

SELF-HOSTING > INSTALLER & DÉPLOYER DES GUIDES >

# Déploiement Unifié bêta

Afficher dans le centre d'aide:

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

## Déploiement Unifié bêta

### Note

Cette solution est en bêta et destinée à un usage personnel. Les plans d'affaires devraient utiliser l'option de déploiement standard officiellement soutenue.

Alors que le déploiement unifié auto-hébergé de Bitwarden est en bêta, ceux qui installent unifié **ne devraient pas** mettre en place des procédures de mise à niveau automatiques qui récupèrent les dernières images disponibles. Bitwarden recommande de laisser un peu de temps pour la stabilisation d'une version avant de la mettre à niveau.

[Découvrez comment signaler des problèmes](#) .

Cet article vous guidera à travers l'installation et le lancement du déploiement unifié auto-hébergé de Bitwarden. Utilisez cette méthode de déploiement pour :

- Simplifiez la configuration et optimisez l'utilisation des ressources (CPU, mémoire) en déployant Bitwarden avec une seule image Docker.
- Utilisez différentes solutions de base de données telles que MSSQL, PostgreSQL, SQLite, et MySQL/MariaDB.
- Exécutez sur l'architecture ARM pour des systèmes alternatifs tels que Raspberry Pi et les serveurs NAS.

## Exigences du système

Le déploiement unifié de Bitwarden nécessite :

- Au moins 200 Mo de RAM
- Stockage 1GB
- Moteur Docker 19+

## Installez Docker

Le déploiement unifié fonctionnera sur votre machine en utilisant un [conteneur Docker](#). Le déploiement unifié peut être exécuté avec n'importe quelle édition ou plan Docker. Évaluez quelle édition est la meilleure pour votre installation.

**Installez Docker sur votre machine avant de procéder à l'installation.** Reportez-vous à la documentation Docker suivante pour obtenir de l'aide :

- [Installez Docker Engine](#)

## Exécutez Bitwarden unifié

Le déploiement unifié peut être exécuté en utilisant la commande `docker run` (voir [ici](#)) ou en utilisant Docker Compose (voir [ici](#)). Dans les deux cas, vous devrez spécifier des variables d'environnement pour le conteneur.

## Spécifiez les variables d'environnement

L'exécution du déploiement unifié nécessitera que des variables d'environnement soient définies pour le conteneur. Les variables d'environnement peuvent être spécifiées en créant un fichier `settings.env`, dont vous pouvez trouver un exemple dans notre dépôt [GitHub](#), ou en utilisant le drapeau `--env` si vous utilisez la méthode `docker run`. Plusieurs variables optionnelles sont disponibles pour une expérience de déploiement unifiée plus personnalisée. Des détails supplémentaires sur ces variables peuvent être trouvés [ici](#).

Au minimum, définissez des valeurs pour les variables qui se trouvent sous la section # Paramètres Requis # de l'exemple de fichier .env :

Variable	Description
DOMAINE_BW	Remplacez <code>bitwarden.yourdomain.com</code> par le domaine où Bitwarden sera accessible.
FOURNISSEUR_DB_BW	Le fournisseur de base de données que vous utiliserez pour votre serveur Bitwarden. Les options disponibles sont <code>sqlserver</code> , <code>postgresql</code> , <code>sqlite</code> , ou <code>mysql/mariadb</code> .
SERVEUR_BW_DB	Le nom du serveur sur lequel votre base de données fonctionne.
BW_DB_BASE DE DONNÉES	Le nom de votre base de données Bitwarden.
BW_DB_NOM_UTILISATEUR	Le nom d'utilisateur pour accéder à la base de données Bitwarden.
BW_DB_MOTDEPASSE	Le mot de passe pour accéder à la base de données Bitwarden.
FICHER_BD_BW	Uniquement requis pour <code>sqlite</code> si vous souhaitez spécifier le chemin d'accès à votre fichier de base de données. Si non spécifié, <code>sqlite</code> créera automatiquement un fichier <code>vault.db</code> sous le volume <code>/etc/bitwarden</code> .
ID_INSTALLATION_BW	Un ID d'installation valide généré à partir de <a href="https://bitwarden.com/host/">https://bitwarden.com/host/</a> .
CLÉ_D'INSTALLATION_BW	Une clé d'installation valide générée à partir de <a href="https://bitwarden.com/host/">https://bitwarden.com/host/</a> .

