bitwarden | Artículo del Centro de Ayuda

SELF-HOSTING > INSTALAR & DESPLEGAR GUÍAS >

Autoaloja con Helm

Ver en el centro de ayuda:
https://bitwarden.com/help/self-host-with-helm/

Autoaloja con Helm

Este artículo le guiará a través del procedimiento para instalar y desplegar Bitwarden en diferentes despliegues de Kubernetes utilizando un gráfico de Helm.

Este artículo describirá los pasos genéricos para alojar Bitwarden en Kubernetes. Guías específicas del proveedor están disponibles para profundizar en cómo podrías alterar una implementación basada en las ofertas específicas de cada proveedor:

- Despliegue de Azure AKS
- Despliegue de OpenShift
- Despliegue de AWS EKS

Requisitos

Antes de proceder con la instalación, asegúrese de que se cumplan los siguientes requisitos:

- kubectl está instalado.
- Helm 3 está instalado.
- Tienes un certificado SSL y una clave o acceso para crear uno a través de un proveedor de certificados.
- Tienes un servidor SMTP o acceso a un proveedor de SMTP en la nube.
- Una clase de almacenamiento que admite ReadWriteMany.
- Tienes una identificación de instalación y una clave obtenida de https://bitwarden.com/host.

Prepara el gráfico

Agrega el repositorio a Helm

Agrega el repositorio a Helm usando los siguientes comandos:

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

Crear un espacio de nombres

Crea un espacio de nombres para desplegar Bitwarden. Nuestra documentación asume un espacio de nombres llamado `Bitwarden`, así que asegúrate de modificar los comandos si eliges un nombre diferente.

```bash
kubectl create namespace bitwarden
```

Crear una configuración

Crea un archivo de configuración `my-values.yaml`, que utilizarás para personalizar tu implementación, utilizando el siguiente comando:

```bash
helm show values bitwarden/self-host > my-values.yaml
```

Como mínimo, debes configurar los siguientes valores en tu archivo `my-values.yaml`:

Valor	Descripción
`dominio.general:`	El dominio que apuntará a la dirección IP pública de tu clúster.
`general.ingress.habilitado:`	Si usar el controlador de ingreso nginx definido en el gráfico (ver un ejemplo usando un controlador de ingreso no incluido).
`general.ingreso.nombreClase:`	Por ejemplo, `"nginx"` o `"azure-application-gateway"` (ver un ejemplo). Establezca `general.ingress.enabled: false` para usar otros controladores de ingreso.
`general.ingress.anotaciones:`	Anotaciones para agregar al controlador de ingreso. Si está utilizando el controlador nginx incluido, se proporcionan valores predeterminados que debe descomentar y personalizar según sea necesario.
`general.ingreso.caminos:`	Si estás utilizando el controlador nginx predeterminado, se proporcionan valores predeterminados que puedes personalizar según sea necesario.
`general.ingress.cert.tls.nombre:`	El nombre de tu certificado TLS. Caminaremos a través de un ejemplo más tarde, así que ingrésalo ahora si lo tienes o vuelve más tarde.
`general.ingress.cert.tls.clusterIssuer:`	El nombre de su emisor de certificado TLS. Caminaremos a través de un ejemplo más tarde, así que ingrésalo ahora si lo tienes o vuelve más tarde.
`general.email.responderAlCorreo:`	Dirección de correo electrónico utilizada para invitaciones, típicamente `no_reply@smtp_host`.

Valor	Descripción
`general.email.smtpHost:`	El nombre de host o la dirección IP de su servidor SMTP.
`general.email.smtpPort:`	El puerto SMTP utilizado por el servidor SMTP.
`general.email.smtpSsl:`	Si su servidor SMTP utiliza un protocolo de cifrado (`verdadero` = SSL, `falso` = TLS).
`habilitarComunicacionEnLaNube:`	Establezca en `verdadero` para permitir la comunicación entre su servidor y nuestro sistema en la nube. Hacerlo permite la sincronización de facturación y licencia.
`región de nube`	Por defecto, `EE.UU.`. Establezca en `EU` si su organización fue iniciada a través del servidor en la nube de la UE.
`sharedStorageClassName:`	El nombre de la clase de almacenamiento compartido, que deberá proporcionar y debe admitir ReadWriteMany (vea un ejemplo usando Azure File Storage) a menos que sea un clúster de un solo nodo.
`secrets.nombreSecreto:`	El nombre de tu objeto secreto de Kubernetes. Crearás este objeto en el próximo paso, así que decide un nombre ahora o vuelve a este valor más tarde.
`base de datos habilitada:`	Si usar o no el pod SQL incluido en el gráfico. Solo establezca en `falso` si está utilizando un servidor SQL externo.
`componente.scim.habilitado`	El pod SCIM está desactivado por defecto. Para habilitar el pod SCIM, establece el valor `= verdadero`.
`componente.volumen.registros.habilitado:`	Aunque no es necesario, recomendamos ajustar a `verdadero` para fines de resolución de problemas.

