

ADMINISTRADOR DE SECRETOS > HERRAMIENTAS DE DESARROLLO

Administrador de secretos CLI

Ver en el centro de ayuda:
<https://bitwarden.com/help/secrets-manager-cli/>

Administrador de secretos CLI

La Interfaz de línea de comandos (ILC) del Administrador de secretos es una herramienta poderosa para recuperar e inyectar tus secretos. El ILC del Administrador de secretos puede ser utilizado para organizar tu caja fuerte con **crear**, **eliminar**, **editar**, y **listar** tus secretos y proyectos.

El ILC del Administrador de secretos está auto documentado. Desde la línea de comandos, aprende más sobre los comandos disponibles utilizando:

Bash

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

Descargar e instalar

El ILC se puede utilizar en varias plataformas en Windows, macOS y distribuciones de Linux. Para descargar e instalar el ILC del Administrador de secretos:

Descarga el ILC del Administrador de secretos desde <https://github.com/bitwarden/sdk/releases>.

Autenticación

El Administrador de secretos ILC 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. Hay algunas formas en las que puedes autenticar una sesión de ILC:

⇒ Variable de entorno

Puedes autenticar una sesión de ILC 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.C0tMmQqHnAp1h0gL8bngprlP0Yutt0:B3h5D+YgLvFiQhWkIq6Bow==
```

⇒ En línea

Puedes autenticar solicitudes individuales de ILC utilizando la bandera **-t**, **--access-token** con cualquier comando individual, por ejemplo:

Bash

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

⚠ Warning

Si su flujo de trabajo utiliza muchas sesiones separadas (donde cada uso de un token de acceso para autenticar constituye una "sesión") para hacer solicitudes desde la misma dirección IP en un corto período de tiempo, puede encontrar límites de tasa.

Órdenes

Los comandos se utilizan para interactuar con el ILC del Administrador de secretos. Los secretos y proyectos pueden ser leídos o escritos dependiendo de los permisos otorgados a tu token de acceso específico. Para obtener detalles adicionales sobre los comandos disponibles para **secreto** y **proyecto**, use:

- `bws secreto --ayuda`
- `bws proyecto --ayuda`

📌 Note

A partir de la versión 0.3.0 del Administrador de secretos, la sintaxis de ILC ha sido cambiada. El comando para listar secretos, por ejemplo, ha cambiado de `bws list secrets` a `bws secret list`.

La antigua sintaxis seguirá siendo compatible temporalmente en el ILC del Administrador de secretos. Si no estás seguro de qué versión del ILC del Administrador de secretos estás utilizando, ingresa `bws --version`.

secreto

El comando **secreto** se utiliza para acceder, manipular y crear **secretos**. Como con todos los comandos, los secretos y **proyectos** fuera del alcance de acceso de tu token no pueden ser leídos o escritos.

secreto crear

Usa `bws secret create` para crear un nuevo secreto. Este comando requiere una **CLAVE**, **VALOR**, y **ID_DE_PROYECTO**:

Bash

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

Opcionalmente, puedes agregar una nota usando la opción `--nota` . Por ejemplo:

Bash

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

Este comando, por defecto, devolverá un objeto JSON y guardará el secreto en el Administrador de secretos. Puedes cambiar el formato de salida usando la bandera `--output` ([aprende más](#)).

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"
}
```

secreto eliminar

Utilice `bws secret eliminar` para eliminar uno o más secretos designados por los `SECRET_IDS`.

Bash

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

Para eliminar un solo secreto con el `id be8e0ad8-d545-4017-a55a-b02f014d4158`:

Bash

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

Para múltiples secretos donde los `ids` son `382580ab-1368-4e85-bfa3-b02e01400c9f` y `47201c5c-5653-4e14-9007-b02f015b2d82`:

Bash

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

Salida:

Bash

```
1 secret deleted successfully.
```

edición secreta

Para editar un secreto, la siguiente estructura aplicará cambios al valor elegido. Desde la ILC estos comandos pueden editar la clave secreta **KEY**, **VALUE**, la nota **NOTE**, o **PROJECT_ID**.

Bash

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

Por ejemplo, si deseas agregar una nota a un secreto existente:

Bash

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

Note

Incluya comillas alrededor de la cadena al editar una **NOTA** que contiene espacios.

