

SELF-HOSTING > インストール&デプロイガイド >

Windows 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-windows/>

Windows 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Windows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 사용되는 **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 EngineとDocker Composeがインストールされ、使用準備が整っています。この設定中には、**チェックを外す**必要があります**Hyper-Vの代わりにWSL2を使用する (推奨)** オプション。
- 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、このファイルをあなたの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。
- (オプション) OpenSSL Windowsバイナリがサーバーにインストールされ、使用準備が整いました。あなたの希望により、OpenSSLの代わりに自己署名証明書を使用することも可能です。

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	6GB RAM	8+ GB RAM
ストレージ	76GB	90GB
Dockerバージョン	エンジン19+とコンポーズ1.24+	エンジン19+およびコンポーズ1.24+

ネストされた仮想化

Windows Server上でBitwardenを実行するには、ネストされた仮想化を使用する必要があります。あなたのハイパーバイザーのドキュメンテーションを確認して、ネストされた仮想化がサポートされているかどうか、そしてそれをどのように有効にするかを調べてください。

💡 Tip

Azure VMとしてWindows Serverを実行している場合、ネストされた仮想化をサポートするすべてのシステム要件を満たすStandard D2s v3 Virtual Machine running Windows Server 2022を推奨します。また、セキュリティタイプを選択する必要があります：スタンダードをデフォルトの信頼できる起動仮想マシンではなく。

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放して、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。

💡 Tip

Windows Firewallを使用している場合、Docker Desktop for Windowsは自動的にWindows Firewallの例外に自身を追加しません。関連するエラーを防ぐために、TCPポート80および443（または選択した代替ポート）の例外を追加してください。

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）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

PowerShellを開き、次のコマンドを実行してBitwardenローカルユーザーを作成します:

Bash

```
PS C:\> $Password = Read-Host -AsSecureString
```

上記のコマンドを実行した後、テキスト入力ダイアログに希望のパスワードを入力してください。パスワードを指定した後、次のコマンドを実行してください:

Bash

```
New-LocalUser "Bitwarden" -Password $Password -Description "Bitwarden Local Admin"
```

新しく作成されたユーザーとして、C:\の下にBitwardenフォルダーを作成してください。

Bash

```
PS C:\> mkdir Bitwarden
```

Docker Desktopをインストールしたら、**設定** → **リソース** → **ファイル共有**に移動し、作成したディレクトリ (`C:\Bitwarden`) をリソースリストに追加します。適用して再起動を選択して変更を適用してください。

このドキュメントのすべての後続の手順を完了する前に、新しく作成したユーザーとしてログインすることをお勧めします。

あなたのマシンを設定してください

あなたのBitwardenサーバーに必要なアセットを機器に設定するには：

💡 Tip

あなたがBitwardenユーザー&ディレクトリを作成した場合、以下をBitwardenユーザーとして完了してください。

1. `C:\Bitwarden`に新しいディレクトリを作成し、その名前を**`bwdata`**とし、そこに**`docker-stub.zip`**を解凍してください。

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](#)を参照してください。

💡 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](#)。

3. IDコンテナのための**`ID.pfx`**証明書を生成します。OpenSSLを使用するか、任意のツールを使用して自己署名証明書を生成することができます。あなたがOpenSSLを使用している場合、次のコマンドを実行してください：

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、`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. `ID.pfx`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、`.\bwddata\ID`）に移動します。
5. `identity.pfx`を`.\bwddata\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`.\bwddata\ssl`にサブディレクトリを作成してください。
7. 信頼できるSSL証明書と秘密鍵を新しく作成された`.\bwddata\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コンテナにマップされています。信頼できるSSL証明書を提供できない場合は、HTTPSエンドポイントをBitwardenクライアントアプリケーションに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. `.\bwddata\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`ヘッダー内も含めて。
 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実行してください：
 - 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. `.\bwddata\env\mysql.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。
10. `.\bwddata\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。

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：

- 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てください。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。
- 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
```

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

```
PS C:\Bitwarden> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
74ed54e04647	bitwarden/nginx:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	80/tcp, 0.0.0.0:80->8080/tcp, 0.0.0.0:443->8443/tcp	bitwarden-nginx
dd96a8387b94	bitwarden/admin:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	5000/tcp	bitwarden-admin
8260151e801d	bitwarden/portal:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	5000/tcp	bitwarden-portal
9e617bfa6f2e	bitwarden/sso:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	5000/tcp	bitwarden-sso
881371a30963	bitwarden/identity:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	5000/tcp	bitwarden-identity
25c66921ceb6	bitwarden/api:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	5000/tcp	bitwarden-api
bd904779cdf3	bitwarden/icons:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	5000/tcp	bitwarden-icons
f13f3ecc8d7b	bitwarden/mssql:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	5000/tcp	bitwarden-mssql
eaf9ea842f79	bitwarden/events:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	5000/tcp	bitwarden-events
860f5490b53f	bitwarden/web:2.23.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	5000/tcp	bitwarden-web
2772884733c6	bitwarden/notifications:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)	5000/tcp	bitwarden-notifications
fa6d2d05a582	bitwarden/attachments:1.43.0	"/entrypoint.sh"	36 minutes ago	Up 36 minutes (healthy)		bitwarden-attachments

健康的な容器的リスト

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、[組織を自己ホスト型で開始する](#)を参照してください。
- 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に記載されている通り。
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
```