OpenShift implementatie

OpenShift implementatie

Dit artikel gaat in op hoe je je Bitwarden self-hosted Helm Chart-implementatie kunt aanpassen op basis van de specifieke mogelijkheden van OpenShift.

OpenShift routes

Dit voorbeeld demonstreert OpenShift Routes in plaats van de standaard ingress controllers.

Standaard invoer uitschakelen

1. Open `my-values.yaml`.
2. Schakel de standaard ingress uit door `ingress.enabled: false` op te geven:

```
Bash

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

De overige ingress waarden hoeven niet aangepast te worden, omdat het instellen van `ingress.enabled: false` de kaart zal vragen om ze te negeren.

Ruw manifest toevoegen voor routes

Zoek de `rawManifests` sectie in `my-values.yaml`. Hier worden de OpenShift Route manifesten toegewezen.

Een voorbeeldbestand voor een `rawManifests` sectie die OpenShift Routes gebruikt kan worden gedownload ⭳ type: asset-hyperlink id: 33Or6BrWsFLL9FLZbPSLlc.

> **Note**
>
> In het bovenstaande voorbeeld is `destinationCACertificate` ingesteld op een lege tekenreeks. Dit zal de standaard certificaatinstelling in OpenShift gebruiken. Als alternatief kun je hier een certificaatnaam opgeven, of je kunt Let's Encrypt gebruiken door deze handleiding te volgen. Als je dat doet, moet je `kubernetes.io/tls-acme: "true"` toevoegen aan de annotaties voor elke route.

Klasse voor gedeelde opslag

Een shared storage klasse is vereist voor de meeste OpenShift implementaties. `ReadWriteMany` opslag moet ingeschakeld zijn. Dit kan gedaan worden via een methode naar keuze, een optie is om de NFS Subdir External Provisioner te gebruiken.

Geheimen

Het `oc` commando kan gebruikt worden om secrets te deployen. Een geldige installatie-id en -sleutel kunnen worden opgehaald van bitwarden.com/host/. Zie voor meer informatie Waarvoor worden mijn installatie-id en installatiesleutel gebruikt?

Het volgende commando is een voorbeeld:

> ⚠ **Warning**
> Dit voorbeeld zal commando's opnemen in je shell geschiedenis. Er kunnen andere methoden worden overwogen om een geheim veilig in te stellen.

```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
```

Een serviceaccount maken

Een service account in OpenShift is nodig omdat elke container verhoogde commando's moet uitvoeren bij het opstarten. Deze commando's worden geblokkeerd door de beperkte SCC's van OpenShift. We moeten een service account aanmaken en deze toewijzen aan de `anyuid` SCC.

1. Voer de volgende commando's uit met het `oc` commandoregeltool:

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

2. Werk vervolgens `my-values.yaml` bij om de nieuwe serviceaccount te gebruiken. Stel de volgende sleutels in op de naam van de serviceaccount `bitwarden-sa` die in de vorige stap is aangemaakt:

```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
```

Hier is een voorbeeld in het bestand `my-values.yaml`:

```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

U kunt uw eigen SCC maken om de beveiliging van deze pods te verfijnen. SCC's beheren in OpenShift beschrijft de kant-en-klare SSC's en hoe u desgewenst uw eigen SCC's kunt maken.