bitwarden | Artikel Helpcentrum

GEHEIMEN MANAGER > INTEGRATIES

Secrets Manager Kubernetes Operator

Weergeven in het Helpcentrum:
https://bitwarden.com/help/secrets-manager-kubernetes-operator/

Secrets Manager Kubernetes Operator

> ⚠ **Warning**
>
> The Bitwarden Secrets Manager Helm integration is currently in **Beta status**. Some features may not provide full functionality at this time.

The Bitwarden Secrets Manager Kubernetes Operator will allow teams to integrate Secrets Manager into Kubernetes workflows securely and efficiently. Using the operator, which is deployed using Helm package manager, secrets can be stored and retrieved from Secrets Manager.

Bitwarden Secrets Manager Kubernetes Operator

The `sm-operator` uses a controller to synchronize Bitwarden secrets into Kubernetes secrets. The operator registers the Custom Resource Definition: `BitwardenSecret` into the Kubernetes cluster. The cluster will listen for the newly registered `BitwardenSecret`, and synchronize on a configurable interval.

Requirements

To get started, an active Bitwarden organization with Secrets Manager is required. Additionally, one or more access tokens associated with a machine account are required.

Additional dependencies

- kubectl
- Helm 3

Add the repository to Helm

Add the Bitwarden Secrets Manager chart repository:

Plain Text

```
helm repo add bitwarden https://charts.bitwarden.com/
```

Update information of locally available charts:

Plain Text

```
helm repo update
```

Installation

Create a configuration file

Create a custom values file used for deployment:

Plain Text

```
helm show values bitwarden/sm-operator --devel > my-values.yaml
```

Update configuration file

Locate `my-values.yaml` and fill out required values. An example can be located in the Bitwarden repository. We recommend that the following values be adjusted for your setup:

Value	Description
`settings.bwSecretsManagerRefreshInterval`	How often the secrets synchronize (in seconds). Minimum value is 180.
`settings.cloudRegion`	Self-hosted users set to `US` or `EU` cloud region for sync. See enable cloud communication for additional information.
`settings.bwApiUrlOverride`	For self-hosted users only. This is the URL for your instance API.
`settings.bwIdentityUrlOverride`	For self-hosted users only. This is the URL for your instance's identity service.
`containers.enableSeccompProfileRuntimeDefault`	Set to `false` to work on older Kubernetes versions (< 1.19) or on vendor versions that do not support this field by default (such as OpenShift < 4.11). This setting is recommended for most common cases that do not require escalating privileges to make containers restrictive. See Kubernetes documentation for more information.

> **ⓘ Note**
>
> To use a different operate image version than the one included with the chart, update: `containers.manager.image.tag`.

Upgrade Helm chart

Once your `values.yaml` file has been configured, upgrade the release to a new chart by running:

Plain Text

```
helm upgrade sm-operator bitwarden/sm-operator -i --debug -n sm-operator-system --create-namespace --values my-values.yaml --devel
```

This command installs or upgrades a release with the name `sm-operator`, in the namespace `sm-operator-system`, with the values from `my-values.yaml`.

> **ⓘ Note**
>
> To see information for the `helm install` or `helm upgrade` commands, run `helm install --help` or `helm upgrade --help`.

Create Bitwarden secrets

To synchronize secrets stored in Bitwarden Secrets Manager into Kubernetes secrets, we must create a BitwardenSecret object.

1. Create a Kubernetes secret to authenticate with Secrets Manager:

   Plain Text

   ```
   kubectl create secret generic bw-auth-token -n <YOUR_NAMESPACE> --from-literal=token="<TOKEN_HERE>"
   ```

   > **⚠ Warning**
   >
   > This command is recorded in your shell history. To avoid exposing access token data, consider deploying with an ephemeral pipeline agent.

Deploy BitwardenSecret

The `BitwardenSecret` object is the synchronization setting that will be used by the operator to create and synchronize a Kubernetes secret. The Kubernetes secret belongs to a namespace and will be injected with the data that the Secrets Manager machine account has access to.

Example BitwardenSecret deployment with custom mapping:

Plain Text

```
cat <<EOF | kubectl apply -n <YOUR_NAMESPACE> -f -
apiVersion: k8s.bitwarden.com/v1
kind: BitwardenSecret
metadata:
  labels:
    app.kubernetes.io/name: bitwardensecret
    app.kubernetes.io/instance: bitwardensecret-sample
    app.kubernetes.io/part-of: sm-operator
    app.kubernetes.io/managed-by: kustomize
    app.kubernetes.io/created-by: sm-operator
  name: bitwardensecret-sample
spec:
  organizationId: "a08a8157-129e-4002-bab4-b118014ca9c7"
  secretName: bw-sample-secret
  map:
    - bwSecretId: 6c230265-d472-45f7-b763-b11b01023ca6
       secretKeyName: test__secret__1
    - bwSecretId: d132a5ed-12bd-49af-9b74-b11b01025d58
       secretKeyName: test__secret__2
  authToken:
    secretName: bw-auth-token
    secretKey: token
EOF
```

In the BitwardenSecret deployment example, the `custom map` element is optional.

Setting	Description
`metadata.name`	The name of the BitwardenSecret object you are deploying.
`spec.organizationId`	The Bitwarden organization ID you are pulling Secrets Manager data from.

Setting	Description
`spec.secretName`	The name of the Kubernetes secret that will be created and injected with Secrets Manager data.
`spec.authToken`	The name of a secret inside of the Kubernetes namespace that the BitwardenSecrets object is being deployed into that contains the Secrets Manager machine account authorization token being used across secrets.

⚠ **Warning**

Secrets Manager does not guarantee unique secret names across projects. By default, secrets will be created with the Secrets Manager secret UUID used as the Key.

To make generated secrets easier to use, you can create a map of Bitwarden Secrets IDs to Kubernetes secret keys. The generated secret will replace the Bitwarden Secret IDs with the mapped name you provide.

Available map settings:

Setting	Description
`bwSecretId`	This is the UUID (universally unique identifier) of the secret in Secrets Manager. This can be found under the secret name in the Secrets Manager web portal or by using the Bitwarden Secrets Manager CLI.
`secretKeyName`	The resulting key inside the Kubernetes secret that replaced the UUID.

Example usage chart

Plain Text

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
  labels:
    app: my-deployment
spec:
  selector:
    matchLabels:
      app: my-deployment
  template:
    metadata:
      labels:
        app: my-deployment
    spec:
      containers:
      - name: my-deployment
        image: <some-image>
      imagePullSecrets:
      - name: <my-secret-name>
      envFrom:
      - secretRef:
          name: bw-sample-secret
```