Déploiement OpenShift

Déploiement OpenShift

Cet article se penche sur la manière dont vous pourriez modifier votre déploiement de Bitwarden auto-hébergé Helm Chart en fonction des offres spécifiques d'OpenShift.

Routes OpenShift

Cet exemple démontrera les Routes OpenShift au lieu des contrôleurs d'ingress par défaut.

Désactiver l'ingress par défaut

1. Accédez à `my-values.yaml`.

2. Désactivez l'ingress par défaut en spécifiant `ingress.enabled: false` :

```
Bash

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

Les valeurs d'ingress restantes ne nécessitent pas de modification, car la définition `ingress.enabled: false` incitera le graphique à les ignorer.

Ajouter un manifeste brut pour les itinéraires

Localisez la section `rawManifests` dans `my-values.yaml`. Cette section est où les manifestes de Route OpenShift seront assignés.

Un exemple de fichier pour une section `rawManifests` qui utilise OpenShift Routes peut être téléchargé ⭳ saisir: lien hypertexte d'actif id: 33Or6BrWsFLL9FLZbPSLIc.

> **Note**
>
> Dans l'exemple fourni ci-dessus, `destinationCACertificate` a été défini comme une chaîne de caractères vide. Cela utilisera la configuration de certificat par défaut dans OpenShift. Alternativement, spécifiez ici un nom de certificat, ou vous pouvez utiliser Let's Encrypt en suivant ce guide. Si vous le faites, vous devrez ajouter `kubernetes.io/tls-acme: "true"` aux annotations pour chaque route.

Classe de stockage partagé

Une classe de stockage partagé est requise pour la plupart des déploiements OpenShift. Le stockage `ReadWriteMany` doit être activé. Cela peut être fait par la méthode de votre choix, une option est d'utiliser le Provisionneur Externe de Sous-répertoire NFS.

Secrets

La commande `oc` peut être utilisée pour déployer des secrets. Un identifiant d'installation valide et une clé peuvent être récupérés sur bitwarden.com/host/. Pour plus d'informations, voir À quoi servent mon identifiant d'installation et ma clé d'installation ?

La commande suivante est un exemple :

> ⚠ **Warning**
> Cet exemple enregistrera des commandes dans l'historique de votre shell. D'autres méthodes peuvent être envisagées pour définir un secret de manière sécurisée dans les paramètres.

```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
```

Créez un compte de service

Un compte de service dans OpenShift est nécessaire car chaque conteneur doit exécuter des commandes élevées au démarrage. Ces commandes sont bloquées par les SCC restreints d'OpenShift. Nous devons créer un compte de service et l'attribuer à la SCC `anyuid`.

1. Exécutez les commandes suivantes avec l'outil de ligne de commande `oc` :

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

2. Ensuite, mettez à jour `my-values.yaml` pour utiliser le nouveau compte de service. Définissez les clés suivantes sur le nom du compte de service `bitwarden-sa` qui a été créé à l'étape précédente :

```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
```

Voici un exemple dans le fichier `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

Vous pouvez créer votre propre SCC pour affiner la sécurité de ces pods. Gérer les SCCs dans OpenShift décrit les SCCs prêts à l'emploi et comment créer les vôtres si vous le souhaitez.