

SELF-HOSTING > インストール&デプロイガイド >

# 統一デプロイメントベータ

ヘルプセンターで表示:

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

## 統一デプロイメントベータ

### Note

このソリューションはベータ版で、個人使用を目的としています。ビジネスプランは、公式にサポートされている標準的なデプロイメントオプションを使用すべきです。

Bitwarden統合自己ホスト型デプロイメントがベータ版である間、統合をインストールする人々は、最新のイメージを引き出す自動アップグレード手順を設定すべきではありません。Bitwardenは、アップグレードする前にリリースの安定化に時間をかけることを推奨します。

[問題を報告する方法を学びます。](#)

この記事では、Bitwarden統合自己ホスト型デプロイメントのインストールと起動方法を説明します。このデプロイメント方法を使用して：

- 単一のDockerイメージを使用してBitwardenをデプロイすることで、設定を簡素化し、リソース使用量（CPU、メモリ）を最適化します。
- MSSQL、PostgreSQL、SQLite、およびMySQL/MariaDBなどの異なるデータベースソリューションを利用します。
- Raspberry PiやNASサーバーなどの代替システムでARMアーキテクチャを実行します。

## システム要件

Bitwardenの統一デプロイメントには以下が必要です：

- 少なくとも200 MBのRAM
- ストレージ 1GB
- Dockerエンジン19+

## Dockerをインストールする

統一されたデプロイメントは、あなたのマシン上でDockerコンテナを使用して実行されます。統一デプロイメントは、任意のDocker編集またはプランで実行できます。あなたのインストールに最適な版を評価してください。

インストールを進める前に、あなたのマシンにDockerをインストールしてください。次のDockerのドキュメントを参照してください：

- [Dockerエンジンをインストールします](#)

## Bitwardenを統一して実行します

統一デプロイメントは、`docker run`コマンド（[こちらを参照](#)）またはDocker Compose（[こちらを参照](#)）を使用して実行できます。いずれの場合でも、コンテナの環境変数を指定する必要があります。

## 環境変数を指定する

統一デプロイメントを実行するには、コンテナの環境変数を設定する必要があります。環境変数は、`settings.env`ファイルを作成することで指定できます。その例は私たちの[GitHubリポジトリ](#)で見つけることができます。または、`docker run`メソッドを使用している場合は、`--env`フラグを使用することで指定できます。よりパーソナライズされた統一デプロイメント体験のために、いくつかのオプションの変数が利用可能です。これらの変数に関する追加の詳細は、[こちら](#)で見つけることができます。

少なくとも、例の`.env`ファイルの# 必須設定 #セクションにある変数の値を設定してください：

変数	説明
<code>BW_DOMAIN</code>	<code>bitwarden.yourdomain.com</code> をBitwardenがアクセスされるドメインに置き換えてください。

変数	説明
BW_DB_PROVIDER	あなたのBitwardenサーバーに使用するデータベースプロバイダー。利用可能なオプションは、 <code>sqlserver</code> 、 <code>postgresql</code> 、 <code>sqlite</code> 、または <code>mysql/mariadb</code> です。
BW_DB_SERVER	あなたのデータベースが稼働しているサーバーの名前。
BW_DB_データベース	あなたのBitwardenデータベースの名前。
BW_DB_USERNAME	Bitwardenデータベースにアクセスするためのユーザー名。
BW_DB_パスワード	Bitwardenデータベースにアクセスするためのパスワード。
BW_DB_FILE	あなたのデータベースファイルへのパスを指定したい場合にのみ、 <code>sqlite</code> が必要です。指定しない場合、 <code>sqlite</code> は自動的に <code>vault.db</code> ファイルを <code>/etc/bitwarden</code> ボリュームの下に作成します。
BW_INSTALLATION_ID	<a href="https://bitwarden.com/host/">https://bitwarden.com/host/</a> から生成された有効なインストールID。
BW_インストールキー	<a href="https://bitwarden.com/host/">https://bitwarden.com/host/</a> から生成された有効なインストールキー。

### ① Note

Bitwardenの標準デプロイメントとは異なり、統一デプロイメントはデータベースが付属していません。既存のデータベースを使用することも、この例に記載されているように新しいものを作成することもできますが、いずれの場合もここに記載されている**BW\_DB\_...**変数に有効な情報を入力する必要があります。

非MSSQLデータベースプロバイダーを使用すると、パフォーマンスの問題が発生する可能性があります。これらのプラットフォームのサポートは、ベータ版全体で引き続き作業が行われています。あなたのBitwarden統一デプロイメントに関連する何でもレポートするために、この問題テンプレートを使用してください。既知の問題を追跡したり、ディスカッションに参加したりするためには、このページをご覧ください。

