

SELF-HOSTING > インストール&デプロイガイド >

Linux 手動デプロイメント

ヘルプセンターで表示:

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

Linux 手動デプロイメント

この記事では、Bitwardenを自分のサーバーに手動でインストールし、デプロイする手順を説明します。Bitwardenソフトウェアリリースサポートのドキュメンテーションをご確認ください。

⚠ Warning

手動インストールは、**上級ユーザーのみが行うべきです**。Docker技術に非常に詳しく、Bitwardenのインストールに対してより多くのコントロールを望む場合のみ、進めてください。

手動インストールでは、Bitwardenのインストールの特定の依存関係を自動的に更新する能力が欠けています。Bitwardenの一つのバージョンから次のバージョンへアップグレードすると、必要な環境変数の変更、nginx `default.conf`の変更、`docker-compose.yml`の変更など、あなたが責任を持って変更を行う必要があります。

これらをGitHubのリリースメモで強調しようと思います。Bitwardenのインストールスクリプトで使用される**依存関係テンプレート**の変更もGitHubで監視することができます。

要件

インストールを進める前に、**Docker Engine**と**Docker Compose**がサーバーにインストールされ、使用準備が整っていること、そしてあなたのマシンが必要なシステム仕様を満たしていることを確認してください。

自分でBitwardenのイメージを作成している場合、公式の.NET Core Runtimeイメージ (**DockerHub**からダウンロード) が必要です。これらのイメージは、Bitwardenのイメージと同じメジャーリリースバージョンである必要があります。

システム仕様

	最小限	おすすめ
プロセッサ	x64、1.4GHz	x64、2GHzデュアルコア
記憶	2GB RAM	4GB RAM
ストレージ	12GB	25GB
Dockerバージョン	エンジン19+およびコンポーズ1.24+	エンジン19+およびコンポーズ1.24+

インストール手順

Bitwarden ローカルユーザー&ディレクトリを作成する

私たちは、Bitwardenをインストールし実行するための専用のBitwardenサービスアカウントを持つサーバーを設定することをお勧めします。これにより、サーバー上で実行されている他のアプリケーションからBitwardenインスタンスが隔離されます。

これらの手順はBitwardenが推奨するベストプラクティスですが、必須ではありません。詳細については、DockerのLinux用のインストール後の手順のドキュメンテーションをご覧ください。

1. Bitwardenユーザーを作成する：

```
Bash
```

```
sudo adduser bitwarden
```

2. Bitwardenユーザーのパスワードを設定します:

```
Bash
```

```
sudo passwd bitwarden
```

3. Docker グループを作成します（まだ存在しない場合）：

```
Bash
```

```
sudo groupadd docker
```

4. Bitwardenユーザーをdockerグループに追加します:

```
Bash
```

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

5. Bitwardenディレクトリを作成します：

```
Bash
```

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

6. `/opt/bitwarden`ディレクトリの権限を設定します：

```
Bash
```

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

7. `/opt/bitwarden`ディレクトリのBitwardenユーザー所有権を設定します:

```
Bash
```

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

ダウンロード&設定

⚠ Warning

あなたがBitwardenユーザー&ディレクトリを作成した場合、**bitwarden**ユーザーとして`/opt/bitwarden`ディレクトリから以下を完了してください。インストール中に問題が発生する可能性があるため、**Bitwarden**を**root**としてインストールしないでください。

Bitwardenをダウンロードし、Bitwardenサーバーのアセットを設定するには：

1. Bitwardenの依存関係のスタブ版 (**docker-stub.zip**) をGitHubのリリースページからダウンロードしてください。例えば：

Bash

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

2. 新しいディレクトリを**bwdata**という名前で作成し、**docker-stub.zip**をそれに抽出します。例えば：

Bash

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

一度解凍すると、**bwdata**ディレクトリは、**docker-compose.yml**ファイルのボリュームマッピングが期待するものと一致します。あなたの希望により、これらのマッピングの位置をホストマシン上で変更することができます。

