

SELF-
HOSTING



INSTALLATIONS- &
BEREITSTELLUNGSANLEITUNGEN



Windows Offline- Bereitstellung

Ansicht im Hilfezentrum:

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

Windows Offline-Bereitstellung

Dieser Artikel führt Sie durch das Verfahren zur Installation und Bereitstellung von Bitwarden auf Ihrem eigenen Windows-Server in einer **offline oder luftdicht abgeschotteten** Umgebung. Bitte überprüfen Sie die Dokumentation zur [Software-Release-Unterstützung](#) von Bitwarden.

⚠ Warning

Manuelle Installationen sollten nur von fortgeschrittenen Benutzern durchgeführt werden. Fahren Sie nur fort, wenn Sie sehr vertraut mit Docker-Technologien sind und mehr Kontrolle über Ihre Bitwarden-Installation wünschen.

Manuelle Installationen haben nicht die Fähigkeit, bestimmte Abhängigkeiten der Bitwarden-Installation automatisch zu aktualisieren. Wenn Sie von einer Version von Bitwarden auf die nächste upgraden, sind Sie verantwortlich für Änderungen an erforderlichen Umgebungsvariablen, Änderungen an `nginx default.conf`, Änderungen an `docker-compose.yml` und so weiter.

Wir werden versuchen, diese in den [Veröffentlichungsnotizen auf GitHub](#) hervorzuheben. Sie können auch Änderungen an den [Abhängigkeitsvorlagen](#) überwachen, die vom Bitwarden-Installationskript auf GitHub verwendet werden.

Anforderungen

Bevor Sie mit der Installation fortfahren, stellen Sie bitte sicher, dass die folgenden Anforderungen erfüllt sind:

- [Docker Engine](#) und [Docker Compose](#) sind auf Ihrem Server installiert und einsatzbereit. Während dieser Einrichtung müssen Sie die Option **abwählen Verwenden Sie WSL2 anstelle von Hyper-V (empfohlen)**.
- Mit einem internetfähigen Gerät haben Sie die neueste `docker-stub.zip` Datei von der Releases-Seite des Bitwarden Server Repository heruntergeladen und diese Datei auf Ihren Server übertragen.
- Ein offline SMTP-Server ist in Ihrer Umgebung eingerichtet und aktiv.
- **(Optional)** [OpenSSL Windows Binärdateien](#) sind installiert und einsatzbereit auf Ihrem Server. Sie können ein selbstsigniertes Zertifikat anstelle von OpenSSL verwenden, wenn Sie möchten.

Systemspezifikationen

	Minimum	Empfohlen
Prozessor	x64, 1,4GHz	x64, 2GHz Dual Core
Erinnerung	6GB RAM	8+ GB RAM
Speicher	76GB	90GB
Docker-Version	Motor 19+ und Komponieren 1.24+	Motor 19+ und Komponieren 1.24+

Verschachtelte Virtualisierung

Bitwarden auf einem Windows-Server auszuführen erfordert die Verwendung von verschachtelter Virtualisierung. Bitte überprüfen Sie die Dokumentation Ihres Hypervisors, um herauszufinden, ob verschachtelte Virtualisierung unterstützt wird und wie Sie diese aktivieren können.

💡 Tip

Wenn Sie Windows Server als Azure VM ausführen, empfehlen wir eine **Standard D2s v3 Virtuelle Maschine mit Windows Server 2022**, die alle [Systemanforderungen](#) erfüllt, einschließlich Unterstützung für verschachtelte Virtualisierung. Sie müssen auch den **Sicherheitstyp** auswählen: **Standard** anstelle des Standard **Vertrauenswürdige Start virtuelle Maschinen**.

Installationsverfahren

Konfigurieren Sie Ihre Domain

Standardmäßig wird Bitwarden über die Ports 80 ([http](#)) und 443 ([https](#)) auf dem Host-Rechner bereitgestellt. Öffnen Sie diese Ports, damit auf Bitwarden sowohl innerhalb als auch außerhalb des Netzwerks zugegriffen werden kann. Sie können sich während der Installation für verschiedene Ports entscheiden.

💡 Tip

Wenn Sie die Windows-Firewall verwenden, wird Docker Desktop für Windows nicht automatisch eine Ausnahme für sich selbst in der Windows-Firewall hinzufügen. Fügen Sie Ausnahmen für TCP-Ports 80 und 443 (oder gewählte alternative Ports) hinzu, um damit verbundene Fehler zu verhindern.

Wir empfehlen die Konfiguration eines Domainnamens mit DNS-Einträgen, die auf Ihre Host-Maschine verweisen (zum Beispiel [bitwarden.example.com](#)), insbesondere wenn Sie Bitwarden über das Internet bereitstellen.

