bitwarden | Help Center Article

SECRETS MANAGER > INTEGRATIONS

Terraform Provider

View in the help center:
https://bitwarden.com/help/terraform-provider/

Terraform Provider

Bitwarden offers a Terraform Provider that can fetch, create, and manage secrets, helping secure your infrastructure secrets using Terraform. Further provider documentation is available in the Bitwarden Terraform registry.

Requirements

- Terraform version 1.5 or higher.
- A Secrets Manager organization with a machine account and attached access token.

> **Tip**
>
> We recommend:
>
> - Using the Secrets Manager web app to dictate what projects and secrets your machine account has access to **before** proceeding with configuration.
> - Using the Secrets Manager web app to generate an access token **when you're ready to configure the Terraform Provider**, as access token values can only be copied on creation.

Configuration

At a minimum, your Terraform configuration file(s) (`.tf`) must include the following:

Bash

```
terraform {
  required_providers {
    bitwarden-secrets = {
      source = "registry.terraform.io/bitwarden/bitwarden-secrets"
    }
  }
}
```

Several optional attributes can be added to your `.tf` files. Some of these values should be considered sensitive. All of these values can instead be provided by an environment variable, however all of these values **must** be provided using one of those two methods:

Attribute	Equivalent variable	Description
`access_token`	`BW_ACCESS_TOKEN`	(Sensitive) The access token value for the configured machine account. This will grant the Terraform Provider access to only specific data in Secrets Manager.
`organization_id`	`BW_ORGANIZATION_ID`	The unique identifier of your organization. Available from the address bar when you're logged in to the Secrets Manager web app.
`api_url`	`BW_API_URL`	URI of the Bitwarden Secrets Manager `/api` endpoint. For US and EU cloud-hosted customers, this will be `https://api.bitwarden.com` and `https://identity.bitwarden.eu` respectively, and for self-hosted customers will be determined by the deployment.
`identity_url`	`BW_IDENTITY_URL`	URI of the Secrets Manager `/identity` endpoint. For US and EU cloud-hosted customers, this will be `https://api.bitwarden.com` and `https://identity.bitwarden.eu` respectively, and for self-hosted customers will be determined by the deployment.

A `.tf` file with all attributes explicitly included, instead of passed by environment variables, should look like the following:

Bash

```
terraform {
  required_providers {
    bitwarden-secrets = {
      source = "registry.terraform.io/bitwarden/bitwarden-secrets"
    }
  }
}

provider "bitwarden-secrets" {
  api_url         = "https://api.bitwarden.com"
  identity_url    = "https://identity.bitwarden.com"
  access_token    = "<access_token_value>"
  organization_id = "< organization_unique_identifier>"
}
```

Quick start

The following steps will walk you through bringing a secret that exists in Bitwarden Secrets Manager under Terraform management by adding it to both state and configuration:

1. Use the `bitwarden-secrets_secret` resource to add a secret to a `.tf` configuration file:

```
Bash

resource "bitwarden-secrets_secret" "my_secret" {}
```

2. Use the `terraform import` command in the Terraform CLI to import the secret to state, substituting the indicated placeholder value with the unique identifier of the secret (which can be copied directly from the secret entry in the Secrets Manager web app):

```
Bash

terraform import "bitwarden-secrets_secret.my_secret" "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
```

3. With your secret imported to the Terraform state, use the `terraform show` command in the Terraform CLI to show the imported data. From the output, copy the key value and add it to the `.tf` configuration to complete setup:

```
Bash

resource "bitwarden-secrets_secret" "my_secret" {
  key = "db_admin_password"
}
```

The `bitwarden-secrets_secret` resource will use the key value (in this case, `db_admin_password`) to manipulate the secret in further operations, ensuring the secret stays secure and synced between Bitwarden Secrets Manager and Terraform.

Data sources

bitwarden-secrets_projects

The `bitwarden-secrets_projects` data source fetches a list of all projects accessible by the machine account. The following is an example `data` block with a `bitwarden-secrets_projects` data source declaration and example `output` block that references it using the `.projects` attribute:

```Bash
data "bitwarden-secrets_projects" "example" {}

output "example" {
  value = data.bitwarden-secrets_projects.projects
}
```

For each project, the following attributes can be exported in output:

- `creation_date`: (String) The timestamp at which the project was created.
- `id`: (String) The unique identifier of the project.
- `name`: (String) The name of the project.
- `organization_id`: (String) The unique identifier of the organization the project belongs to.
- `revision_date`: (String) The timestamp at which the project was most recently revised.

bitwarden-secrets_list_secrets

The `bitwarden-secrets_list_secrets` data source fetches a list of all secrets accessible by the machine account. The following is an example `data` block with a `bitwarden-secrets_list_secrets` data source declaration and example `output` block that references it using the `.secrets` attribute:

```Bash
data "bitwarden-secrets_list_secrets" "example" {}

output "example" {
  value = data.bitwarden-secrets_list_secrets.secrets
}
```

For each secret, the following attributes can be exported in output:

- `id`: (String) The unique identifier of the secret.
- `key`: (String) The key associated with the secret, referred to as "Name" within the Secrets Manager UI.

