bitwarden | Help Center Article

SELF-HOSTING > INSTALL & DEPLOY GUIDES >

OpenShift Deployment

View in the help center:
https://bitwarden.com/help/openshift-deployment/

OpenShift Deployment

This article dives into how you might alter your Bitwarden self-hosted Helm Chart deployment based on the specific offerings of OpenShift.

OpenShift routes

This example will demonstrate OpenShift Routes instead of the default ingress controllers.

Disable default ingress

1. Access `my-values.yaml`.

2. Disable the default ingress by specifying `ingress.enabled: false`:

```
Bash

general:
  domain: "replaceme.com"
  ingress:
    enabled: false
```

The remaining ingress values do not require modification, as setting `ingress.enabled: false` will prompt the chart to ignore them.

Add raw manifest for routes

Locate the `rawManifests` section in `my-values.yaml`. This section is where the OpenShift Route manifests will be assigned.

An example file for a `rawManifests` section that uses OpenShift Routes can be downloaded here.

> **Note**
>
> In the example provided above, `destinationCACertificate` has been set to an empty string. This will use the default certificate setup in OpenShift. Alternatively, specify a certificate name here, or you can use Let's Encrypt by following this guide. If you do, you will be required to add `kubernetes.io/tls-acme: "true"` to the annotations for each route.

Shared storage class

A shared storage class is required for most OpenShift deployments. `ReadWriteMany` storage must be enabled. This can be done through the method of your choice, one option is to use the NFS Subdir External Provisioner.

Secrets

The `oc` command can be used to deploy secrets. A valid installation id and key can be retrieved from bitwarden.com/host/. For more information, see What are my installation id and installation key used for?

The following command is an example:

> **Warning**
>
> This example will record commands to your shell history. Other methods may be considered to securely set a secret.

```bash
oc create secret generic custom-secret -n bitwarden \
    --from-literal=globalSettings__installation__id="REPLACE" \
    --from-literal=globalSettings__installation__key="REPLACE" \
    --from-literal=globalSettings__mail__smtp__username="REPLACE" \
    --from-literal=globalSettings__mail__smtp__password="REPLACE" \
    --from-literal=globalSettings__yubico__clientId="REPLACE" \
    --from-literal=globalSettings__yubico__key="REPLACE" \
    --from-literal=globalSettings__hibpApiKey="REPLACE" \
    --from-literal=SA_PASSWORD="REPLACE" # If using SQL pod
    # --from-literal=globalSettings__sqlServer__connectionString="REPLACE" # If using your own SQL server
```

Create a service account

A service account in OpenShift is required as each container needs to run elevated commands on start-up. These commands are blocked by OpenShift's restricted SCCs. We need to create a service account and assign it to the `anyuid` SCC.

1. Run the following commands with the `oc` command line tool:

   ```bash
   oc create sa bitwarden-sa
   oc adm policy add-scc-to-user anyuid -z bitwarden-sa
   ```

2. Next, update `my-values.yaml` to use the new service account. Set the following keys to the name of the service account `bitwarden-sa` that was created in the previous step:

```Bash
component.admin.podServiceAccount
component.api.podServiceAccount
component.attachments.podServiceAccount
component.events.podServiceAccount
component.icons.podServiceAccount
component.identity.podServiceAccount
component.notifications.podServiceAccount
component.scim.podServiceAccount
component.sso.podServiceAccount
component.web.podServiceAccount
database.podServiceAccount
```

Here is an example in the `my-values.yaml` file:

```Bash
component:
  # The Admin component
  admin:
    # Additional deployment labels
    labels: {}
    # Image name, tag, and pull policy
    image:
      name: bitwarden/admin
    resources:
      requests:
        memory: "64Mi"
        cpu: "50m"
      limits:
        memory: "128Mi"
        cpu: "100m"
    securityContext:
    podServiceAccount: bitwarden-sa
```

Note

You can create your own SCC to fine-tune the security of these pods. Managing SCCs in OpenShift describes the out-of-the-box SSCs and how to create your own if desired.