

Migration Script

View in the help center:
<https://bitwarden.com/help/migration-script/>

Migration Script

The [Bitwarden public API](#) allows administrators to automate administrative tasks using scripts. The script documented in this article is written to help Bitwarden customers migrate their existing setup from a previous Bitwarden Password Manager environment into a new organization, providing a way to migrate organization vault data, groups, and associated groups' and members' permissions to a new installation.

The script is written in Python and can be run on any operating system with Python v3 installed. Download the script and an example configuration file [here](#).

Installation and setup

System requirements

Other than the default libraries shipped with most Python distributions included by default on Linux and macOS, and [available](#) for Windows), this script requires an additional module called **requests** be installed before the script can run successfully.

A common tool to install Python modules is called pip. To install the module using pip:

Bash

```
pip3 install requests
```

Note

pip3 – Some machines will have multiple versions of Python installed. Using **pip3**, instead of just **pip**, specifies that you install **requests** with Python v3. If your machine only has one Python version installed, use **pip** instead.

Required files

The above download contains two files:

- **bwAdminTools.py**: This is the script you will need to execute migration. It requires a fully-configured configuration file.
- **config-example.cfg**: This is the configuration file required for migration, which you will need to create and setup before running the script.

Unpack the **.zip** and save these files to the same directory. Once you do, add the following files to the same directory:

- Bitwarden Password Manager [CLI native executable](#).

Create destination organization

Before you can continue, you must create the destination organization that you'll be migrating to. [Learn how to create an organization](#).

Note

We recommend inviting users prior to running the migration script. Users must be in at least an invited state in order to migrate group and permissions settings.

Migrate with Self-hosted Instance

If your organization license originated from the US cloud server, and self-hosted instance was enabled using US cloud credentials, the follow steps will be required in order to migrate the self hosted instance and organization credentials to the EU:

1. Instruct all organization members to [export their individual vaults](#).



Individually download any file attachments for vault items and note which items they belong to.

2. [Request a new installation Id and Key](#). Be sure to set the **Data Region** to the destination you wish to migrate the Bitwarden instance to.
3. Access the `./bwdata/env/global.override.env` file on your self hosted instance. Update the environment variables following the example [here](#).
4. Login and access the cloud organization and download a new subscription license file using the new EU or US Installation Id.
5. Create a new organization on the self-hosted instance. Manually apply the new subscription license file to the newly created organization. The subscription license **can not be applied an existing organization** on the self-hosted instance.
6. Set up your new organization, configuring things like enterprise policies, login with SSO, constructing group-collection relationships, and inviting users with Directory Connector or SCIM. For help, refer to the [Proof-of-Concept Checklist](#).
7. Instruct organization members to import their individual vaults.

Environment configuration

Before running any `bwAdminTools.py` script functions, you will need to create a configuration file. Copy the contents of `config-example.cfg` into a new `config.cfg` file in the same directory, and fill in the following variables. Note that, as this is a migration script, variables are broken into **Source** and **Destination** groupings in this documentation:

Source organization variable

Variable description

`bw_vault_uri=`

FQDN of your source web vault, e.g. `https://company.bitwarden.com` if you're self-hosting or `https://vault.bitwarden.com` if you're using US-based Bitwarden cloud services.

`bw_org_client_id=`

Source organization API key client ID. [Learn where to find it](#).

`bw_org_client_secret=`

Source organization API key client secret. [Learn where to find it](#).

Source organization variable	Variable description
bw_org_id=	Source organization's GUID. Copy the <code>_client_id=</code> value and remove the <code>organization.</code> piece.
bw_acc_client_id	Source organization admin's or owner's personal API key client ID. Learn where to find it.
bw_acc_client_secret=	Source organization admin's or owner's personal API key client secret. Learn where to find it.

Destination organization variable	Variable description
dest_bw_vault_uri=	FQDN of your source web vault, e.g. <code>https://company.bitwarden.com</code> if you want to self-host or <code>https://vault.bitwarden.eu</code> if you want to use EU-based Bitwarden cloud services.
dest_bw_org_client_id=	Destination organization API key client ID. Learn where to find it.
dest_bw_org_client_secret=	Destination organization API key client secret. Learn where to find it.
dest_bw_org_id=	Destination organization's GUID. Copy the <code>_client_id=</code> value and remove the <code>organization.</code> piece.
dest_bw_acc_client_id=	Destination organization admin's or owner's personal API key client ID. Learn where to find it.
dest_bw_ac_client_secret=	Destination organization admin's or owner's personal API key client secret. Learn where to find it.

Once you've setup these variables, you're ready to start migration using the `bwAdminTools.py` script functions.

Script Functions

From the directory where you've stored your `bwAdminTools.py` file, `config.cfg` file, and Password Manager CLI executable, you can run the following commands:

Note

`python3` – Some machines will have multiple versions of Python installed. Using `python3`, instead of just `python`, specifies that commands run with Python v3. If your machine only has one Python version installed, use `python` instead. Some distributions will also have a `python` instead of `python3` binary for v3.

- To print script helper text:

Bash

```
python3 bwAdminTools.py -h
```

- To compare source and destination organizations:

Bash

```
python3 bwAdminTools.py -c diffbw
```

- To migrate organization vault data, groups, and groups' permissions from a source organization to a destination organization:

Bash

```
python3 bwAdminTools.py -c migratebw
```

Users must be in at least an invited state in the destination organization for `migratebw` to be successful.

- To migrate members' permissions (outside of groups) from a source organization to a destination organization:

Bash

```
python3 bwAdminTools.py -c migratebwusers
```

Users must be in at least an invited state in the destination organization for `migratebwusers` to be successful.

- To delete all collections from the source organization:

Bash

```
python3 bwAdminTools.py -c purgecol
```

- To delete all collections from the destination organization:

Bash

```
python3 bwAdminTools.py -c purgecoldest
```

- To delete all groups from the source organization:

Bash

```
python3 bwAdminTools.py -c purgegroup
```

- To delete all groups from the destination organization:

Bash

```
python3 bwAdminTools.py -c purgegroupdest
```