### Note

Contrairement au déploiement standard de Bitwarden, le déploiement unifié n'est pas livré avec une base de données intégrée. Vous pouvez utiliser une base de données existante, ou en créer une nouvelle comme documenté dans [cet exemple](#), et dans les deux cas, vous devez entrer des informations valides dans les variables `BW_DB_...` documentées ici.

L'utilisation de fournisseurs de bases de données autres que MSSQL peut entraîner des problèmes de performance, car le support pour ces plateformes continue d'être travaillé tout au long de la bêta. Veuillez utiliser [ce modèle de problème](#) pour rapporter tout ce qui concerne votre déploiement unifié de Bitwarden et consultez [cette page](#) pour suivre les problèmes connus ou participer à la discussion.

## Utilisation de docker run

Le déploiement unifié peut être exécuté avec la commande `docker run`, comme dans l'exemple suivant :

### Bash

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

La commande présentée ci-dessus comporte plusieurs options requises pour la commande `docker run`, y compris :

Nom, abréviation	Description
<code>--détacher, -d</code>	Exécutez le conteneur en arrière-plan et imprimez l'ID du conteneur.
<code>--nom</code>	Fournissez un nom pour le conteneur. <code>Bitwarden</code> est utilisé dans l'exemple.
<code>--volume, -v</code>	Montez un volume en bind. Au minimum, montez <code>/etc/bitwarden</code> .
<code>--publier, -p</code>	Mapez les ports du conteneur à l'hôte. L'exemple montre le port <code>80:8080</code> cartographié. Le port 8443 est requis lors de la configuration SSL.
<code>--fichier-env</code>	Chemin du fichier à lire pour obtenir les variables d'environnement. Alternativement, utilisez le drapeau <code>--env</code> pour déclarer des variables d'environnement en ligne ( <a href="#">en savoir plus</a> ).

Une fois que vous avez exécuté la commande, vérifiez que le conteneur fonctionne et est en bonne santé avec :

```
Bash
```

```
docker ps
```

Félicitations ! Votre déploiement unifié est maintenant opérationnel à l'adresse <https://your.domain.com>. Visitez le coffre web dans votre navigateur pour confirmer qu'il fonctionne. Vous pouvez maintenant enregistrer un nouveau compte et vous connecter.

### Utilisation de Docker Compose

L'exécution du déploiement unifié avec Docker Compose nécessitera la version 1.24+ de Docker Compose. Pour exécuter le déploiement unifié avec Docker compose, créez un fichier `docker-compose.yml`, par exemple :

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:
```

Dans le fichier `docker-compose.yml`, effectuez toutes les configurations souhaitées, y compris :

- Cartographie des volumes pour les journaux et les données Bitwarden.
- Mappage des ports.

- Configuration d'une image de base de données.<sup>a</sup>

<sup>a</sup>Configurez uniquement une base de données dans `docker-compose.yml`, comme dans l'exemple ci-dessus, si vous souhaitez **créer un nouveau serveur de base de données** à utiliser avec Bitwarden. Des configurations d'exemple pour MySQL, MSSQL et PostgreSQL sont incluses dans notre [fichier d'exemple](#).

Une fois que votre fichier `docker-compose.yml` et `settings.env` sont créés, démarrez votre serveur unifié en exécutant :

*Bash*

```
docker compose up -d
```

Vérifiez que tous les conteneurs fonctionnent correctement :

*Bash*

