

SELF-HOSTING > INSTALLATIE- EN IMPLEMENTATIEHANDLEIDINGEN >

Unified Deployment beta

Weergeven in het Helpcentrum:

<https://bitwarden.com/help/install-and-deploy-unified-beta/>

Unified Deployment beta

Note

Deze oplossing is in bèta en bedoeld voor persoonlijk gebruik. Bedrijfsplannen moeten de officieel ondersteunde, standaard implementatieoptie gebruiken.

Terwijl de Bitwarden unified self-hosted implementatie in bèta is, **moeten** degenen die unified installeren geen automatische upgradeprocedures instellen die de nieuwste images ophalen. Bitwarden raadt aan om enige tijd de tijd te nemen om een release te stabiliseren voordat u een upgrade uitvoert.

[Leer hoe u problemen kunt melden.](#)

Dit artikel helpt je bij het installeren en starten van de Bitwarden unified self-hosted deployment. Gebruik deze implementatiemethode om:

- Vereenvoudig de configuratie en optimaliseer het gebruik van bronnen (CPU, geheugen) door Bitwarden te implementeren met een enkel Docker-image.
- Verschillende databaseoplossingen gebruiken, zoals MSSQL, PostgreSQL, SQLite en MySQL/MariaDB.
- Draaien op ARM-architectuur voor alternatieve systemen zoals Raspberry Pi en NAS-servers.

Systemeisen

Bitwarden unified deployment vereist:

- Minimaal 200 MB RAM
- Opslag 1GB
- Docker 19+

Docker installeren

De unified deployment draait op jouw machine met behulp van een [Docker-container](#). De unified deployment kan worden uitgevoerd met elke Docker-editie of -plan. Evalueer welke editie het beste is voor jouw installatie.

Installeer Docker op je machine voordat je verder gaat met de installatie. Raadpleeg de volgende Docker-documentatie voor hulp:

- [Docker installeren](#)

Run Bitwarden verenigd

De unified deployment kan worden uitgevoerd met het commando `docker run` (zie [hier](#)) of met Docker Compose (zie [hier](#)). In beide gevallen moet je omgevingsvariabelen opgeven voor de container.

Omgevingsvariabelen opgeven

Voor het uitvoeren van de unified deployment moeten omgevingsvariabelen worden ingesteld voor de container. Omgevingsvariabelen kunnen worden opgegeven door een `settings.env` bestand aan te maken, waarvan je een voorbeeld kunt vinden in onze [GitHub repository](#), of door de `--env` vlag te gebruiken als je de `docker run` methode gebruikt. Er zijn verschillende optionele variabelen beschikbaar voor een meer gepersonaliseerde unified deployment ervaring. Meer informatie over deze variabelen vind je [hier](#).

Stel minimaal waarden in voor de variabelen die vallen onder de sectie `# Vereiste instellingen #` van het voorbeeld `.env` bestand:

Variabele	Beschrijving
BW_DOMAIN	Vervang <code>bitwarden.uwdomein.com</code> door het domein waar Bitwarden toegankelijk is.
BW_DB_PROVIDER	De databaseprovider die u gaat gebruiken voor uw Bitwarden-server. Beschikbare opties zijn <code>sqlserver</code> , <code>postgres</code> , <code>sqlite</code> of <code>mysql / mariadb</code> .
BW_DB_SERVER	De naam van de server waarop je database draait.
BW_DB_DATABASE	De naam van uw Bitwarden-database.
BW_DB_USERNAME	De gebruikersnaam voor toegang tot de Bitwarden-database.
BW_DB_PASWOORD	Het wachtwoord voor toegang tot de Bitwarden-database.
BW_DB_BESTAND	Alleen vereist voor <code>sqlite</code> als je het pad naar je databasebestand wilt opgeven. Indien niet opgegeven, zal <code>sqlite</code> automatisch een <code>vault.db</code> bestand aanmaken onder het <code>/etc/bitwarden</code> volume.
BW_INSTALLATIE_ID	Een geldige installatie-ID gegenereerd via https://bitwarden.com/host/ .
BW_INSTALLATIE_SLEUTEL	Een geldige installatiesleutel gegenereerd op https://bitwarden.com/host/ .

