OpenShift-Bereitstellung

Ansicht im Hilfezentrum:
https://bitwarden.com/help/openshift-deployment/

OpenShift-Bereitstellung

Dieser Artikel geht darauf ein, wie Sie Ihre selbst gehostete Bitwarden Helm Chart Bereitstellung basierend auf den spezifischen Angeboten von OpenShift ändern könnten.

OpenShift-Routen

Dieses Beispiel wird OpenShift Routes anstelle der standardmäßigen Ingress-Controller demonstrieren.

Deaktivieren Sie den Standard-Zugang

1. Zugriff auf `my-values.yaml`.
2. Deaktivieren Sie den Standard-Ingress, indem Sie `ingress.enabled: false` angeben:

```
Bash

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

Die verbleibenden Ingress-Werte benötigen keine Änderung, da die Einstellung `ingress.enabled: false` das Diagramm dazu auffordert, sie zu ignorieren.

Fügen Sie rohe Manifeste für Routen hinzu

Suchen Sie den Abschnitt `rawManifests` in `my-values.yaml`. Dieser Abschnitt ist der Ort, an dem die OpenShift Route-Manifeste zugewiesen werden.

Ein Beispiel für eine Datei für einen Abschnitt `rawManifests`, der OpenShift Routes verwendet, kann heruntergeladen werden ⤓ Typ: Asset-Hyperlink-ID: 33Or6BrWsFLL9FLZbPSLIc.

> **ⓘ Note**
>
> Im oben genannten Beispiel wurde `destinationCACertificate` auf einen leeren String gesetzt. Dies wird das Standardzertifikat-Setup in OpenShift verwenden. Alternativ können Sie hier einen Zertifikatsnamen angeben, oder Sie können Let's Encrypt verwenden, indem Sie dieser Anleitung folgen. Wenn Sie dies tun, müssen Sie `kubernetes.io/tls-acme: "true"` zu den Annotationen für jede Route hinzufügen.

Geteilter Speicherklasse

Eine gemeinsame Speicherklassen ist für die meisten OpenShift-Bereitstellungen erforderlich. `ReadWriteMany` Speicher muss aktiviert sein. Dies kann durch die Methode Ihrer Wahl erfolgen, eine Option ist die Verwendung des NFS Subdir External Provisioner.

Geheimnisse

Der `oc` Befehl kann verwendet werden, um Geheimnisse zu implementieren. Eine gültige Installations-ID und Schlüssel können von bitwarden.com/host/ abgerufen werden. Für weitere Informationen, siehe Wofür werden meine Installations-ID und mein Installations-Schlüssel verwendet?

Der folgende Befehl ist ein Beispiel:

> ⚠ **Warning**
> Dieses Beispiel wird Befehle in Ihrer Shell-Historie aufzeichnen. Andere Methoden können in Betracht gezogen werden, um ein Geheimnis sicher festzulegen.

```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
```

Dienstkonto erstellen

Ein Service-Konto in OpenShift ist erforderlich, da jeder Container beim Start erhöhte Befehle ausführen muss. Diese Befehle werden von OpenShifts eingeschränkten SCCs blockiert. Wir müssen ein Service-Konto erstellen und es der `anyuid` SCC zuweisen.

1. Führen Sie die folgenden Befehle mit dem `oc` Befehlszeilen-Tool aus:

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

2. Aktualisieren Sie als nächstes `my-values.yaml` , um das neue Service-Konto zu verwenden. Setzen Sie die folgenden Schlüssel auf den Namen des Dienstkonto `bitwarden-sa`, das im vorherigen Schritt erstellt wurde:

```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 ist ein Beispiel in der `my-values.yaml` Datei:

```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**
>
> Sie können Ihre eigene SCC erstellen, um die Sicherheit dieser Pods zu optimieren. SCCs in OpenShift verwalten beschreibt die standardmäßigen SSCs und wie man bei Bedarf eigene erstellen kann.