

SELF-HOSTING > インストール&デプロイガイド >

# Linux オフラインデプロイメント

ヘルプセンターで表示:

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

## Linux オフラインデプロイメント

この記事では、**オフラインまたはエアギャップ環境**の自分のサーバーにBitwardenをインストールして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、以下の要件が満たされていることを確認してください:

- あなたのサーバーに[Dockerエンジン](#)と[Docker Compose](#)がインストールされ、使用準備が整っています。
- インターネットアクセスが可能なマシンを使用して、Bitwardenサーバーリポジトリの[リリースページ](#)から最新の[docker-stub.zip](#)ファイルをダウンロードし、このファイルをサーバーに転送しました。
- あなたの環境では、オフラインのSMTPサーバーが設定され、アクティブになっています。

### 📌 Note

Bitwardenは現在、DockerおよびDocker Compose環境でのみサポートされています。Bitwardenのインストールスクリプトと手動インストールの成果物は、BitwardenスタックとKubernetesの広範な知識がなければ、正確にKubernetesマニフェストに変換することはできません。インストールアーティファクトの自動変換は現時点では推奨されておらず、デプロイメント環境が壊れる可能性があります。

## 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

	最小限	おすすめ
Dockerバージョン	エンジン19+およびコンポーザ1.24+	エンジン19+およびコンポーザ1.24+

## インストール手順

### あなたのドメインを設定してください

デフォルトでは、Bitwardenはホストマシンのポート80 ([http](http://)) および443 ([https](https://)) を通じて提供されます。これらのポートを開けて、ネットワーク内外からBitwardenにアクセスできるようにしてください。インストール中に異なるポートを選択することもできます。

インターネット経由でBitwardenを提供している場合、特に、ホストマシンを指すDNSレコードを持つドメイン名 (例: [bitwarden.example.com](http://bitwarden.example.com)) を設定することをお勧めします。

### 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あなたの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
```

## あなたのマシンを設定してください

### ⚠ 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。

あなたのBitwardenサーバーに必要なアセットを構成するためのマシンを設定するには :

1. 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。

2. `./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。
- `globalSettings__pushRelayBaseUri=`: この変数は空白であるべきです。詳細については、[プッシュリレーの設定](#)をご覧ください。

#### 💡 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](#)。

3. `./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2**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。

4. `identity.pfx`を`./bwdata/ssl`ディレクトリにコピーしてください。
5. 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
```

6. 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。

#### ① 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。

7. `./bwdata/nginx/default.conf`:にて

1. すべての `bitwarden.example.com` のインスタ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6**で提供された証明書と秘密鍵のパスに設定します。
3. あなたの証明書の設定に応じて、以下のアクションのいずれかを取ってください:
  - 信頼できるSSL証明書を使用している場合は、`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。

8. `./bwdata/env/mssql.override.env`で、`RANDOM_DATABASE_PASSWORD`を**ステップ2**で作成したパスワードに置き換えてください。

9. `./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。

10. `./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
```

## 画像をダウンロード&転送する

あなたのオフラインマシンで使用するためのDockerイメージを取得するには：

1. インターネットに接続されたマシンから、`bitwarden/xxx:latest`のすべてのdockerイメージをダウンロードします。これらは`docker-compose.yml`ファイルにリストされており、`docker-stub.zip`にあります。
2. 各画像を `.img`ファイルに保存します。例えば：

#### Bash

```
docker image save -o mssql.img bitwarden/mssql:version
```

3. すべての `.img`ファイルをオフラインのマシンに転送してください。

4. あなたのオフラインマシンで、各 `.img`ファイルをロードしてローカルのDockerイメージを作成します。例えば：

Bash

```
docker image load -i mssql.img
```

## 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
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
5731d5d966df   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
406eadf1a6c5c   bitwarden/sso:1.38.2               "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-sso
9a0e8cb75b29   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
4ed457418a79   bitwarden/mssql:1.38.2             "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-mssql
fec45a34b02c   bitwarden/icons:1.38.2             "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp                                bitwarden-icons
bitwarden@bitwarden:/opt/bitwarden$
```

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 (環境変数を参照) を設定する必要があります。

## 次のステップ:

- Bitwardenの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てください。  
しないでください、既存の`bwdata`ディレクトリを新しい`docker-stub.zip`アーカイブの内容で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. 最新のコンテナイメージをダウンロードし、オフラインマシンに転送してください上記の文書化された通りに。
4. 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
```