Note

In tegenstelling tot de standaard implementatie van Bitwarden, wordt unified deployment niet standaard geleverd met een database. Je kunt een bestaande database gebruiken of een nieuwe aanmaken zoals in [dit voorbeeld](#) wordt gedocumenteerd, en in beide gevallen moet je geldige informatie invoeren in de `BW_DB_...` variabelen die hier worden gedocumenteerd.

Het gebruik van niet-MSSQL databaseproviders kan resulteren in prestatieproblemen, omdat er gedurende de bèta nog gewerkt wordt aan ondersteuning voor deze platforms. Gebruik [dit issue-sjabloon](#) om alles te melden met betrekking tot uw Bitwarden unified implementatie en bekijk [deze pagina](#) om bekende problemen te volgen of deel te nemen aan de discussie.

Gebruik docker run

De unified deployment kan worden uitgevoerd met het commando `docker run`, zoals in het volgende voorbeeld:

Bash

```
docker run -d --name bitwarden -v /$(pwd)/bwdata:/etc/bitwarden -p 80:8080 --env-file settings.env bitwarden/self-host:beta
```

Het bovenstaande commando heeft verschillende vereiste opties voor het `docker run` commando, waaronder:

Naam, steno	Beschrijving
<code>--detach, -d</code>	De container op de achtergrond uitvoeren en container-ID afdrukken.
<code>-naam</code>	Geef een naam op voor de container. <code>bitwarden</code> wordt gebruikt in het voorbeeld.
<code>--volume, -v</code>	Een volume binden. Koppel op zijn minst <code>/etc/bitwarden</code> aan.
<code>--publiceer, -p</code>	Breng containerpoorten in kaart voor de host. In het voorbeeld wordt poort <code>80:8080</code> toegewezen. Poort 8443 is vereist bij het configureren van SSL.
<code>-bestand</code>	Pad van het bestand om omgevingsvariabelen uit te lezen. Als alternatief kun je de <code>--env</code> vlag gebruiken om omgevingsvariabelen inline te declareren(meer informatie).

Controleer na het uitvoeren van het commando of de container draait en gezond is met:

Bash

```
docker ps
```

Gefeliciteerd! Je uniforme implementatie is nu operationeel op <https://your.domain.com>. Bezoek de webkluis in je browser om te controleren of deze werkt. Je kunt nu een nieuwe account registreren en inloggen.

Docker Compose gebruiken

Voor het uitvoeren van de unified deployment met Docker Compose is Docker Compose versie 1.24+ nodig. Om de unified deployment uit te voeren met Docker compose, maak je bijvoorbeeld een `docker-compose.yml` bestand:

Bash

```
---
version: "3.8"

services:
  bitwarden:
    depends_on:
      - db
    env_file:
      - settings.env
    image: bitwarden/self-host:beta
    restart: always
    ports:
      - "80:8080"
    volumes:
      - bitwarden:/etc/bitwarden

  db:
    environment:
      MARIADB_USER: "bitwarden"
      MARIADB_PASSWORD: "super_strong_password"
      MARIADB_DATABASE: "bitwarden_vault"
      MARIADB_RANDOM_ROOT_PASSWORD: "true"
    image: mariadb:10
    restart: always
    volumes:
      - data:/var/lib/mysql

volumes:
  bitwarden:
  data:
```

Maak in het `docker-compose.yml` bestand alle gewenste configuraties, inclusief:

- Volumes in kaart brengen voor logs en Bitwarden-gegevens.
- Poorten in kaart brengen.

- Een databaseafbeelding configureren. ^a

^aStel **alleen** een database in `docker-compose.yml` in, zoals in het bovenstaande voorbeeld, als je **een nieuwe databaseserver wilt maken** voor gebruik met Bitwarden. Voorbeeldconfiguraties voor MySQL, MSSQL en PostgreSQL zijn opgenomen in ons [voorbeeldbestand](#).

Zodra je `docker-compose.yml` en `settings.env` bestand zijn aangemaakt, start je je unified server door het volgende uit te voeren:

Bash

```
docker compose up -d
```

Controleer of alle containers correct draaien:

Bash

```
docker ps
```

Gefeliciteerd! Je uniforme implementatie is nu operationeel op <https://your.domain.com>. Bezoek de webkuis in je browser om te controleren of deze werkt. Je kunt nu een nieuwe account registreren en inloggen.