## Docker runを使用する

統一デプロイメントは、以下の例のように`docker run`コマンドで実行できます。

### Bash

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

上記のコマンドには、`docker run`コマンドに必要ないくつかのオプションが機能として含まれています。これには以下が含まれます：

名前、省略形	説明
--detach, -d	コンテナをバックグラウンドで実行し、コンテナIDを出力します。
--名前	コンテナに名前を付けてください。Bitwarden が例として使用されています。
--ボリューム, -v	ボリュームをバインドマウントします。最低限、/etc/bitwardenをマウントしてください。
--公開, -p	ホストにコンテナポートをマップします。この例では、ポート80:8080 がマッピングされています。SSLを設定する際には、ポート8443が必要です。
--env-file	環境変数を読み込むファイルへのパス。あるいは、--env フラグを使用して環境変数をインラインで宣言します (詳細を学ぶ)。

コマンドを実行したら、次のコマンドでコンテナが実行中で健康であることを確認してください:

*Bash*

```
docker ps
```

おめでとうございます! あなたの統合デプロイメントは現在、<https://your.domain.com>で稼働しています。それが機能していることを確認するために、ブラウザでウェブ保管庫を訪れてください。あなたは今、新しいアカウントを登録してログインすることができます。

### Docker Composeを使用する

Docker Composeを使用した統一デプロイメントの実行には、Docker Composeバージョン1.24以上が必要です。Docker composeを使用して統一デプロイメントを実行するには、たとえばdocker-compose.ymlファイルを作成します。

```
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:
```

`docker-compose.yml` ファイルで、以下を含む任意の設定を行ってください：

- ログおよびBitwardenデータのボリュームをマッピングします。
- ポートのマッピング。
- データベースイメージの設定。<sup>a</sup>

<sup>a</sup>上記の例のように、`docker-compose.yml` でデータベースを設定するのは、Bitwardenで使用するための新しいデータベースサーバーを作成したい場合だけです。MySQL、MSSQL、およびPostgreSQLのサンプル設定は、私たちの例のファイルに含まれています。

あなたの`docker-compose.yml`と`settings.env`ファイルが作成されたら、以下のコマンドを実行して統合サーバーを起動します：

```
Bash
```

```
docker compose up -d
```

すべてのコンテナが正しく動作していることを確認してください。

```
Bash
```

```
docker ps
```

おめでとうございます！あなたの統一デプロイメントは現在、<https://your.domain.com>で稼働しています。それが機能していることを確認するために、ブラウザでウェブ保管庫を訪れてください。あなたは今、新しいアカウントを登録してログインすることができます。

## あなたのサーバーを更新してください

あなたの統一デプロイメントを更新するには：

### ⇒Docker run 更新

1. 実行中のDockerコンテナを停止します:

```
Bash
```

```
docker stop bitwarden
```

2. Dockerコンテナを削除します:

```
Bash
```

```
docker rm bitwarden
```

3. 次のコマンドを実行して、最新のBitwarden統合イメージをプルしてください:

```
Bash
```

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

4. Dockerコンテナを再度実行します:

```
Bash
```

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

### ⇒Docker Composeの更新

1. 実行中のDockerコンテナを停止します:

```
Bash
```

```
docker compose down
```

2. 次のコマンドを実行して、最新のBitwarden統合イメージをプルしてください:

```
Bash
```

```
docker compose pull
```

3. 更新が必要なコンテナを再作成してください。

```
Bash
```

```
docker compose up -d
```

4. コンテナが稼働していることを確認してください:

```
Bash
```

```
docker compose ps
```

## 環境変数

統一デプロイメントは、いくつかの標準的なBitwardenサービスなしでデフォルトで動作します。これにより、統合デプロイメントのカスタマイズと最適化が向上します。これらのサービスを設定し、さらにオプションの設定を、さまざまな環境変数を編集することで構成します。

### Note

環境変数を変更するたびに、Dockerコンテナを再作成する必要があります。もっと詳しくは[こちら](#)をご覧ください。

## ウェブサーバーポート

変数	説明
BW_PORT_HTTP	HTTPトラフィックに使用するポートを変更します。デフォルトでは、 <b>8080</b> 。
BW_PORT_HTTPS	HTTPSトラフィックに使用するポートを変更します。デフォルトでは、 <b>8443</b> 。