```
docker ps
```

Félicitations ! Votre déploiement unifié est maintenant opérationnel à l'adresse <https://your.domain.com>. Visitez le coffre web dans votre navigateur pour confirmer qu'il fonctionne. Vous pouvez maintenant enregistrer un nouveau compte et vous connecter.

## Mettez à jour votre serveur

Pour mettre à jour votre déploiement unifié :

### ⇒ Docker exécute mettre à jour

1. Arrêtez le conteneur Docker en cours d'exécution:

*Bash*

```
docker stop bitwarden
```

2. Supprimer le conteneur Docker :

*Bash*

```
docker rm bitwarden
```

3. Exécutez la commande suivante pour récupérer l'image unifiée Bitwarden la plus récente :

*Bash*

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

4. Exécutez à nouveau le conteneur Docker :

*Bash*

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

## ⇒ Mise à jour de Docker Compose

1. Arrêtez le conteneur Docker en cours d'exécution:

*Bash*

```
docker compose down
```

2. Exécutez la commande suivante pour récupérer l'image unifiée Bitwarden la plus récente :

*Bash*

```
docker compose pull
```

3. Recréez tous les conteneurs qui doivent être mis à jour :

*Bash*

```
docker compose up -d
```

4. Vérifiez que les conteneurs sont en fonctionnement :

*Bash*

