

SECRETS MANAGER > OUTILS DE DÉVELOPPEMENT

Secrets Manager CLI

Afficher dans le centre d'aide:

<https://bitwarden.com/help/secrets-manager-cli/>

Secrets Manager CLI

L'interface en ligne de commande (CLI) de Secrets Manager est un outil puissant pour récupérer et injecter vos secrets. Le CLI de Secrets Manager peut être utilisé pour organiser votre coffre avec `créer`, `supprimer`, `éditer`, et `lister` vos secrets et projets.

Le CLI de Secrets Manager est auto-documenté. Depuis la ligne de commande, apprenez-en plus sur les commandes disponibles en utilisant :

Bash

```
bws --help, -h
```

Téléchargez et installez

Le CLI peut être utilisé sur plusieurs plateformes telles que Windows, macOS et les distributions Linux. Pour télécharger et installer le CLI de Secrets Manager :

Téléchargez le CLI de Secrets Manager à partir de <https://github.com/bitwarden/sdk/releases>.

Authentification

Le CLI de Secrets Manager peut se connecter en utilisant un [jeton d'accès](#) généré pour un certain [compte de service](#). Cela signifie que **seuls les secrets et les projets auxquels le compte de service a accès** peuvent être manipulés à l'aide du CLI. Il existe plusieurs façons d'authentifier une session CLI :

⇒Variable d'environnement

Vous pouvez authentifier une session CLI en enregistrant une variable d'environnement `BWS_ACCESS_TOKEN` avec la valeur de votre jeton d'accès, par exemple :

Bash

```
export BWS_ACCESS_TOKEN=0.48c78342-1635-48a6-accd-afbe01336365.C0tMmQqHnAp1h0gL8bngprlP0Yutt0:B3h5D+YgLvFiQhWkIq6Bow==
```

⇒En ligne

Vous pouvez authentifier des requêtes CLI individuelles en utilisant le drapeau `-t`, `--access-token` avec n'importe quelle commande individuelle, par exemple :

Bash

```
bws secret list --access-token 0.48c78342-1635-48a6-accd-afbe01336365.C0tMmQqHnAp1h0gL8bngprlP0Yutt0:B3h5D+YgLvFiQhWkIq6Bow==
```

⚠ Warning

Si votre flux de travail utilise de nombreuses sessions séparées (où chaque utilisation d'un jeton d'accès pour l'authentification constitue une "session") pour faire des demandes à partir de la même adresse IP dans un court laps de temps, vous pouvez rencontrer des limites de taux.

Commandes

Les commandes sont utilisées pour interagir avec le CLI de Secrets Manager. Les secrets et les projets peuvent être lus ou écrits en fonction des autorisations données à votre jeton d'accès spécifique. Pour plus de détails concernant les commandes disponibles pour `secret` et `projet`, utilisez :

- `bws secret --aide`
- `bws projet --aide`

📘 Note

À partir de la version 0.3.0 de Secrets Manager, la syntaxe CLI a été modifiée. La commande pour lister les secrets, par exemple, a changé de `bws list secrets` à `bws secret list`.

L'ancienne syntaxe restera temporairement prise en charge dans le CLI de Secrets Manager. Si vous n'êtes pas sûr de la version du CLI de Secrets Manager que vous utilisez, entrez `bws --version`.

secret

La commande `secrète` est utilisée pour accéder, manipuler et créer des [secrets](#). Comme pour toutes les commandes, les secrets et les [projets](#) en dehors du champ d'accès de votre jeton d'accès ne peuvent pas être lus ou écrits.

créer secret

Utilisez `bws secret create` pour créer un nouveau secret. Cette commande nécessite une `CLÉ`, une `VALEUR` et un `ID_PROJET` :

Bash

```
bws secret create <KEY> <VALUE> <PROJECT_ID>
```

Facultativement, vous pouvez ajouter une note en utilisant l'option `--note`. Par exemple:

Bash

```
bws secret create SES_KEY 0.982492bc-7f37-4475-9e60 f588b2f2-4780-4a78-be2a-b02d014d622f --note "API Key for AWS SES"
```

Cette commande, par défaut, renverra un objet JSON et enregistrera le secret dans Secrets Manager. Vous pouvez modifier le format de sortie en utilisant le drapeau `--output` (en savoir plus).