Para editar múltiples campos donde **SES_KEY2** es la nueva **clave** y **0.1982492bc-7f37-4475-9e60** es el nuevo **valor**:

Bash

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

Salida:

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"
}
```

secreto obtener

Usa `bws secret get` para recuperar un secreto específico:

Bash

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

Por defecto, este comando recuperará el objeto secreto con el `SECRET_ID`.

Bash

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

Por defecto, `get` devolverá objetos como un array JSON, como se muestra en el siguiente ejemplo. Puedes cambiar el formato de salida usando la bandera `--output` ([aprende más](#)).

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"
}
```

lista secreta

Para enumerar los secretos a los que la cuenta de servicio puede acceder, use el siguiente comando:

Bash

```
bws secret list
```

También puedes listar solo los secretos en un proyecto específico utilizando el siguiente comando, donde `e325ea69-a3ab-4dff-836f-b02e013fe530` representa un identificador de proyecto:

Bash

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

Por defecto, `list` devolverá objetos como un array JSON, como en el siguiente ejemplo. Puedes cambiar el formato de salida usando la bandera `--output` ([aprende más](#)).

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": "1234567ertthrjytkey",
    "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"
  }
]
```

proyecto

El comando del proyecto se utiliza para acceder, manipular y crear **proyectos**. El alcance del acceso asignado a su cuenta de servicio determinará qué acciones se pueden completar con el comando de **proyecto**.

Note

Los proyectos pueden ser creados por una cuenta de servicio con acceso de sólo lectura. Sin embargo, los proyectos existentes que no fueron creados por la cuenta de servicio no pueden ser editados sin acceso de **lectura** y **escritura**.

crear proyecto

Usa **bws project create** para crear un nuevo proyecto. Este comando requiere un **NOMBRE**.

Bash

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

En este ejemplo, se creará un proyecto con el nombre **Mi proyecto**.

Bash

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

Por defecto, `bws project create` devolverá objetos como un array JSON, como en el siguiente ejemplo. Puedes cambiar el formato de salida usando la bandera `--output` ([aprende más](#)).

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"
}
```

eliminar proyecto

Utilice `bws project eliminar` para eliminar uno o más proyectos designados por los **PROJECT_IDS**.

Bash

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

Para un solo proyecto donde `f1fe5978-0aa1-4bb0-949b-b03000e0402a` representa el **ID_DEL_PROYECTO**:

Bash

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

Para múltiples proyectos donde `1c80965c-acb3-486e-ac24-b03000dc7318` y `f277fd80-1bd2-4532-94b2-b03000e00c6c` representan los **ID_DE_PROYECTOS**:

Bash

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

Salida:

Bash

```
1 project deleted successfully.
```

edición del proyecto

Usando el comando **editar** puedes cambiar el nombre de un proyecto con la siguiente entrada:

Bash

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

Por ejemplo, este comando cambiará el nombre del proyecto a **Mi proyecto 2**.

Bash

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

Por defecto, **editar proyecto bws** devolverá objetos como un array JSON, como en el siguiente ejemplo. Puedes cambiar el formato de salida usando la bandera **--output** ([aprende más](#)).

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"
}
```

proyecto consigue

El comando `get` recupera un proyecto específico al que la cuenta de servicio iniciada puede acceder desde tu caja fuerte. Los objetos en su caja fuerte a los que la cuenta de servicio no tiene acceso no pueden ser recuperados.

Bash

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

Para obtener un proyecto específico, use el siguiente comando donde `e325ea69-a3ab-4dff-836f-b02e013fe530` representa un `PROJECT_ID`:

Bash

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

Por defecto, `get` devolverá objetos como un array JSON, como en el siguiente ejemplo. Puedes cambiar el formato de salida usando la bandera `--output` ([aprende más](#)).

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"
}
```

lista de proyectos

Para enumerar los proyectos a los que esta cuenta de servicio tiene acceso, use el siguiente comando:

Bash

```
bws project list
```

Por defecto, `list` devolverá objetos como un array JSON, como en el siguiente ejemplo. Puedes cambiar el formato de salida usando la bandera `--output` ([aprende más](#)).