```
docker compose ps
```



## Variabes d'environnement

Le déploiement unifié fonctionnera par défaut sans plusieurs des services standard de Bitwarden. Cela permet une personnalisation et une optimisation accrues de votre déploiement unifié. Configurez ces services, et plus de paramètres optionnels, en éditant diverses variables d'environnement.

### i Note

Chaque fois que vous modifiez une variable d'environnement, le conteneur Docker devra être recréé. Apprenez-en plus [ici](#).

## Ports de serveur web

Variable	Description
BW_PORT_HTTP	Changez le port utilisé pour le trafic HTTP. Par défaut, <b>8080</b> .
BW_PORT_HTTPS	Changez le port utilisé pour le trafic HTTPS. Par défaut, <b>8443</b> .

## SSL

Utilisez ces valeurs pour modifier les paramètres du certificat.

Variable	Description
BW_ENABLE_SSL	Utilisez SSL/TLS. <b>vrai/faux</b> . Défaut <b>faux</b> . SSL est requis pour que Bitwarden fonctionne correctement. Si vous n'utilisez pas SSL configuré dans le conteneur Bitwarden, vous devriez mettre Bitwarden devant un proxy SSL.
BW_SSL_CERT	Le nom de votre fichier de certificat SSL. Le fichier doit être situé dans le répertoire <b>/etc/bitwarden</b> à l'intérieur du conteneur. Défaut <b>ssl.crt</b> .
CLÉ_BW_SSL	Le nom de votre fichier de clé SSL. Le fichier doit être situé dans le répertoire <b>/etc/bitwarden</b> à l'intérieur du conteneur. Défaut <b>ssl.key</b> .
BW_ENABLE_SSL_CA	Utilisez SSL avec un service soutenu par une autorité de certification (CA). <b>vrai/faux</b> . Défaut <b>faux</b> .

Variable	Description
BW_SSL_CA_CERT	Le nom de votre certificat CA SSL. Le fichier doit être situé dans le répertoire <code>/etc/bitwarden</code> à l'intérieur du conteneur. Défaut <code>ca.crt</code> .
BW_ENABLE_SSL_DH	Utilisez SSL avec l'échange de clés Diffie-Hellman. <code>vrai/faux</code> . Défaut <code>faux</code> .
BW_SSL_DH_CERT	Le nom de votre fichier de paramètres Diffie-Hellman. Le fichier doit être situé dans le répertoire <code>/etc/bitwarden</code> à l'intérieur du conteneur. Défaut <code>dh.pem</code> .
PROTOCOLES_BW_SSL	Version SSL utilisée par NGINX. Laissez vide pour la valeur par défaut recommandée. <a href="#">En savoir plus</a> .
BW_SSL_CIPHERS	Suites de chiffrement SSL utilisées par NGINX. Laissez vide pour la valeur par défaut recommandée. <a href="#">En savoir plus</a> .

### Note

Si vous utilisez un certificat SSL existant, vous devrez activer les options SSL appropriées dans `settings.env`. Les fichiers SSL doivent être stockés dans `/etc/bitwarden`, qui peuvent être référencés dans le fichier `docker-compose.yml`. Ces fichiers doivent correspondre aux noms configurés dans `settings.env`.

Le comportement par défaut est de générer un certificat auto-signé si SSL est activé et qu'aucun fichier de certificat existant ne se trouve à l'emplacement attendu (`/etc/bitwarden`).

## Services

Des services supplémentaires peuvent être activés ou désactivés pour des cas d'utilisation spécifiques, tels que les besoins de l'entreprise ou des équipes, en modifiant les valeurs suivantes :

Variable	Description
ACTIVER_ADMIN	<p><b>Ne désactivez pas ce service.</b></p> <p>En savoir plus sur les capacités du panneau d'admin <a href="#">ici</a>. Défaut <code>vrai</code>.</p>

Variable	Description
ACTIVER_API_BW	<b>Ne désactivez pas ce service.</b> Défaut <b>vrai</b> .
ACTIVER_EVENEMENTS_BW	Activez ou désactivez les journaux d'événements Bitwarden pour le suivi des événements des équipes et de l'entreprise. Défaut <b>faux</b> .
ACTIVER_ICÔNES_BW	Activez ou désactivez les icônes de marque Bitwarden qui sont définies avec les URI de l'élément d'identifiant. Apprenez-en plus <a href="#">ici</a> . Défaut <b>vrai</b> .
BW_ENABLE_IDENTITY	<b>Ne désactivez pas ce service.</b> Défaut <b>vrai</b> .
ACTIVER_LES_NOTIFICATIONS	Activez ou désactivez les services de notification pour recevoir des notifications push sur les appareils mobiles, en utilisant l'identifiant avec l'appareil, la synchronisation du coffre mobile, et plus encore. Défaut <b>vrai</b> .
ACTIVER_SCIM_BW	Activer ou désactiver SCIM pour les organisations d'Entreprise. Défaut <b>faux</b> .
ACTIVER_SSO	Activer ou désactiver les services SSO pour les organisations de l'Entreprise. Défaut <b>faux</b> .
ICÔNES_BW_PROXY_VERS_CLOUD	L'activation de ce service permettra aux demandes de service d'icônes proxy de fonctionner via les services cloud afin de réduire la charge de la mémoire système. Si vous choisissez d'utiliser ce paramètre, <b>BW_ENABLE_ICONS</b> doit être réglé sur <b>faux</b> afin de réduire la charge du conteneur. Défaut <b>faux</b> .

## Courrier

Configurez les paramètres SMTP pour votre déploiement unifié. Copiez les informations de votre fournisseur de messagerie SMTP choisi dans les champs suivants:

Variable	Description
<code>paramètresGlobaux__mail__répondreÀEmail</code>	Entrez le courriel de réponse pour votre serveur.
<code>paramètresGlobaux__mail__smtp__hôte</code>	Entrez le domaine hôte pour votre serveur SMTP.
<code>paramètresGlobaux__mail__smtp__port</code>	Entrez le nombre de port de l'hôte SMTP.
<code>paramètresGlobaux__mail__smtp__ssl</code>	Si votre hôte SMTP utilise SSL, entrez <b>vrai</b> . Définissez la valeur sur <b>false</b> si votre hôte utilise le service TLS.
<code>paramètresGlobaux__mail__smtp__nomUtilisateur</code>	Entrez le nom d'utilisateur SMTP.
<code>paramètresGlobaux__mail__smtp__motDePasse</code>	Entrez le mot de passe SMTP.

### API Yubico (YubiKey)

Variable	Description
<code>paramètresGlobaux__yubico__identifiantClient</code>	Remplacez la valeur par l'ID reçu de votre clé Yubico. Inscrivez-vous pour la clé Yubico <a href="#">ici</a> .
<code>paramètresGlobaux__yubico__clé</code>	Entrez la valeur de clé reçue de Yubico.

### Configurations de base de données

L'utilisation de la variété d'options de base de données compatibles avec le déploiement unifié nécessitera des configurations **.env** supplémentaires.

#### ⇒MySQL/MariaDB

Dans **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

Dans `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

Dans `settings.env`:

### Bash

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

L'attribution de la valeur `sqlite` créera automatiquement un fichier `vault.db` dans le volume `/etc/bitwarden`. `BW_DB_FILE` n'est nécessaire que si vous souhaitez spécifier le chemin vers un autre fichier de base de données.

### ⇒PostgreSQL

Dans `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
```