Bash

```
{
  "object": "secret",
  "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": "API Key for AWS SES",
  "creationDate": "2023-06-28T20:13:20.643567Z",
  "revisionDate": "2023-06-28T20:13:20.643567Z"
}
```

secret supprimer

Utilisez `bws secret supprimer` pour supprimer un ou plusieurs secrets désignés par les `SECRET_IDS`.

Bash

```
bws secret delete <SECRET_IDS>
```

Pour supprimer un seul secret avec l'`id` `be8e0ad8-d545-4017-a55a-b02f014d4158` :

Bash

```
bws secret delete be8e0ad8-d545-4017-a55a-b02f014d4158
```

Pour plusieurs secrets où les `ids` sont `382580ab-1368-4e85-bfa3-b02e01400c9f` et `47201c5c-5653-4e14-9007-b02f015b2d82` :

Bash

```
bws secret delete 382580ab-1368-4e85-bfa3-b02e01400c9f 47201c5c-5653-4e14-9007-b02f015b2d82
```

Sortie:

Bash

```
1 secret deleted successfully.
```

édition secrète

Pour éditer un secret, la structure suivante appliquera des modifications à la valeur choisie. Depuis le CLI, ces commandes peuvent éditer la clé secrète `KEY`, la note `VALUE`, la note `NOTE`, ou `PROJECT_ID`.

Bash

```
bws secret edit <SECRET_ID> --key <KEY> --value <VALUE> --note <NOTE> --project-id <PROJECT_ID>
```

Par exemple, si vous souhaitez ajouter une note à un secret existant :

Bash

```
bws secret edit be8e0ad8-d545-4017-a55a-b02f014d4158 --note "I am adding a note"
```

Note

Incluez des guillemets autour de la chaîne lors de l'édition d'une `NOTE` contenant des espaces.

Pour éditer plusieurs champs où `SES_KEY2` est la nouvelle `clé` et `0.1982492bc-7f37-4475-9e60` est la nouvelle `valeur` :

Bash

```
bws secret edit be8e0ad8-d545-4017-a55a-b02f014d4158 --key SES_KEY2 --value 0.1982492bc-7f37-4475-9e60
```

Sortie:

Bash

```
{
  "object": "secret",
  "id": "be8e0ad8-d545-4017-a55a-b02f014d4158",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "projectId": "e325ea69-a3ab-4dff-836f-b02e013fe530",
  "key": "SES_KEY2",
  "value": "0.1982492bc-7f37-4475-9e60",
  "note": "I am adding a note",
  "creationDate": "2023-06-28T20:13:20.643567Z",
  "revisionDate": "2023-06-28T20:45:37.46232Z"
}
```

obtenir secret

Utilisez `bws secret get` pour récupérer un secret spécifique :

Bash

```
bws secret get <SECRET_ID>
```

Par défaut, cette commande récupérera l'objet secret avec le `SECRET_ID`.

Bash

```
bws secret get be8e0ad8-d545-4017-a55a-b02f014d4158
```

Par défaut, `get` renverra des objets sous forme de tableau JSON, comme le montre l'exemple suivant. Vous pouvez modifier le format de sortie en utilisant le drapeau `--output` ([en savoir plus](#)).

Bash

```
{
  "object": "secret",
  "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"
}
```

liste secrète

Pour lister les secrets auxquels le compte de service peut accéder, utilisez la commande suivante :

Bash

```
bws secret list
```

Vous pouvez également lister uniquement les secrets dans un projet spécifique en utilisant la commande suivante, où `e325ea69-a3ab-4dff-836f-b02e013fe530` représente un identifiant de projet :

Bash

```
bws secret list e325ea69-a3ab-4dff-836f-b02e013fe530
```

Par défaut, `list` renverra des objets sous forme de tableau JSON, comme dans l'exemple suivant. Vous pouvez modifier le format de sortie en utilisant le drapeau `--output` ([en savoir plus](#)).

