Självvärd hos Helm

Självvärd hos Helm

Den här artikeln leder dig genom proceduren för att installera och distribuera Bitwarden i olika Kubernetes-distributioner med hjälp av ett Helm-diagram.

Den här artikeln kommer att beskriva de allmänna stegen för att vara värd för Bitwarden på Kubernetes. Leverantörsspecifika guider finns tillgängliga för att fördjupa dig i hur du kan ändra en distribution baserat på varje leverantörs specifika erbjudanden:

- Azure AKS-distribution
- OpenShift-distribution
- AWS EKS-distribution

Krav

Innan du fortsätter med installationen, se till att följande krav är uppfyllda:

- kubectl är installerat.
- Helm 3 är installerad.
- Du har ett SSL-certifikat och nyckel eller tillgång till att skapa ett via en certifikatleverantör.
- Du har en SMTP-server eller tillgång till en moln-SMTP-leverantör.
- En lagringsklass som stöder ReadWriteMany.
- Du har ett installations-id och nyckel hämtade från https://bitwarden.com/host.

Förbered diagrammet

Lägg till repet till Helm

Lägg till repet till Helm med följande kommandon:

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

Skapa ett namnutrymme

Skapa ett namnområde att distribuera Bitwarden till. Vår dokumentation förutsätter ett namnområde som kallas `bitwarden`, så se till att ändra kommandon om du väljer ett annat namn.

```Bash
kubectl create namespace bitwarden
```

Skapa en konfiguration

Skapa en `my-values.yaml-konfigurationsfil` som du använder för att anpassa din distribution med följande kommando:

```
Bash

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

Som ett minimum måste du konfigurera följande värden i filen `my-values.yaml`:

Värde	Beskrivning
`general.domain:`	Domänen som kommer att peka på ditt kluster offentliga IP-adress.
`general.ingress.enabled:`	Om du ska använda nginx-ingångskontrollern som definieras i diagrammet (se ett exempel med en icke-inkluderad ingångskontroller).
`general.ingress.className:`	Till exempel `"nginx"` eller `"azure-application-gateway"` (se ett exempel). Ställ in `general.ingress.enabled: false` för att använda andra ingresskontroller.
`general.ingress.annotations:`	Anteckningar att lägga till i ingångskontrollern. Om du använder den medföljande nginx-kontrollern finns standardinställningar som du måste avkommentera och kan anpassa efter behov.
`general.ingress.paths:`	Om du använder standard nginx-kontrollern finns standardinställningar som du kan anpassa efter behov.
`general.ingress.tls.name:`	Namnet på ditt TLS-certifikat. Vi kommer att gå igenom ett exempel senare, så skriv in det nu om du har det eller ring tillbaka senare.
`general.ingress.tls.clusterIssuer:`	Namnet på din TLS-certifikatutfärdare. Vi kommer att gå igenom ett exempel senare, så skriv in det nu om du har det eller ring tillbaka senare.
`general.email.replyToEmail:`	E-postadress som används för inbjudningar, vanligtvis `no_reply@smtp_host`.

Värde	Beskrivning
`general.email.smtpHost:`	Din SMTP-server värdnamn eller IP-adress.
`general.email.smtpPort:`	SMTP-porten som används av SMTP-servern.
`general.email.smtpSsl:`	Om din SMTP-server använder ett krypteringsprotokoll (`true` = SSL, `false` = TLS).
`aktivera CloudCommunication:`	Ställ in på `sant` för att tillåta kommunikation mellan din server och vårt molnsystem. Om du gör det möjliggörs fakturering och licenssynkronisering.
`cloudRegion:`	Som standard, US. Ställ in på EU om din organisation startades via EU:s molnserver.
`sharedStorageClassName:`	Namnet på den delade lagringsklassen, som du måste tillhandahålla och måste stödja ReadWriteMany (se ett exempel som använder Azure File Storage) såvida det inte är ett ennodkluster.
`secrets.secretName:`	Namnet på ditt Kubernetes hemliga objekt. Du kommer att skapa det här objektet i nästa steg, så bestäm ett namn nu eller ring tillbaka till detta värde.
`database.enabled:`	Om SQL-podden som ingår i diagrammet ska användas. Endast inställt på `false` om du använder en extern SQL-server.
`component.scim.enabled`	SCIM-podden är inaktiverad som standard. För att aktivera SCIM-podden, ställ in värde `= sant`.
`volume.logs.enabled:`	Även om det inte krävs, rekommenderar vi att du ställer in till `sant` för felsökningsändamål.

Skapa ett hemligt objekt

Skapa ett Kubernetes-hemligt objekt för att som ett minimum ställa in följande värden:

Värde	Beskrivning
`globalSettings__installation__id`	Ett giltigt installations-ID hämtat från https://bitwarden.com/host. För mer information, se vad används mitt installations-id och installationsnyckel till?
`globalSettings__installation__key`	En giltig installationsnyckel hämtad från https://bitwarden.com/host. För mer information, se vad används mitt installations-id och installationsnyckel till?
`globalSettings__mail__smtp__användarnamn`	Ett giltigt användarnamn för din SMTP-server.
`globalSettings__mail__smtp__lösenord`	Ett giltigt lösenord för det angivna SMTP-serverns användarnamn.
`globalSettings__yubico__clientId`	Klient-ID för YubiCloud Validation Service eller egenvärd Yubico Validation Server. Om YubiCloud, hämta ditt klient-ID och hemliga nyckel här.
`globalSettings__yubico__key`	Hemlig nyckel för YubiCloud Validation Service eller egenvärd Yubico Validation Server. Om YubiCloud, hämta ditt klient-ID och hemliga nyckel här.
`globalSettings__hibpApiKey`	Din HaveIBeenPwned (HIBP) API-nyckel, tillgänglig här. Den här nyckeln gör det möjligt för användare att köra dataintrångsrapporten och kontrollera sitt huvudlösenord för att se om det finns intrång när de skapar ett konto.
Om du använder Bitwarden SQL-podden, `SA_PASSWORD` Om du använder din egen SQL-server, `globalSettings__sqlServer__connectionString`	Autentiseringsuppgifter för databasen som är ansluten till din Bitwarden-instans. Vad som krävs beror på om du använder den medföljande SQL-podden eller en extern SQL-server.

Om du till exempel använder kommandot `kubectl create secret` för att ställa in dessa värden skulle se ut så här:

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