Werk uw server bij

Om je unified deployment bij te werken:

⇒ Docker uitvoeren update

1. Stop de Docker-container:

Bash

```
docker stop bitwarden
```

2. Verwijder de Docker-container:

Bash

```
docker rm bitwarden
```

3. Voer de volgende opdracht uit om de meest recente Bitwarden unified image op te halen:

Bash

```
docker pull bitwarden/self-host:beta
```

4. Voer de Docker-container opnieuw uit:

Bash

```
docker run -d --name bitwarden -v /$(pwd)/bwdata:/etc/bitwarden -p 80:8080 --env-file settings.  
env bitwarden/self-host:beta
```

⇒ Docker Compose update

1. Stop de Docker-container:

Bash

```
docker compose down
```

2. Voer de volgende opdracht uit om de meest recente Bitwarden unified image op te halen:

Bash

```
docker compose pull
```

3. Maak alle containers die moeten worden bijgewerkt opnieuw aan:

Bash

```
docker compose up -d
```

4. Controleer of de containers draaien:

Bash

```
docker compose ps
```

Omgevingsvariabelen

De uniforme implementatie werkt standaard zonder een aantal van de standaard Bitwarden-services. Dit zorgt voor meer maatwerk en optimalisatie van uw uniforme implementatie. Configureer deze services en meer optionele instellingen door verschillende omgevingsvariabelen aan te passen.

Note

Wanneer je een omgevingsvariabele wijzigt, moet de Docker-container opnieuw worden aangemaakt. Lees [hier](#) meer.

Webserver poorten

Variabele	Beschrijving
BW_PORT_HTTP	Wijzig de poort die wordt gebruikt voor HTTP-verkeer. Standaard 8080 .
BW_PORT_HTTPS	Wijzig de poort die wordt gebruikt voor HTTPS-verkeer. Standaard 8443 .

SSL

Gebruik deze waarden om de certificaatinstellingen te wijzigen.

Variabele	Beschrijving
BW_ENABLE_SSL	Gebruik SSL/TLS. waar onwaar . Standaard vals . SSL is nodig om Bitwarden goed te laten functioneren. Als u geen SSL configureert in de Bitwarden-container, moet u Bitwarden voorzien van een SSL-proxy.
BW_SSL_CERT	De naam van uw SSL-certificaatbestand. Het bestand moet zich in de map /etc/bitwarden in de container bevinden. Standaard ssl.crt .
BW_SSL_KEY	De naam van uw SSL-sleutelbestand. Het bestand moet zich in de map /etc/bitwarden in de container bevinden. Standaard ssl.key .
BW_SCHAKEL_SSL_CA IN	Gebruik SSL met een door een certificeringsautoriteit (CA) ondersteunde service. waar onwaar . Standaard vals .
BW_SSL_CA_CERT	De naam van uw SSL CA certificaat. Het bestand moet zich in de map /etc/bitwarden in de container bevinden. Standaard ca.crt .

Variabele	Beschrijving
BW_INSCHAKELEN_SSL_DH	Gebruik SSL met Diffie-Hellman-sleuteluitwisseling. waar onwaar . Standaard vals .
BW_SSL_DH_CERT	De naam van uw Diffie-Hellman parameterbestand. Het bestand moet zich in de map /etc/bitwarden in de container bevinden. Standaard dh.pem .
BW_SSL_PROTOCOLLEN	SSL-versie gebruikt door NGINX. Leeg laten voor aanbevolen standaard. Meer informatie .
BW_SSL_CIPHERS	SSL-ciphersuites gebruikt door NGINX. Leeg laten voor aanbevolen standaard. Meer informatie .

Note

Als u een bestaand SSL-certificaat gebruikt, moet u de juiste SSL-opties inschakelen in **settings.env**. SSL-bestanden moeten worden opgeslagen in **/etc/bitwarden**, waarnaar kan worden verwezen in het **docker-compose.yml** bestand. Deze bestanden moeten overeenkomen met de namen die zijn geconfigureerd in **settings.env**.