Bash

```
[
  {
    "object": "secret",
    "id": "382580ab-1368-4e85-bfa3-b02e01400c9f",
    "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
    "projectId": "e325ea69-a3ab-4dff-836f-b02e013fe530",
    "key": "Repository 1",
    "value": "1234567ertthrytkuy",
    "note": "Main Repo",
    "creationDate": "2023-06-27T19:25:15.822004Z",
    "revisionDate": "2023-06-27T19:25:15.822004Z"
  },
  {
    "object": "secret",
    "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"
  }
]
```

projet

La commande de projet est utilisée pour accéder, manipuler et créer des [projets](#). La portée de l'accès attribué à votre compte de service déterminera quelles actions peuvent être effectuées avec la commande `projet`.

Note

Les projets peuvent être créés par un compte de service avec un accès en lecture seule. Cependant, les projets existants qui n'ont pas été créés par le compte de service ne peuvent pas être édités sans accès en **lecture** et en **écriture**.

créer un projet

Utilisez `bws project create` pour créer un nouveau projet. Cette commande nécessite un `NOM`.

Bash

```
bws project create <NAME>
```

Dans cet exemple, un projet sera créé avec le nom `Mon projet`.

Bash

```
bws project create "My project"
```

Par défaut, `bws project create` renverra des objets sous forme de tableau JSON, comme dans l'exemple suivant. Vous pouvez modifier le format de sortie en utilisant le drapeau `--output` ([en savoir plus](#)).

Bash

```
{
  "object": "project",
  "id": "1c80965c-acb3-486e-ac24-b03000dc7318",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "name": "My project",
  "creationDate": "2023-06-29T13:22:37.942559Z",
  "revisionDate": "2023-06-29T13:22:37.942559Z"
}
```

projet supprimer

Utilisez `bws project supprimer` pour supprimer un ou plusieurs projets désignés par les `PROJECT_IDS`.

Bash

```
bws project delete <PROJECT_IDS>
```

Pour un seul projet où `f1fe5978-0aa1-4bb0-949b-b03000e0402a` représente le `ID_DU_PROJET` :

Bash

```
bws project delete f1fe5978-0aa1-4bb0-949b-b03000e0402a
```

Pour plusieurs projets où `1c80965c-acb3-486e-ac24-b03000dc7318` et `f277fd80-1bd2-4532-94b2-b03000e00c6c` représentent les `ID_DES_PROJETS` :

Bash

```
bws project delete 1c80965c-acb3-486e-ac24-b03000dc7318 f277fd80-1bd2-4532-94b2-b03000e00c6c
```

Sortie:

Bash

```
1 project deleted successfully.
```

édition de projet

En utilisant la commande `éditer`, vous pouvez changer le nom d'un projet avec l'entrée suivante :

Bash

```
bws project edit <PROJECT_ID> --name <NEW_NAME>
```

Par exemple, cette commande changera le nom du projet en `Mon projet 2`.

Bash

```
bws project edit 1c80965c-acb3-486e-ac24-b03000dc7318 --name "My project 2"
```

Par défaut, `éditer le projet bws` renverra des objets sous forme de tableau JSON, comme dans l'exemple suivant. Vous pouvez modifier le format de sortie en utilisant le drapeau `--output` ([en savoir plus](#)).

Bash

```
{
  "object": "project",
  "id": "1c80965c-acb3-486e-ac24-b03000dc7318",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "name": "My project 2",
  "creationDate": "2023-06-29T13:22:37.942559Z",
  "revisionDate": "2023-06-29T13:31:07.927829Z"
}
```

projet obtenir

La commande `get` récupère un projet spécifique auquel le compte de service connecté peut accéder depuis votre coffre. Les objets dans votre coffre auxquels le compte de service n'a pas accès ne peuvent pas être récupérés.

