

SELF-HOSTING > INSTALLER & DÉPLOYER DES GUIDES >

Déploiement Manuel de Linux

Afficher dans le centre d'aide:

<https://bitwarden.com/help/install-on-premise-manual/>

Déploiement Manuel de Linux

Cet article vous guidera à travers la procédure pour installer et déployer manuellement Bitwarden sur votre propre serveur. Veuillez consulter la documentation de soutien à la [version logicielle](#) de Bitwarden.

⚠ Warning

Les installations manuelles ne doivent être effectuées que par des utilisateurs avancés. Ne poursuivez que si vous êtes très familier avec les technologies Docker et que vous désirez plus de contrôle sur votre installation Bitwarden.

Les installations manuelles n'ont pas la capacité de mettre à jour automatiquement certaines dépendances de l'installation de Bitwarden. Lorsque vous passez d'une version de Bitwarden à la suivante, vous serez responsable des modifications des variables d'environnement requises, des modifications de `nginx default.conf`, des modifications de `docker-compose.yml`, et ainsi de suite.

Nous essaierons de mettre en évidence ces points dans les [notes de version sur GitHub](#). Vous pouvez également surveiller les modifications apportées aux [modèles de dépendance](#) utilisés par le script d'installation de Bitwarden sur GitHub.

Exigences

Avant de procéder à l'installation, veuillez vous assurer que [Docker Engine](#) et [Docker Compose](#) sont installés et prêts à être utilisés sur votre serveur et que votre machine répond aux spécifications système requises.

Si vous construisez vos propres images Bitwarden, les images officielles de .NET Core Runtime (téléchargeables depuis [DockerHub](#)) de la même version majeure que les images Bitwarden sont nécessaires.

Spécifications du système

	Minimum	Recommandé
Processeur	x64, 1.4GHz	x64, 2GHz double cœur
Mémoire	2GB de RAM	4GB de RAM
Stockage	12GB	25GB
Version de Docker	Moteur 19+ et Compose 1.24+	Moteur 19+ et Compose 1.24+

Procédure d'installation

Créer un utilisateur local Bitwarden & répertoire

Nous recommandons de configurer votre serveur avec un compte de service Bitwarden dédié à partir duquel installer et exécuter Bitwarden. En faisant cela, votre instance Bitwarden sera isolée des autres applications en cours d'exécution sur votre serveur.

Ces étapes sont les meilleures pratiques recommandées par Bitwarden, mais ne sont pas obligatoires. Pour plus d'informations, consultez la documentation de Docker sur les [étapes post-installation pour Linux](#).

1. Créez un utilisateur Bitwarden :

Bash

```
sudo adduser bitwarden
```

2. Définissez un mot de passe pour l'utilisateur Bitwarden :

Bash

```
sudo passwd bitwarden
```

3. Créez un groupe Docker (s'il n'existe pas déjà) :

Bash

```
sudo groupadd docker
```

4. Ajoutez l'utilisateur Bitwarden au groupe docker:

Bash

```
sudo usermod -aG docker bitwarden
```

5. Créez un répertoire Bitwarden :

Bash

```
sudo mkdir /opt/bitwarden
```

6. Définissez les autorisations pour le répertoire `/opt/bitwarden` :

Bash

```
sudo chmod -R 700 /opt/bitwarden
```

7. Définissez la propriété de l'utilisateur Bitwarden du répertoire `/opt/bitwarden` :

```
Bash
```

```
sudo chown -R bitwarden:bitwarden /opt/bitwarden
```

Télécharger & configurer

⚠ Warning

Si vous avez créé un utilisateur & répertoire Bitwarden, complétez ce qui suit en tant qu'utilisateur **Bitwarden** depuis le répertoire **/opt/bitwarden**. **N'installez pas Bitwarden en tant que root**, car vous rencontrerez des problèmes lors de l'installation.

