installée sur votre système.
- Bitwarden Secrets Manager avec un compte de service actif.

Avant de configurer la collection Ansible, nous vous recommandons également d'ouvrir Secrets Manager pour accéder à votre jeton d'accès et à tous les secrets que vous souhaitez inclure dans les paramètres.

Installez la collection Ansible Bitwarden

Le guide suivant est un exemple de configuration pour la collection Bitwarden en utilisant une machine Linux.

1. Installez le SDK Bitwarden :

```Bash
pip install bitwarden-sdk
```

2. Installez la collection bitwarden.secrets:

```Bash
ansible-galaxy collection install bitwarden.secrets
```

Maintenant que la collection Ansible a été installée, nous pouvons commencer à appeler les secrets Bitwarden à partir d'un playbook Ansible avec `bitwarden.secrets.lookup`. La section suivante inclura des exemples pour démontrer ce processus.

> **Note**
>
> Les utilisateurs de macOS peuvent avoir besoin de définir la variable d'environnement suivante dans le shell afin d'éviter les problèmes Ansible en amont.
>
> - `exporter OBJC_DISABLE_INITIALIZE_FORK_SAFETY=YES`

Récupérer les secrets de Bitwarden

Pour récupérer des secrets du Secrets Manager dans votre livre de jeu, il existe deux méthodes :

Enregistrez le jeton d'accès en tant que variable d'environnement.

En utilisant le Secrets Manager, nous pouvons définir en toute sécurité notre jeton d'accès comme une variable d'environnement dans le shell et utiliser le playbook pour récupérer le secret. Pour authentifier le jeton d'accès:

1. Dans le shell, exécutez la commande suivante pour définir votre variable d'environnement de jeton d'accès :

```Bash
export BWS_ACCESS_TOKEN=<ACCESS_TOKEN_VALUE>
```

2. Maintenant que la variable d'environnement a été définie, nous pouvons utiliser le plugin de recherche pour peupler les variables dans notre playbook. Par exemple:

```Bash
  vars:
        database_password: "{{ lookup('bitwarden.secrets.lookup', '<SECRET_ID>') }}"
```

> **Note**
> En définissant `BWS_ACCESS_TOKEN` comme une variable d'environnement, le jeton d'accès peut être référencé sans inclure la valeur brute du jeton d'accès dans le playbook.

Fournissez le jeton d'accès dans le livre de jeu

Le jeton d'accès Secrets Manager peut également être référencé dans le playbook lui-même. Cette méthode ne vous obligerait pas à utiliser la variable d'environnement `BWS_ACCESS_TOKEN` dans votre shell, cependant, la valeur du jeton d'accès sera stockée dans le playbook lui-même.

1. Les jetons d'accès peuvent être inclus dans le playbook avec l'exemple suivant :

```Bash
vars:
    password_with_a_different_access_token: "{{ lookup('bitwarden.secrets.lookup', '<SECRET_ID_V
ALUE>',
access_token='<ACCESS_TOKEN_VALUE>') }}"
```

En utilisant cette méthode, plusieurs jetons d'accès peuvent être référencés dans un seul livre de jeu.

Récupérer le secret d'un serveur différent

Les utilisateurs auto-hébergés de Bitwarden peuvent récupérer des secrets de leur serveur Bitwarden en incluant le `base_url`, `api_url` et `identity_url` :

```
Bash

vars:
    secret_from_other_server: "{{ lookup('bitwarden.secrets.lookup', '<SECRET_ID>', base_url='http
s://bitwarden.example.com' ) }}"
    secret_advanced: >-
      {{ lookup('bitwarden.secrets.lookup', '<SECRET_ID>',
        api_url='https://bitwarden.example.com/api',
        identity_url='https://bitwarden.example.com/identity' ) }}
```

Exemple de playbook

Voici un exemple de fichier de playbook avec plusieurs options de configuration.

```
Bash

---
- name: Using secrets from Bitwarden

  vars:
    bws_access_token: "{{ lookup('env', 'CUSTOM_ACCESS_TOKEN_VAR') }}"
    state_file_dir: "{{ '~/.config/bitwarden-sm' | expanduser }}"
    secret_id: "9165d7a8-2c22-476e-8add-b0d50162c5cc"

    secret: "{{ lookup('bitwarden.secrets.lookup', secret_id) }}"
    secret_with_field: "{{ lookup('bitwarden.secrets.lookup', secret_id, field='note' ) }}"
    secret_with_access_token: "{{ lookup('bitwarden.secrets.lookup', secret_id, access_token=bws_ac
cess_token ) }}"
    secret_with_state_file: "{{ lookup('bitwarden.secrets.lookup', secret_id, state_file_dir=state_
file_dir ) }}"

  tasks:
    - name: Use the secret in a task
      include_tasks: tasks/add_db_user.yml # reference the secrets with "{{ secret }}", "{{ secret_
with_field }}", etc.
```

> **Note**
> Dans l'exemple ci-dessus, le `CUSTOM_ACCESS_TOKEN_VAR` démontre que vous pouvez inclure plusieurs jetons d'accès différents. Ces derniers n'ont pas besoin d'être gravés sur une carte de paiement et peuvent être fournis de manière sécurisée à votre carnet de jeu.

Variable	Informations supplémentaires
`bws_access_token`	Recherchez le jeton d'accès à la variable `env`.
`répertoire_fichier_état`	Un répertoire où votre état d'authentification peut être mis en cache.
`id_secret`	ID du secret que vous souhaitez rechercher.
`secret`	Recherchez une valeur secrète et stockez-la en tant que variable nommée `"secret"`.
`secret_avec_champ`	Recherchez un secret avec une sortie de champ supplémentaire. Dans cet exemple, la recherche renverra la valeur de la `'note'` du secret.
`secret_avec_token_d'accès`	Recherchez un secret avec la valeur du jeton d'accès incluse dans la demande.
`fichier_secret_avec_etat`	Recherchez un secret avec le fichier d'état préconfiguré inclus dans la demande.

Demandes supplémentaires et champs

En plus du `secret_id`, plusieurs champs peuvent être inclus dans le `bitwarden.secrets.lookup`. L'objet JSON suivant comprend tous les champs qui peuvent être référencés dans la recherche du playbook :

```Bash
{
  "id": "be8e0ad8-d545-4017-a55a-b02f014d4158",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "projectId": "e325ea69-a3ab-4dff-836f-b02e013fe530",
  "key": "SES_KEY",
  "value": "0.982492bc-7f37-4475-9e60",
  "note": "",
  "creationDate": "2023-06-28T20:13:20.643567Z",
  "revisionDate": "2023-06-28T20:13:20.643567Z"
}
```

Pour récupérer des champs supplémentaires tels que `"note"`, la commande suivante peut être ajoutée au playbook :

```Bash
  vars:
        database_password: "{{ lookup('bitwarden.secrets.lookup', '0037ed90-efbb-4d59-a798-b1030124
87a0', field='note') }}"
```