> **Note**
>
> The `bitwarden-secrets_list_secrets` data source does not fetch secret values.

bitwarden-secrets_secret

The `bitwarden-secrets_secret` data source fetches a particular secret, which must be accessible by the machine account. The following arguments are required when fetching a secret with the `bitwarden-secrets_secret` data source:

- `id`: (String) The unique identifier of the secret to fetch. This value can be copied directly from the secret entry in the Secrets Manager web app.

The following is an example `data` block with a `bitwarden-secrets_secret` data source declaration and example `output` block that references all exportable attributes:

Bash

```
data "bitwarden-secrets_secret" "example" {
  id = "e6a8066c-81e6-428e-bf5d-b1b900fe1b42"
}

output "example" {
  value = {
    id  = data.bitwarden-secrets_secret.secret.id
    key = data.bitwarden-secrets_secret.secret.key
    value           = data.bitwarden-secrets_secret.secret.value #The actual secret value is marked
sensitive and will not be printed to stdout
    note            = resource.bitwarden-secrets_secret.secret.note
    project_id      = resource.bitwarden-secrets_secret.secret.project_id
    organization_id = resource.bitwarden-secrets_secret.secret.organization_id
    creation_date   = resource.bitwarden-secrets_secret.secret.creation_date
    revision_date   = resource.bitwarden-secrets_secret.secret.revision_date
  }
}
```

For each secret, the following attributes can be exported in output:

- `id`: (String) The unique identifier of the secret.

- `key`: (String) The key associated with the secret, referred to as "Name" within the Secrets Manager UI.
- `value`: (String) The value associated with the secret. Considered sensitive and never printed to stdout.
- `note`: (String) Any text saved in the secret's **Notes** field.
- `project_id`: (String) The unique identifier of the project the secret belongs to.
- `organization_id`: (String) The unique identifier of the organization the secret belongs to.
- `creation_date`: (String) The timestamp at which the secret was created.
- `revision_date`: (String) The timestamp at which the secret was most recently revised.

Resources

bitwarden-secrets_secret

The `bitwarden-secrets_secret` resource can be used create new or manage existing secrets in Bitwarden Secrets Manager. At a minimum, the following arguments are required for a `bitwarden-secrets_secret` resource block declaration:

- `key`: (String) The key associated with the secret, referred to as "Name" within the Secrets Manager UI.

The following is an example `resource` block with a `bitwarden-secrets_secret` resource declaration:

Bash

```
resource "bitwarden-secrets_secret" "db_admin_secret" {
  key = "db_admin_password"
  value      = var.value #It is not recommended to provide the actual secret value via configuratio
n file! By using a terraform variable, users can inject the secret value during runtime via environ
ment variables.
  project_id = var.project_id
  note       = "The secret value was provided via terraform configuration."
}
```

Additional optional arguments, as seen in the above example, include:

- `value`: (String) The value of the secret. It is **not recommended** to provide the secret value via configuration file. By using a terraform variable, users can inject the secret value during runtime via environment variables.
- `project_id`: (String) The unique identifier of the project the secret should be added to. The machine account **must** have access to that project.

- `note`: (String) Any text to save in the secret's **Notes** field.

Secret value generation

If no secret value is provided, the Terraform Provider will generate one for you. **This is the suggested approach**. You can specify optional attributes to to customize value generation, for example:

```
Bash

resource "bitwarden-secrets_secret" "db_admin_secret" {
  key         = "db_admin_password"
  project_id  = var.project_id
  length      = 32
  special     = true
  min_special = 5
}
```

For each secret, the following attributes can be used to customize value generation:

- `avoid_ambiguous`: (Boolean) Defaults to `false`. When set to `true`, the generated value will not contain ambiguous characters (`I`, `l`, `1`, `0`, `0`).
- `length`: (Number) Defaults to `64` characters. When set to another number, the generated value will be that number of characters.
- `lowercase`: (Boolean) Defaults to `true`. If set to `false`, the generated value will not contain lowercase characters.
  - `min_lowercase`: (Number) Ignored if `lowercase` is `false`. If set to a number, the generated value will contain at least that number of lowercase characters (must be between 1-9).
- `uppercase`: (Boolean) Defaults to `true`. If set to `false`, the generated value will not contain uppercase characters.
  - `min_uppercase`: (Number) Ignored if `uppercase` is `false`. If set to a number, the generated value will contain at least that number of uppercase characters (must be between 1-9).
- `numbers`: (Boolean) Defaults to `true`. If set to `false`, the generated value will not contain numbners (`0` - `9`).
  - `min_numbers`: (Number) Ignored if `numbers` is `false`. If set to a number, the generated value will contain at least that number of numbers (must be between 1-9).
- `special`: (Boolean) Defaults to `true`. If set to `false`, the generated value will not contain special characters (`@`, `#`, `$`, `%`, `^`, `&`, `*`).
  - `min_special`: (Number) Ignored if `special` is `false`. If set to a number, the generated value will contain at least that number of special characters (must be between 1-9).