Pour télécharger Bitwarden et configurer les ressources du serveur Bitwarden :

1. Téléchargez une version ébauchée des dépendances de Bitwarden (**docker-stub.zip**) depuis les [pages de versions sur GitHub](#). Par exemple:

```
Bash
```

```
curl -L https://github.com/bitwarden/server/releases/download/v<version_number>/docker-stub-US.zip \
-o docker-stub-US.zip
```

2. Créez un nouveau répertoire nommé **bwdata** et extrayez **docker-stub.zip** dedans, par exemple :

```
Bash
```

```
unzip docker-stub-US.zip -d bwdata
```

Une fois décompressé, le répertoire **bwdata** correspondra à ce que le fichier **docker-compose.yml** attend pour le mappage de volume. Vous pouvez, si vous le souhaitez, modifier l'emplacement de ces mappages sur la machine hôte.

3. Dans **./bwdata/env/global.override.env**, éditer les variables d'environnement suivantes :

- **globalSettings__baseServiceUri__vault=**: Entrez le domaine de votre instance Bitwarden.
- **globalSettings__sqlServer__ConnectionString=**: Remplacez le **MOT_DE_PASSE_ALÉATOIRE_DE_LA_BASE_DE_DONNÉES** par un mot de passe sécurisé à utiliser dans une étape ultérieure.
- **globalSettings__identityServer__certificatePassword**: Définissez un mot de passe de certificat sécurisé pour une utilisation dans une étape ultérieure.
- **globalSettings__internalIdentityKey=**: Remplacez **RANDOM_IDENTITY_KEY** par une chaîne de clé aléatoire.
- **globalSettings__oidcIdentityClientKey=**: Remplacez **RANDOM_IDENTITY_KEY** par une chaîne de clés aléatoire.
- **globalSettings__duo__aKey=**: Remplacez **RANDOM_DUO_AKEY** par une chaîne de clés aléatoire.

- `globalSettings__installation__id=`: Entrez un identifiant d'installation récupéré depuis <https://bitwarden.com/host>.
- `globalSettings__installation__key=`: Entrez une clé d'installation récupérée depuis <https://bitwarden.com/host>.

💡 Tip

À ce moment, envisagez également de définir des valeurs pour toutes les variables `globalSettings__mail__smtp__` et pour `adminSettings__admins`. En faisant cela, vous configurerez le serveur de messagerie SMTP utilisé pour envoyer des invitations aux nouveaux membres de l'organisation et fournir l'accès au [Portail de l'Administrateur Système](#).

[En savoir plus sur les variables d'environnement.](#)

4. À partir de `./bwdata`, générer un fichier de certificat `.pfx` pour le conteneur d'identité et le déplacer vers le répertoire du volume mappé (par défaut, `./bwdata/identity/`). Par exemple, exécutez les commandes suivantes :

Bash

```
openssl req -x509 -newkey rsa:4096 -sha256 -nodes -keyout identity.key -out identity.crt -subj  
"/CN=Bitwarden IdentityServer" -days 10950
```

et

Bash

```
openssl pkcs12 -export -out ./identity/identity.pfx -inkey identity.key -in identity.crt -passou  
t pass:IDENTITY_CERT_PASSWORD
```

Dans la commande ci-dessus, remplacez `IDENTITY_CERT_PASSWORD` par le mot de passe du certificat créé et utilisé dans **Étape 3**.

5. Copier `identity.pfx` dans le répertoire `./bwdata/ssl`.
6. Créez un sous-répertoire dans `./bwdata/ssl` nommé pour votre domaine, par exemple :

Bash

```
mkdir ./ssl/bitwarden.example.com
```

7. Fournissez un certificat SSL de confiance et une clé privée dans le sous-répertoire nouvellement créé `./bwdata/ssl/bitwarden.example.com`.

📌 Note

Ce répertoire est mappé au conteneur NGINX à `/etc/ssl`. Si vous ne pouvez pas fournir un certificat SSL de confiance, placez devant l'installation un proxy qui fournit un point de terminaison HTTPS aux applications client Bitwarden.

8. Dans `./bwdata/nginx/default.conf`:

1. Remplacez toutes les instances de `bitwarden.example.com` par votre domaine, y compris dans l'en-tête **Politique de Sécurité de Contenu**.
2. Définissez les variables `ssl_certificate` et `ssl_certificate_key` sur les chemins du certificat et de la clé privée fournis dans l'Étape 7.
3. Effectuez l'une des actions suivantes, en fonction de la configuration de votre certificat :
 - Si vous utilisez un certificat SSL de confiance, définissez la variable `ssl_trusted_certificate` sur le chemin d'accès à votre certificat.
 - Si vous utilisez un certificat auto-signé, mettez en commentaire la variable `ssl_trusted_certificate`.