## SSL

これらの値を使用して証明書の設定を変更します。

変数	説明
BW_ENABLE_SSL	SSL/TLSを使用します。真/偽。デフォルト 偽。  Bitwardenが正常に機能するためにはSSLが必要です。あなたがBitwardenコンテナでSSLを設定していない場合、BitwardenをSSLプロキシでフロントするべきです。
BW_SSL_CERT	あなたのSSL証明書ファイルの名前。ファイルはコンテナ内の/etc/bitwarden ディレクトリに配置されていなければなりません。デフォルト ssl.crt。
BW_SSL_KEY	あなたのSSLキーファイルの名前。ファイルはコンテナ内の/etc/bitwarden ディレクトリに配置されていなければなりません。デフォルト ssl.key。
BW_ENABLE_SSL_CA	証明機関 (CA) によるサービスでSSLを使用します。真/偽。デフォルト 偽。
BW_SSL_CA_CERT	あなたのSSL CA証明書の名前。ファイルはコンテナ内の/etc/bitwarden ディレクトリに配置されていなければなりません。デフォルト ca.crt。
BW_ENABLE_SSL_DH	Diffie-Hellmanキー交換を使用してSSLを使用します。真/偽。デフォルト 偽。
BW_SSL_DH_CERT	あなたのDiffie-Hellmanパラメータファイルの名前。ファイルはコンテナ内の/etc/bitwarden ディレクトリに配置されていなければなりません。デフォルト dh.pem。
BW_SSL_PROTOCOLS	NGINXによって使用されるSSLバージョン。推奨のデフォルトのために空白にしておいてください。もっと学ぶ
BW_SSL_CIPHERS	NGINXによって使用されるSSL暗号スイート。推奨のデフォルトのために空白にしておいてください。もっと学ぶ

### ① Note

既存のSSL証明書を使用している場合、`settings.env`で適切なSSLオプションを有効にする必要があります。SSLファイルは/etc/bitwarden に保存する必要があり、`docker-compose.yml`ファイルで参照できます。これらのファイルは、`settings.env`で設定された名前と一致しなければなりません。

デフォルトの動作は、SSLが有効で、期待される場所 (/etc/bitwarden) に既存の証明書ファイルがない場合、自己署名証明書を生成することです。

### サービス

追加のサービスは、エンタープライズやチームのニーズなど、特定の使用ケースに対して有効化または無効化することができます。以下の値を変更することにより行います：



変数	説明
BW_ENABLE_ADMIN	<p>このサービスを無効にしないでください。</p> <p>管理者パネルの機能についてここで詳しく学びましょう。デフォルト <b>真</b>。</p>
BW_ENABLE_API	<p>このサービスを無効にしないでください。</p> <p>デフォルト <b>真</b>。</p>
BW_ENABLE_EVENTS	<p>チームとエンタープライズイベントモニタリングのためのBitwardenイベントログを有効にするか無効にします。デフォルト <b>偽</b>。</p>
BW_ENABLE_ICONS	<p>ログインアイテムのURIで設定されたBitwardenブランドのアイコンを有効にするか無効にするかを設定します。もっと詳しくは<a href="#">こちら</a>をご覧ください。デフォルト <b>真</b>。</p>
BW_ENABLE_IDENTITY	<p>このサービスを無効にしないでください。</p> <p>デフォルト <b>真</b>。</p>
BW_ENABLE_NOTIFICATIONSを有効にする	<p>モバイルデバイスへのプッシュ通知を受信するための通知サービスを有効または無効にし、デバイスでのログイン、モバイル保管庫の同期などを使用します。デフォルト <b>真</b>。</p>
BW_ENABLE_SCIM	<p>エンタープライズ組電のSCIMを有効にするか無効にするか。デフォルト <b>偽</b>。</p>
BW_ENABLE_SSO	<p>エンタープライズ組織のSSOサービスを有効にするか無効にするか。デフォルト <b>偽</b>。</p>
BW_ICONS_PROXY_TO_CLOUD	<p>このサービスを有効にすると、システムメモリの負荷を軽減するために、アイコンサービスのリクエストがクラウドサービスを介して操作されるようになります。この設定を使用する場合、コンテナの負荷を軽減するために、<b>BW_ENABLE_ICONS</b> は <b>false</b> に設定する必要があります。デフォルト <b>偽</b>。</p>