### Autre

#### Variable

paramètresGlobaux\_\_désactiverEnregistrementUtilisateur

#### Description

Activer ou désactiver les capacités d'enregistrement de compte utilisateur.

paramètresGlobaux\_\_hibpApiKey

Entrez la clé API fournie par Have I Been Pwnd. Inscrivez-vous pour recevoir la clé API [ici](#).

paramètresAdmin\_\_admins

Entrez les adresses de courriel de l'admin.

BW\_REAL\_IPS

Définissez de vraies adresses IP dans `nginx.conf` dans une liste séparée par des virgules. Utile pour définir les serveurs proxy qui transmettent l'adresse IP du client. [En savoir plus](#).

BW\_CSP

Paramètre de la politique de sécurité du contenu. La reconfiguration de ce paramètre peut interrompre des fonctionnalités. En modifiant ce paramètre, vous devenez responsable de la maintenance de cette valeur.

BW\_DB\_PORT

Spécifiez un port personnalisé pour le trafic de base de données. Si non spécifié, la valeur par défaut dépendra de votre fournisseur de base de données choisi.

### Redémarrez le conteneur

Pour redémarrer votre conteneur Docker après avoir modifié les variables d'environnement, exécutez les commandes suivantes à partir du répertoire de déploiement unifié de Bitwarden :

## ⇒ Docker exécuter

1. Arrêtez le conteneur Docker en cours d'exécution:

*Bash*

```
docker stop bitwarden
```

2. Supprimer le conteneur Docker :

*Bash*

```
docker rm bitwarden
```

3. Exécutez à nouveau le conteneur Docker :

*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. Arrêtez le conteneur Docker en cours d'exécution:

*Bash*

```
docker compose down
```

2. Recréez les conteneurs:

*Bash*

```
docker compose up -d
```

3. Assurez-vous que les conteneurs fonctionnent correctement avec :

*Bash*

```
docker compose ps
```

## Utilisation de la mémoire

Par défaut, le conteneur Bitwarden consommera la mémoire qui est disponible pour lui, souvent plus que le minimum nécessaire pour fonctionner. Pour les environnements soucieux de la mémoire, vous pouvez utiliser docker `-m` ou `--memory=` pour limiter l'utilisation de la mémoire du conteneur Bitwarden.

Nom, abréviation	Description
<code>--mémoire=, -m</code>	La quantité maximale de mémoire que le conteneur peut utiliser. Bitwarden nécessite au moins 200m. Voir la <a href="#">documentation Docker</a> pour en savoir plus.

Pour contrôler l'utilisation de la mémoire avec Docker Compose, utilisez la clé `mem_limit` :

### Bash

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

## Signaler des problèmes

Alors que le déploiement unifié de Bitwarden reste en version bêta, nous vous encourageons à signaler les problèmes et à donner votre avis via GitHub. Veuillez utiliser [ce modèle de problème](#) pour signaler tout ce qui concerne votre déploiement unifié de Bitwarden et consultez [cette page](#) pour suivre les problèmes connus ou participer à la discussion.

## Ressources supplémentaires

- Si vous prévoyez d'auto-héberger une organisation Bitwarden, consultez [auto-héberger une organisation](#) pour commencer.

Pour plus d'informations sur le déploiement standard auto-hébergé de Bitwarden, voir :

- [Installer et Déployer - Linux](#)
- [Installer et Déployer - Windows](#)
- [Installer et Déployer - Manuel](#)