Het standaardgedrag is om een zelfondertekend certificaat te genereren als SSL is ingeschakeld en er geen bestaande certificaatbestanden zijn op de verwachte locatie (**/etc/bitwarden**).

Diensten

Extra diensten kunnen worden in- of uitgeschakeld voor specifieke gebruikssituaties, zoals bedrijfs- of teambehoeften, door de volgende waarden te wijzigen:

Variabele	Beschrijving
BW_INSCHAKELEN_ADMIN	Schakel deze service niet uit. Lees hier meer over de mogelijkheden van het beheerpaneel. Standaard waar .
BW_ENABLE_API	Schakel deze service niet uit. Standaard waar .
BW_INSCHAKELEN_GEBEURTENISSEN	In- of uitschakelen van Bitwarden-gebeurtenislogboeken voor teams en bewaking van bedrijfsgebeurtenissen. Standaard vals .

Variabele	Beschrijving
BW_ENABLE_ICONS	Bitwarden-merkpictogrammen in- of uitschakelen die zijn ingesteld met de URI's van het aanmeldingsitem. Lees hier meer. Standaard waar .
BW_IDENTITEIT INSCHAKELEN	Schakel deze service niet uit. Standaard waar .
BW_MELDINGEN INSCHAKELEN	Meldingsservices in- of uitschakelen voor het ontvangen van pushmeldingen op mobiele apparaten, inloggen met apparaat, mobiele kluisynchronisatie en meer. Standaard waar .
BW_ENABLE_SCIM	SCIM in- of uitschakelen voor Enterprise-organisaties. Standaard vals .
BW_ENABLE_SSO	SSO-services voor Enterprise-organisaties in- of uitschakelen. Standaard vals .
BW_ICONEN_PROXY_NAAR_CLOUD	Door deze service in te schakelen worden verzoeken voor pictogramservices via cloudservices afgehandeld om het systeemgeheugen minder te belasten. Als je ervoor kiest om deze instelling te gebruiken, moet BW_ENABLE_ICONS worden ingesteld op false om de containerbelasting te verminderen. Standaard vals .

Mail

Configureer SMTP-instellingen voor je unified implementatie. Kopieer de informatie van de SMTP-provider van je gekozen e-mail naar de volgende velden:

Variabele	Beschrijving
globalSettings__mail__replyToEmail	Voer de antwoordmail voor uw server in.
globalSettings__mail__smtp__host	Voer het hostdomein voor uw SMTP-server in.
globalSettings__mail__smtp__port	Voer het poortnummer van de SMTP-host in.

Variabele	Beschrijving
<code>algemene instellingen__mail__smtp__ssl</code>	Als je SMTP host SSL gebruikt, voer dan <code>true</code> in. Stel de waarde in op <code>false</code> als uw host de TLS-service gebruikt.
<code>globalSettings__mail__smtp__gebruikersnaam</code>	Voer de SMTP-gebruikersnaam in.
<code>globaalInstellingen__mail__smtp__wachtwoord</code>	Voer het SMTP-wachtwoord in.

Yubico API (YubiKey)

Variabele	Beschrijving
<code>globaleInstellingen__yubico__clientId</code>	Vervang de waarde door de ID die u van uw Yubico Key hebt ontvangen. Meld je hier aan voor Yubico Key.
<code>globale instellingen__yubico__sleutel</code>	Voer de van Yubico ontvangen sleutelwaarde in.

Databaseconfiguraties

Voor het gebruik van de verschillende databaseopties die compatibel zijn met de unified deployment zijn aanvullende `.env` configuraties nodig.

⇒MySQL/MariaDB

In `settings.env`:

Bash

```
# Database
BW_DB_PROVIDER=mysql
BW_DB_SERVER=db
BW_DB_DATABASE=bitwarden_vault
BW_DB_USERNAME=bitwarden
BW_DB_PASSWORD=super_strong_password
```

⇒MSSQL

In `settings.env`:

Bash

```
# Database
BW_DB_PROVIDER=sqlserver
BW_DB_SERVER=db
BW_DB_DATABASE=bitwarden_vault
BW_DB_USERNAME=bitwarden
BW_DB_PASSWORD=super_strong_password
```

⇒SQLite

In `settings.env`:

Bash

```
# Database
BW_DB_PROVIDER=sqlite
BW_DB_FILE=/path/to/.db
```

Het toewijzen van de `sqlite` waarde zal automatisch een `vault.db` bestand aanmaken in het `/etc/bitwarden` volume. `BW_DB_FILE` is alleen nodig als je het pad naar een ander databasebestand wilt opgeven.