3. `./bwdata/env/global.override.env`で、以下の環境変数を編集してください：

- `globalSettings__baseServiceUri__vault=:` あなたのBitwardenインスタンスのドメインを入力してください。
- `globalSettings__sqlServer__ConnectionString=:` 後のステップで使用するために、**RANDOM_DATABASE_PASSWORD**を安全なパスワードに置き換えてください。
- `globalSettings__identityServer__certificatePassword:` 後のステップで使用するための安全な証明書のパスワードを設定します。
- `globalSettings__internalIdentityKey=:` **RANDOM_IDENTITY_KEY**をランダムなキー文字列に置き換えてください。
- `globalSettings__oidcIdentityClientKey=:` **RANDOM_IDENTITY_KEY**をランダムなキー文字列に置き換えてください。
- `globalSettings__duo__aKey=:` **RANDOM_DUO_AKEY**をランダムなキー文字列に置き換えてください。
- `globalSettings__installation__id=:` <https://bitwarden.com/host>から取得したインストールIDを入力してください。
- `globalSettings__installation__key=:` <https://bitwarden.com/host>から取得したインストールキーを入力してください。

💡 Tip

この時点で、すべての`globalSettings__mail__smtp__`変数と`adminSettings__admins`の値も設定することを検討してください。これにより、新しい組織メンバーへの招待をSendするためのSMTPメールサーバーが設定され、システム管理者ポータルへのアクセスが提供されます。

環境変数についてもっと学ぶ。

4. `./bwdata`から、IDコンテナのための`.pfx`証明書ファイルを生成し、それをマップされたボリュームディレクトリ（デフォルトでは`./bwdata/identity/`）に移動してください。例えば、次のコマンドを実行します:

Bash

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

そして

Bash

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

上記のコマンドでは、`IDENTITY_CERT_PASSWORD`をステップ3で作成し使用した証明書のパスワードに置き換えてください。

5. `identity.pfx`を`./bwdata/ssl`ディレクトリにコピーしてください。
6. あなたのドメインの名前で`./bwdata/ssl`にサブディレクトリを作成します。例えば:

Bash

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

7. 信頼できるSSL証明書と秘密鍵を新しく作成された`./bwdata/ssl/bitwarden.example.com`サブディレクトリに提供してください。

📌 Note

このディレクトリは、`/etc/ssl`のNGINXコンテナにマッピングされています。信頼できるSSL証明書を提供できない場合は、BitwardenクライアントアプリケーションにHTTPSエンドポイントを提供するプロキシでインストールをフロントにしてください。

8. `./bwdata/nginx/default.conf`で:

1. すべての `bitwarden.example.com` のインスタンスをあなたのドメインに置き換えてください、`Content-Security-Policy` ヘッダーも含めて。
2. `ssl_certificate` と `ssl_certificate_key` の変数を、**ステップ7** で提供された証明書と秘密鍵のパスに設定します。
3. 次のアクションのいずれかを取ります。証明書の設定によります：
 - 信頼できるSSL証明書を使用している場合、`ssl_trusted_certificate` 変数を証明書のパスに設定します。
 - 自己署名証明書を使用する場合は、`ssl_trusted_certificate` 変数をコメントアウトしてください。
9. `./bwdata/env/mssql.override.env` で、`RANDOM_DATABASE_PASSWORD` を **ステップ3** で作成したパスワードに置き換えてください。
10. `./bwdata/web/app-id.json` で、`bitwarden.example.com` をあなたのドメインに置き換えてください。
11. `./bwdata/env/uid.env` で、コンテナがそれらの下で実行されるように、**前に作成した** `bitwarden` ユーザーとグループの UID と GID を設定します。次に例を示します。

Bash

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

あなたのサーバーを起動してください

次のコマンドであなたのBitwardenサーバーを起動してください:

Bash

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

bitwarden@bitwarden:~/opt/bitwarden\$ docker ps

docker-healthy.png

おめでとうございます！Bitwarden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。新しいアカウントのメールアドレスを確認するためには、SMTP環境変数を設定する必要があります（[環境変数](#)を参照してください）。

次のステップ：

- 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。
- 詳細情報は、[自己ホスト型FAQ](#)をご覧ください。

あなたのサーバーを更新してください

自己ホスト型のサーバーを手動でインストールし、デプロイした場合の更新は、[標準的な更新手順](#)とは異なります。手動でインストールしたサーバーを更新するには：

1. 最新の[docker-stub.zip](#)アーカイブをGitHubのリリースページからダウンロードしてください。
2. 新しい[docker-stub.zip](#)アーカイブを解凍し、その内容を現在の**bwdata**ディレクトリの内容と比較し、新しいものを**bwdata**の既存のファイルにコピーします。
しないでください、新しい[docker-stub.zip](#)アーカイブの内容で既存の**bwdata**ディレクトリを上書きしないでください。これは、あなたが行ったカスタム設定作業を上書きすることになります。
3. 次のコマンドを実行して、更新された設定と最新のコンテナを使用してサーバーを再起動します：

Bash

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