Crea un objeto secreto

Crea un objeto secreto de Kubernetes para establecer, como mínimo, los siguientes valores:

Valor	Descripción
`configuraciónGlobal__instalación__id`	Una id de instalación válida obtenida de https://bitwarden.com/host. Para obtener más información, consulte ¿Para qué se utilizan mi ID de instalación y mi clave de instalación?
`configuraciónGlobal__instalación__clave`	Una clave de instalación válida obtenida de https://bitwarden.com/host. Para obtener más información, consulte ¿Para qué se utilizan mi ID de instalación y mi clave de instalación?
`configuracionesGlobales__correo__smtp__nombreDeUsuario`	Un nombre de usuario válido para su servidor SMTP.
`configuracionesGlobales__correo__smtp__contraseña`	Una contraseña válida para el nombre de usuario ingresado del servidor SMTP.
`globalSettings__yubico__IdCliente`	ID de cliente para el Servicio de Validación YubiCloud o el Servidor de Validación Yubico autoalojado. Si es YubiCloud, obtenga su ID de cliente y clave secreta aquí.
`configuraciónGlobal__yubico__clave`	Clave secreta para el Servicio de Validación YubiCloud o el Servidor de Validación Yubico autoalojado. Si es YubiCloud, obtenga su ID de cliente y clave secreta aquí.
`globalSettings__hibpApiKey`	Tu clave API de HaveIBeenPwned (HIBP), disponible aquí. Esta clave permite a los usuarios ejecutar el informe de filtración de datos y verificar su contraseña maestra para presencia en filtraciones cuando crean una cuenta.
Si estás utilizando el pod SQL de Bitwarden, `SA_PASSWORD` Si estás utilizando tu propio servidor SQL, `globalSettings__sqlServer__connectionString`	Credenciales para la base de datos conectada a tu instancia de Bitwarden. Lo que se requiere dependerá de si está utilizando el pod SQL incluido o un servidor SQL externo.

Por ejemplo, usar el comando `kubectl create secret` para establecer estos valores se vería de la siguiente manera:

> ⚠ **Warning**
> Este ejemplo registrará comandos en el historial de tu terminal. Otros métodos pueden ser considerados para ajustar de manera segura un secreto.

```Bash
kubectl 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"
```

No olvides establecer el valor de `secrets.secretName:` en `my-values.yaml` al nombre del secreto creado, en este caso `custom-secret`.

Ejemplo de configuración de certificado

El despliegue requiere un certificado y clave TLS, o acceso a la creación de uno a través de un proveedor de certificados. El siguiente ejemplo te guiará a través del uso de cert-manager para generar un certificado con Let's Encrypt:

1. Instale cert-manager en el clúster utilizando el siguiente comando:

   ```Bash
   kubectl apply -f https://github.com/cert-manager/cert-manager/releases/download/v1.11.0/cert-manager.yaml
   ```

2. Definir un emisor de certificados. Bitwarden recomienda usar la configuración de **Staging** en este ejemplo hasta que tus registros DNS hayan sido apuntados a tu clúster. Asegúrate de reemplazar el marcador de posición `correo electrónico:` con un valor válido:

⇒Escenificación

```bash
cat <<EOF | kubectl apply -n bitwarden -f -
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt-staging
spec:
  acme:
    server: https://acme-staging-v02.api.letsencrypt.org/directory
    email: me@example.com
    privateKeySecretRef:
      name: tls-secret
    solvers:
      - http01:
          ingress:
            class: nginx #use "azure/application-gateway" for Application Gateway ingress
EOF
```

⇒Producción

```
Bash

cat <<EOF | kubectl apply -n bitwarden -f -
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt-production
spec:
  acme:
    server: https://acme-v02.api.letsencrypt.org/directory
    email: me@example.com
    privateKeySecretRef:
      name: tls-secret
    solvers:
      - http01:
          ingress:
            class: nginx #use "azure/application-gateway" for Application Gateway ingress
EOF
```

3. Si aún no lo has hecho, asegúrate de establecer los valores `general.ingress.cert.tls.name:` y `general.ingress.cert.tls.clusterIssuer:` en `my-values.yaml`. En este ejemplo, ajustarías:
   - `general.ingress.cert.tls.name: tls-secret`
   - `general.ingress.cert.tls.clusterIssuer: letsencrypt-staging`

Agregando archivos rawManifest

El gráfico de Helm autoalojado de Bitwarden te permite incluir otros archivos de manifiesto de Kubernetes ya sea antes o después de la instalación. Para hacer esto, actualiza la sección `rawManifests` del gráfico (aprende más). Esto es útil, por ejemplo, en escenarios donde quieres usar un controlador de ingreso distinto al controlador nginx definido por defecto.

Instala el gráfico

Para instalar Bitwarden con la configuración establecida en `my-values.yaml`, ejecute el siguiente comando:

```
Bash

helm upgrade bitwarden bitwarden/self-host --install --namespace bitwarden --values my-values.yaml
```

¡Felicidades! Bitwarden ahora está en funcionamiento en `https://your.domain.com`, según se define en `my-values.yaml`. Visita la caja fuerte web en tu navegador web para confirmar que está funcionando. Ahora puedes registrar una nueva cuenta e iniciar sesión.

Necesitará haber configurado una configuración SMTP y secretos relacionados para verificar el correo electrónico de su nueva cuenta.

Próximos pasos

Respaldo y restauración de base de datos

En este repositorio, hemos proporcionado dos trabajos de ejemplo ilustrativos para hacer copias de seguridad y restaurar la base de datos en el pod de la base de datos de Bitwarden. Si está utilizando su propia instancia de SQL Server que no se implementó como parte de este gráfico de Helm, siga las políticas de respaldo y restauración de su empresa.

Las copias de seguridad de la base de datos y las políticas de copia de seguridad dependen en última instancia del implementador. La copia de seguridad podría programarse fuera del clúster para ejecutarse a intervalos regulares, o podría modificarse para crear un objeto CronJob dentro de Kubernetes con fines de programación.

El trabajo de respaldo creará versiones con marca de tiempo de los respaldos anteriores. La copia de seguridad actual simplemente se llama `vault.bak`. Estos archivos se colocan en el volumen persistente de copias de seguridad de MS SQL. El trabajo de restauración buscará `vault.bak` en el mismo volumen persistente.