Bash

```
bws project get <PROJECT_ID>
```

Pour obtenir un projet spécifique, utilisez la commande suivante où `e325ea69-a3ab-4dff-836f-b02e013fe530` représente un `PROJECT_ID` :

Bash

```
bws project get e325ea69-a3ab-4dff-836f-b02e013fe530
```

Par défaut, `get` renverra des objets sous forme de tableau JSON, comme dans l'exemple suivant. Vous pouvez modifier le format de sortie en utilisant le drapeau `--output` ([en savoir plus](#)).

Bash

```
{
  "object": "project",
  "id": "e325ea69-a3ab-4dff-836f-b02e013fe530",
  "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
  "name": "App 1",
  "creationDate": "2023-06-27T19:24:42.181607Z",
  "revisionDate": "2023-06-27T19:24:42.181607Z"
}
```

liste de projets

Pour lister les projets auxquels ce compte de service a accès, utilisez la commande suivante :

Bash

```
bws project list
```

Par défaut, `list` renverra des objets sous forme de tableau JSON, comme dans l'exemple suivant. Vous pouvez modifier le format de sortie en utilisant le drapeau `--output` ([en savoir plus](#)).

Bash

```
[
  {
    "object": "project",
    "id": "e325ea69-a3ab-4dff-836f-b02e013fe530",
    "organizationId": "10e8cbfa-7bd2-4361-bd6f-b02e013f9c41",
    "name": "App 1",
    "creationDate": "2023-06-27T19:24:42.181607Z",
    "revisionDate": "2023-06-27T19:24:42.181607Z"
  },
  ...
]
```

configuration

La commande de configuration spécifie les paramètres du serveur que le CLI de Secrets Manager doit utiliser. Une utilisation principale de `bws config` est de connecter le CLI à un serveur Bitwarden auto-hébergé.

Les paramètres disponibles incluent `base-serveur`, `API-serveur`, et `identité-serveur`, par exemple:

Bash

```
bws config server-base https://my_hosted_server.com
```

Lorsqu'il est fait de cette manière, vos valeurs de serveur spécifiées seront enregistrées dans un profil par défaut dans un fichier `~/.bws/config`. Vous pouvez utiliser les options suivantes pour créer des profils alternatifs et des fichiers de configuration :

config --profil

Utilisez l'option `--profile` avec la commande `config` pour enregistrer les valeurs de serveur spécifiées dans des profils alternatifs, par exemple :

Bash

```
bws config server-base http://other_hosted_server.com --profile dev
```

Une fois créé, vous pouvez utiliser ce profil avec d'autres commandes pour acheminer les demandes vers le serveur spécifié, par exemple :

Bash

```
bws secret get 2863ced6-eba1-48b4-b5c0-afa30104877a --profile dev
```

config --fichier-config

Utilisez l'option `--config-file` avec la commande `config` pour enregistrer les valeurs de serveur spécifiées dans des fichiers de configuration alternatifs, par exemple pour enregistrer des valeurs dans un profil par défaut dans un nouveau fichier de configuration :

Bash

```
bws config server-base http://third_hosted_server.com --config-file ~/.bws/alt_config
```

Vous pouvez chaîner `--config-file` avec `--profile` pour enregistrer des valeurs dans des profils alternatifs dans des fichiers de configuration alternatifs, par exemple :

Bash

```
bws config server-base http://third_hosted_server.com --config-file ~/.bws/alt_config --profile alt_dev
```

Une fois créé, vous pouvez utiliser ce profil avec d'autres commandes pour acheminer les demandes vers le serveur spécifié, par exemple :

Bash

```
bws secret get 2863ced6-eba1-48b4-b5c0-afa30104877a --config-file ~/.bws/alt_config --profile alt_dev
```