## メール

あなたの統合デプロイメントのためのSMTP設定を構成します。選択したメールSMTPプロバイダから次のフィールドに情報をコピーしてください。

変数	説明
globalSettings__mail__replyToEmail	あなたのサーバーの返信用メールアドレスを入力してください。

変数	説明
<code>globalSettings__mail__smtp__host</code>	あなたのSMTPサーバーのホストドメインを入力してください。
<code>globalSettings__mail__smtp__port</code>	SMTPホストからポート数値を入力してください。
<code>globalSettings__mail__smtp__ssl</code>	あなたのSMTPホストがSSLを使用している場合、 <b>true</b> を入力してください。 あなたのホストがTLSサービスを使用している場合、 <b>false</b> に値を設定します。
<code>globalSettings__mail__smtp__username</code>	SMTPのユーザー名を入力してください。
<code>globalSettings__mail__smtp__password</code>	SMTPのパスワードを入力してください。

### Yubico API (YubiKey)

変数	説明
<code>globalSettings__yubico__clientId</code>	Yubico Keyから受け取ったIDに値を置き換えてください。 Yubico Keyに <a href="#">ここから登録</a> してください。
<code>globalSettings__yubico__キー</code>	Yubicoから受け取ったキー値を入力してください。

### データベース設定

統一デプロイメントと互換性のあるさまざまなデータベースオプションを利用するには、追加の`.env`設定が必要になります。

#### ⇒MySQL/MariaDB

`settings.env`で：

##### 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

`settings.env`で：

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

settings.envで:

Bash

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

sqlite の値を割り当てると、vault.db ファイルが自動的に/etc/bitwarden ボリュームに作成されます。BW\_DB\_FILE は、別のデータベースファイルへのパスを指定する場合にのみ必要です。

### ⇒PostgreSQL

settings.envで:

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
```

### その他

変数	説明
globalSettings__userRegistrationを無効にする	ユーザーアカウント登録機能を有効にするか無効にするか。
globalSettings__hibpApiKey	Have I Been Pwndから提供されたAPIキーを入力してください。APIキーを受け取るために <a href="#">ここで登録</a> してください。
adminSettings__管理者	管理者のメールアドレスを入力してください。

変数	説明
BW_REAL_IPS	<code>nginx.conf</code> で実際のIPをカンマで区切られたリストで定義します。 クライアントのIPアドレスを転送するプロキシサーバーを定義するのに便利です。 <a href="#">もっと学ぶ</a> 。
BW_CSP	Content-Security-Policy パラメータ。このパラメータを再設定すると、機能が壊れる可能性があります。 このパラメータを変更することで、あなたはこの値を維持する責任を負うことになります。
BW_DB_PORT	データベーストラフィックのためのカスタムポートを指定します。指定されていない場合、 デフォルトは選択したデータベースプロバイダーによります。

### コンテナを再起動してください

環境変数を変更した後でDockerコンテナを再起動するには、Bitwarden統合デプロイメントディレクトリから次のコマンドを実行します:

#### ⇒Dockerを実行する

1. 実行中のDockerコンテナを停止します:

*Bash*

```
docker stop bitwarden
```

2. Dockerコンテナを削除します:

*Bash*

```
docker rm bitwarden
```

3. Dockerコンテナを再度実行します:

*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. 実行中のDockerコンテナを停止します:

*Bash*

```
docker compose down
```

2. コンテナを再作成します。

```
Bash
```

```
docker compose up -d
```

3. 次のようにして、コンテナが正しく動作していることを確認してください:

```
Bash
```

```
docker compose ps
```

## メモリ使用量

デフォルトでは、Bitwardenコンテナは利用可能なメモリを消費し、それはしばしば実行するために必要な最小限以上になります。メモリに配慮した環境では、`docker -m`または`--memory=`を使用してBitwardenコンテナのメモリ使用量を制限できます。

名前、省略形	説明
<code>--memory=</code> , <code>-m</code>	コンテナが使用できる最大のメモリ量。Bitwardenは少なくとも200mが必要です。詳しくは <a href="#">Dockerのドキュメンテーション</a> をご覧ください。

Docker Composeでメモリ使用量を制御するには、`mem_limit`キーを使用します。

```
Bash
```

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

## 問題をレポートする

Bitwardenの統一デプロイメントがベータリリースのままである間、GitHubを通じて問題をレポートし、フィードバックを提供することをお勧めします。あなたのBitwarden統一デプロイメントに関連する何でもレポートするために、[この問題テンプレート](#)を使用してください。既知の問題を追跡したり、ディスカッションに参加したりするためには、[このページ](#)をご覧ください。

## 追加のリソース

- Bitwarden組織を自己ホスト型で運用する予定がある場合は、[組織を自己ホストする](#)を参照して開始してください。

Bitwardenの標準的な自己ホスト型デプロイメントについての詳細は、以下をご覧ください:

- [インストールとデプロイ - リナックス](#)
- [インストールとデプロイ - ウィンドウズ](#)
- [インストールとデプロイ - 手動](#)