```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"
```

Glöm inte att ställa in värdet `secrets.secretName`: i `my-values.yaml` till namnet på den skapade hemligheten, i det här fallet `custom-secret`.

Exempel på certifikatinställning

Implementering kräver ett TLS-certifikat och nyckel, eller tillgång till ett skapande via certifikatleverantör. Följande exempel går igenom hur du använder cert-manager för att generera ett certifikat med Let's Encrypt:

1. Installera cert-manager på klustret med följande kommando:

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

2. Definiera en certifikatutfärdare. Bitwarden rekommenderar att du använder Staging-konfigurationen i det här exemplet tills dina DNS-poster har pekats mot ditt kluster. Se till att ersätta platshållaren för `e-post:` med ett giltigt värde:

⇒Staging

```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
```

⇒Production

```
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. Om du inte redan har gjort det, se till att ställa in värdena `general.ingress.cert.tls.name`: och `general.ingress.cert.tls.clusterIssuer`: i `my-values.yaml`. I det här exemplet skulle du ställa in:
   - `general.ingress.cert.tls.name: tls-secret`
   - `general.ingress.cert.tls.clusterIssuer: letsencrypt-staging`

Lägger till rawManifest-filer

Bitwarden självvärd Helm Chart låter dig inkludera andra Kubernetes-manifestfiler antingen före eller efter installation. För att göra detta uppdaterar du avsnittet `rawManifests` i diagrammet (läs mer). Detta är användbart, till exempel i scenarier där du vill använda en annan ingångskontroller än den nginx-kontroller som är definierad som standard.

Installera diagrammet

För att installera Bitwarden med konfigurationsinställningarna i `my-values.yaml`, kör följande kommando:

```
Bash

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

Grattis! Bitwarden är nu igång `på https://din.domän.com`, enligt definitionen i `my-values.yaml`. Besök webbvalvet i din webbläsare för att bekräfta att det fungerar. Du kan nu registrera ett nytt konto och logga in.

Du måste ha konfigurerat en SMTP-konfiguration och relaterade hemligheter för att verifiera e-postadressen för ditt nya konto.

Nästa steg

Databassäkerhetskopiering och återställning

I det här arkivet har vi tillhandahållit två illustrativa exempeljobb för att säkerhetskopiera och återställa databasen i Bitwardens databaspod. Om du använder din egen SQL Server-instans som inte distribueras som en del av det här Helm-diagrammet, vänligen följ företagets policy för säkerhetskopiering och återställning.

Databassäkerhetskopiering och säkerhetskopieringspolicy är i slutändan upp till implementeraren. Säkerhetskopieringen kan schemaläggas utanför klustret för att köras med ett regelbundet intervall, eller så kan den modifieras för att skapa ett CronJob-objekt i Kubernetes för schemaläggningsändamål.

Säkerhetskopieringsjobbet kommer att skapa tidsstämplade versioner av tidigare säkerhetskopior. Den nuvarande säkerhetskopian kallas helt enkelt `vault.bak`. Dessa filer placeras i MS SQL backups beständig volym. Återställningsjobbet kommer att leta efter `vault.bak` i samma beständiga volym.