⇒PostgreSQL

In `settings.env`:

Bash

```
# Database
BW_DB_PROVIDER=postgresql
BW_DB_SERVER=db
BW_DB_DATABASE=bitwarden_vault
BW_DB_USERNAME=bitwarden
BW_DB_PASSWORD=super_strong_password
```

Andere

Variabele	Beschrijving
globalSettings__disableUserRegistration	De registratiemogelijkheden voor gebruikersaccounts in- of uitschakelen.
globaleInstellingen__hibpApiKey	Voer de API-sleutel in die is geleverd door Have I Been Pwnd. Registreer je hier om de API-sleutel te ontvangen.
adminSettings__admins	Voer e-mailadressen voor beheerders in.
BW_REAL_IPS	Definieer echte IP's in <code>nginx.conf</code> in een door komma's gescheiden lijst. Nuttig voor het definiëren van proxy servers die het IP-adres van de client doorsturen. Meer informatie .
BW_CSP	Parameter inhoud-beveiligingsbeleid. Het herconfigureren van deze parameter kan functies afbreken. Door deze parameter te wijzigen, wordt u verantwoordelijk voor het handhaven van deze waarde.
BW_DB_PORT	Geef een aangepaste poort op voor databaseverkeer. Indien niet gespecificeerd, hangt de standaard af van de gekozen databaseprovider.

De container opnieuw opstarten

Om uw Docker-container opnieuw te starten na het wijzigen van omgevingsvariabelen, voert u de volgende opdrachten uit vanuit de unified deployment-directory van Bitwarden:

⇒ Docker uitvoeren

1. Stop de Docker-container:

```
Bash
```

```
docker stop bitwarden
```

2. Verwijder de Docker-container:

```
Bash
```

```
docker rm bitwarden
```

3. Voer de Docker-container opnieuw uit:

```
Bash
```

```
docker run -d --name bitwarden -v /$(pwd)/bwdata/:/etc/bitwarden -p 80:8080 --env-file settings.env bitwarden/self-host:beta
```

⇒ Docker Compose

1. Stop de Docker-container:

```
Bash
```

```
docker compose down
```

2. Maak de containers opnieuw:

```
Bash
```

```
docker compose up -d
```

3. Controleer of de containers goed draaien met:

```
Bash
```

```
docker compose ps
```

Geheugengebruik

Standaard gebruikt de Bitwarden-container het geheugen dat beschikbaar is, vaak meer dan het minimum dat nodig is om te draaien. Voor geheugenbewuste omgevingen kunt u `docker -m` of `--memory=` gebruiken om het geheugengebruik van de Bitwarden-container te beperken.

Naam, steno	Beschrijving
-geheugen=-, -m	De maximale hoeveelheid geheugen die de container kan gebruiken. Bitwarden heeft minstens 200 m nodig. Raadpleeg de Docker-documentatie voor meer informatie.

Om het geheugengebruik met Docker Compose te controleren, gebruik je de `mem_limit` sleutel:

Bash

```
services:
  bitwarden:
    env_file:
      - settings.env
    image: bitwarden/self-host:beta
    restart: always
    mem_limit: 200m
```

Problemen rapporteren

Terwijl de Bitwarden unified deployment nog in bètaversie is, moedigen we je aan om problemen te melden en feedback te geven via GitHub. Gebruik [dit issue-sjabloon](#) om alles te melden met betrekking tot uw Bitwarden unified implementatie en bekijk [deze pagina](#) om bekende problemen te volgen of deel te nemen aan de discussie.

Extra middelen

- Als u van plan bent om zelf een Bitwarden-organisatie te hosten, zie dan [Zelf een organisatie hosten](#) om te beginnen.

Zie voor meer informatie over de standaard zelf gehoste implementatie van Bitwarden:

- [Installeren en implementeren - Linux](#)
- [Installeren en implementeren - Windows](#)
- [Installeren en implementeren - Handmatig](#)