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"
  },
  ...
]
```

configuración

El comando de configuración especifica los ajustes del servidor para que los use el ILC del Administrador de secretos. Un uso principal de `bws config` es conectar la ILC a un servidor de Bitwarden autoalojado.

Los ajustes disponibles incluyen `base-del-servidor`, `API-del-servidor`, y `identidad-del-servidor`, por ejemplo:

Bash

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

Cuando se hace de esta manera, los valores especificados de su servidor se guardarán en un perfil predeterminado en un archivo `~/.bws/config`. Puedes usar las opciones siguientes para crear perfiles alternativos y archivos de configuración:

config --perfil

Utilice la opción `--profile` con el comando `config` para guardar los valores del servidor especificados en perfiles alternativos, por ejemplo:

Bash

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

Una vez creado, puedes usar ese perfil con otros comandos para dirigir las solicitudes al servidor especificado, por ejemplo:

Bash

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

config --archivo-de-configuración

Utilice la opción `--config-file` con el comando `config` para guardar los valores del servidor especificados en archivos de configuración alternativos, por ejemplo, para guardar valores en un perfil predeterminado en un nuevo archivo de configuración:

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

Puedes encadenar `--config-file` con `--profile` para guardar valores en perfiles alternativos en archivos de configuración alternativos, por ejemplo:

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

Una vez creado, puedes usar ese perfil con otros comandos para dirigir las solicitudes al servidor especificado, por ejemplo:

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

Opciones

`-o, --salida`

Por defecto, el ILC del Administrador de secretos devolverá un objeto JSON o un array de objetos JSON en respuesta a los comandos. El formato de salida se puede modificar para adaptarse a tus necesidades utilizando la bandera `-o, --output` junto con una de las siguientes opciones:

- `json`: Predeterminado. Salida JSON.
- `yaml`: Salida YAML.
- `tabla`: Muestra una tabla ASCII con las claves como encabezados de columnas.
- `tsv`: Salida de valores separados por pestañas sin claves.
- `ninguno`: Solo muestra errores y advertencias.
- `env`: Salida de secretos en formato CLAVE=VALOR.

Por ejemplo, el comando:

Bash

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

devolverá lo siguiente:

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: osiundfpowubefpouef
note: 'These are notes.'
creationDate: 2023-02-08T15:48:33.470701Z
revisionDate: 2023-02-08T15:48:33.470702Z
```

Note

Al usar el formato de salida `env`, si el nombre de la clave no cumple con POSIX, ese par de clave-valor será comentado y se mostrará un comentario en la parte inferior de la salida indicando que la salida ha sido modificada.

Usando la bandera `--output env`, por ejemplo:

Bash

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

devolverá lo siguiente:

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, --color

La salida se puede personalizar aún más indicando si desea una salida en color. Los valores disponibles para esta opción son **sí**, **no** y **automático**.

--token-de-acceso

Puedes autenticar solicitudes individuales de ILC utilizando la opción **-t, --access-token** con cualquier comando individual, por ejemplo:

Bash

```
bws secret list --access-token 0.48c78342-1635-48a6-accd-afbe01336365.C0tMmQqHnAp1h0gL8bngprlP0Yutt
0:B3h5D+YgLvFiQhWkIq6Bow==
```

--perfil

Utilice la opción **--profile** con los comandos de **list** o **get** para especificar qué perfil usar, por ejemplo:

Bash

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

Consulte el comando **config** ([aquí](#)) para obtener ayuda para entender y configurar perfiles alternativos.

--archivo-de-configuración

Utilice la opción **--config-file** con la opción **--profile** y los comandos de **listar** o **obtener** para especificar qué perfil de qué archivo de configuración usar, por ejemplo:

Bash

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

Consulte el comando **config** ([aquí](#)) para obtener ayuda para entender y configurar archivos de ajustes alternativos y perfiles.

--url-del-servidor

Esta opción se puede utilizar para establecer la URL del servidor a la que el ILC enviará la solicitud asociada con un comando dado, por ejemplo:

Bash

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

Esta opción anulará cualquier URL configurada a través del comando **config** ([ver aquí](#)).

--ayuda

Utilice esta opción para imprimir ayuda para cualquier comando **bws** dado.

--versión

Utilice esta opción para imprimir la versión del cliente **bws** que está utilizando.