9. Dans `./bwdata/env/mssql.override.env`, remplacez `RANDOM_DATABASE_PASSWORD` par le mot de passe créé à l'étape 3.

10. Dans `./bwdata/web/app-id.json`, remplacez `bitwarden.example.com` par votre domaine.

11. Dans `./bwdata/env/uid.env`, définissez l'UID et le GID des utilisateurs et du groupe **Bitwarden** que vous avez créé précédemment afin que les conteneurs s'exécutent sous ceux-ci, par exemple :

Bash

```
LOCAL_UID=1001
LOCAL_GID=1001
```

Démarrez votre serveur

Démarrez votre serveur Bitwarden avec la commande suivante :

Bash

```
docker compose -f ./docker/docker-compose.yml up -d
```

Vérifiez que tous les conteneurs fonctionnent correctement :

Bash

```
docker ps
```

```
bitwarden@bitwarden:/opt/bitwarden$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS                    PORTS                                NAMES
4b92b8f5ff16   bitwarden/nginx:1.38.2             "/entrypoint.sh"        2 minutes ago Up 2 minutes (healthy)   80/tcp, 0.0.0.0:80->8080/tcp, 0.0.0.0:443->8443/tcp   bitwarden-nginx
b68c1df89320   bitwarden/portal:1.38.2            "/entrypoint.sh"        3 minutes ago Up 2 minutes (healthy)   5000/tcp                                bitwarden-portal
5731d5d9966df   bitwarden/admin:1.38.2             "/entrypoint.sh"        3 minutes ago Up 2 minutes (healthy)   5000/tcp                                bitwarden-admin
0703a3bee3fd   bitwarden/identity:1.38.2          "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-identity
2000bd327f60   bitwarden/api:1.38.2               "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-api
523644f15d2f   bitwarden/web:2.17.1               "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-web
72e11ccc7d22   bitwarden/attachments:1.38.2       "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-attachments
406adf1a6c5c   bitwarden/sso:1.38.2               "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-sso
9e0e8cb75b29   bitwarden/events:1.38.2            "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-events
d01eff6f324f   bitwarden/notifications:1.38.2     "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-notifications
4e4d57418a79   bitwarden/mssql:1.38.2             "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)                                bitwarden-mssql
fec35a34b02c   bitwarden/icons:1.38.2             "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-icons
```

docker-sain.png

Félicitations ! Bitwarden 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. Vous devrez avoir configuré les variables d'environnement SMTP (voir [Variables d'Environnement](#)) afin de vérifier le courriel pour votre nouveau compte.

Prochaines étapes :

- Si vous prévoyez d'auto-héberger une organisation Bitwarden, consultez [auto-héberger une organisation](#) pour commencer.
- Pour plus d'informations, consultez les [FAQ sur l'auto-hébergement](#).

Mettez à jour votre serveur

Mettre à jour un serveur auto-hébergé qui a été installé et déployé manuellement est différent de la [procédure de mise à jour standard](#). Pour mettre à jour votre serveur installé manuellement :

1. Téléchargez la dernière archive [docker-stub.zip](#) depuis les [pages de versions](#) sur GitHub.
2. Décompressez la nouvelle archive [docker-stub.zip](#) et comparez son contenu avec ce qui se trouve actuellement dans votre répertoire [bwdata](#), en copiant tout ce qui est nouveau dans les fichiers préexistants de [bwdata](#).
Ne remplacez pas votre répertoire [bwdata](#) existant par le contenu de la nouvelle archive [docker-stub.zip](#), car cela écraserait tout travail de configuration personnalisé que vous avez effectué.
3. Exécutez la commande suivante pour redémarrer votre serveur avec votre configuration mise à jour et les derniers conteneurs :

Bash

```
docker compose -f ./docker/docker-compose.yml down && docker compose -f ./docker/docker-compose.yml up -d
```