Options

-o, --sortie

Par défaut, le CLI de Secrets Manager renverra un objet JSON ou un tableau d'objets JSON en réponse aux commandes. Le format de sortie peut être modifié pour répondre à vos besoins en utilisant le drapeau `-o`, `--output` accompagné de l'une des options suivantes :

- `json` : Par défaut. Sortie JSON.
- `yaml` : Sortie YAML.
- `table` : Affichez un tableau ASCII avec des clés en tant que titres de colonnes.
- `tsv` : Sortie de valeurs séparées par des onglets sans clés.
- `aucun` : Affiche uniquement les erreurs et les avertissements.
- `env` : Affichez les secrets au format CLÉ=VALEUR.

Par exemple, la commande:

Bash

```
bws secret get 2863ced6-eba1-48b4-b5c0-afa30104877a --output yaml
```

retournera ce qui suit:

Bash

```
object: secret
id: 2863ced6-eba1-48b4-b5c0-afa30104877a
organizationId: b8824f88-c57c-4a36-8b1a-afa300fe0b52
projectId: 1d0a63e8-3974-4cbd-a7e4-afa30102257e
key: Stripe API Key
value: osiundfpowubefpouwef
note: 'These are notes.'
creationDate: 2023-02-08T15:48:33.470701Z
revisionDate: 2023-02-08T15:48:33.470702Z
```

Note

Lors de l'utilisation du format de sortie env, si le nom de la clé n'est pas conforme à POSIX, cette paire clé-valeur sera mise en commentaire et un commentaire au bas de la sortie sera affiché indiquant que la sortie a été modifiée.

En utilisant le `--output env flag`, par exemple:

Bash

```
bws secret list --output env
```

retournera ce qui suit:

Bash

```
this_is_a_keyname="this is a key value"
CLOUDFLARE_API_TOKEN="123412341234123412341234"
# This is an invalid keyname="this will get commented-out"

# one or more secrets have been commented-out due to a problematic key name
```

-c, --couleur

La sortie peut être davantage personnalisée en indiquant si vous souhaitez une sortie colorée. Les valeurs disponibles pour cette option sont `oui`, `non`, et `auto`.

--jeton-d'accès

Vous pouvez authentifier des requêtes CLI individuelles en utilisant l'option `-t`, `--access-token` avec n'importe quelle commande individuelle, par exemple :

Bash

```
bws secret list --access-token 0.48c78342-1635-48a6-accd-afbe01336365.C0tMmQqHnAp1h0gL8bngprLP0Yutt  
0:B3h5D+YgLvFiQhWkIq6Bow==
```

--profil

Utilisez l'option `--profile` avec les commandes `list` ou `get` pour spécifier quel profil utiliser, par exemple :

Bash

```
bws secret get 2863ced6-eba1-48b4-b5c0-afa30104877a --profile dev
```

Reportez-vous à la commande `config` ([ici](#)) pour obtenir de l'aide pour comprendre et configurer des profils alternatifs.

--fichier-de-configuration

Utilisez l'option `--config-file` avec l'option `--profile` et les commandes `list` ou `get` pour spécifier quel profil utiliser à partir de quel fichier de configuration, par exemple :

Bash

```
bws secret get 2863ced6-eba1-48b4-b5c0-afa30104877a --config-file ~/.bws/alt_config --profile alt_d  
ev
```

Reportez-vous à la commande `config` ([ici](#)) pour obtenir de l'aide pour comprendre et configurer des fichiers de paramètres et des profils alternatifs.

--url-du-serveur

Cette option peut être utilisée pour définir l'URL du serveur à laquelle le CLI enverra la requête associée à une commande donnée, par exemple :

Bash

```
bws list secrets --server-url http://my_hosted_server.com
```

Cette option remplacera toutes les URLs configurées via la commande `config` (voir [ici](#)).

--aide

Utilisez cette option pour imprimer de l'aide pour n'importe quelle commande `bws`.

--version

Utilisez cette option pour imprimer la version du client `bws` que vous utilisez.