Erstellen Sie einen lokalen Bitwarden-Benutzer & Verzeichnis

Öffnen Sie PowerShell und erstellen Sie einen lokalen Bitwarden-Benutzer, indem Sie den folgenden Befehl ausführen:

Bash

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

Nach dem Ausführen des oben genannten Befehls geben Sie das gewünschte Passwort in das Texteingabefeld ein. Nachdem Sie ein Passwort festgelegt haben, führen Sie den folgenden Befehl aus:

Bash

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

Als neu erstellter Benutzer, erstellen Sie einen Bitwarden Ordner unter **C:**:

Bash

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

Sobald Sie Docker Desktop installiert haben, navigieren Sie zu **Einstellungen** → **Ressourcen** → **Dateifreigabe** und fügen Sie das erstellte Verzeichnis (**C:\Bitwarden**) zur Ressourcenliste hinzu. Wählen Sie **Anwenden & Neustarten**, um Ihre Änderungen zu übernehmen.

Wir empfehlen, sich als der neu erstellte Benutzer anzumelden, bevor alle nachfolgenden Verfahren in diesem Dokument abgeschlossen werden.

Konfigurieren Sie Ihre Maschine

Um Ihre Maschine mit den erforderlichen Ressourcen für Ihren Bitwarden-Server zu konfigurieren:

💡 Tip

Wenn Sie einen **Bitwarden-Benutzer & Verzeichnis** erstellt haben, führen Sie die folgenden Schritte als **Bitwarden-Benutzer** durch.

1. Erstellen Sie ein neues Verzeichnis in **C:\Bitwarden** mit dem Namen **bwdata** und extrahieren Sie **docker-stub.zip** dorthin.

Einmal entpackt, wird das Verzeichnis **bwdata** dem entsprechen, was die Volumenzuordnung der Datei **docker-compose.yml** erwartet. Sie können, wenn Sie möchten, den Standort dieser Zuordnungen auf dem Host-Computer ändern.

2. In **bwdata\env\global.override.env**, bearbeiten Sie die folgenden Umgebungsvariablen:

- **globalSettings__baseServiceUri__vault=**: Geben Sie die Domain Ihrer Bitwarden-Instanz ein.
- **globalSettings__sqlServer__ConnectionString=**: Ersetzen Sie das **RANDOM_DATABASE_PASSWORT** durch ein sicheres Passwort für die Verwendung in einem späteren Schritt.
- **globalSettings__identityServer__certificatePassword=**: Legen Sie ein sicheres Zertifikatspasswort für die Verwendung in einem späteren Schritt fest.
- **globalSettings__internalIdentityKey=**: Ersetzen Sie **RANDOM_IDENTITY_KEY** durch eine zufällige Schlüsselzeichenfolge.
- **globalSettings__oidcIdentityClientKey=**: Ersetzen Sie **RANDOM_IDENTITY_KEY** durch eine zufällige Schlüsselzeichenfolge.
- **globalSettings__duo__aKey=**: Ersetzen Sie **RANDOM_DUO_AKEY** durch eine zufällige Schlüsselzeichenfolge.
- **globalSettings__installation__id=**: Geben Sie eine Installations-ID ein, die Sie von <https://bitwarden.com/host> abgerufen haben.
- **globalSettings__installation__key=**: Geben Sie einen Installations-Schlüssel ein, den Sie von <https://bitwarden.com/host> abgerufen haben.
- **globalSettings__pushRelayBaseUri=**: Diese Variable sollte leer sein. Siehe [Konfiguration des Push-Relais](#) für weitere Informationen.

 **Tip**

Betrachten Sie zu diesem Zeitpunkt auch das Festlegen von Werten für alle `globalSettings__mail__smtp__` Variablen und für `adminSettings__admins`. Dies wird den SMTP-Mailserver konfigurieren, der verwendet wird, um Einladungen an neue Mitglieder der Organisation zu senden und den Zugang zum [Systemadministrator-Portal](#) bereitzustellen.

Erfahren Sie mehr über Umgebungsvariablen.

3. Generieren Sie ein `Identität.pfx` Zertifikat für den Identität Container. Sie können OpenSSL verwenden oder jedes beliebige Tool, um ein selbstsigniertes Zertifikat zu generieren. Wenn Sie OpenSSL verwenden, führen Sie die folgenden Befehle aus:

Bash

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

und

Bash

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

Ersetzen Sie in dem obigen Befehl `IDENTITY_CERT_PASSWORD` durch das Passwort des Zertifikats, das in **Schritt 2** erstellt und verwendet wurde.

4. Verschieben Sie `Identität.pfx` in das zugeordnete Volume-Verzeichnis (standardmäßig `.\bwdata\Identität`).
5. Kopieren Sie `identity.pfx` in das Verzeichnis `.\bwdata\ssl`.
6. Erstellen Sie ein Unterverzeichnis in `.\bwdata\ssl` mit dem Namen Ihrer Domain.
7. Stellen Sie ein vertrauenswürdiges SSL-Zertifikat und einen privaten Schlüssel im neu erstellten Unterverzeichnis `.\bwdata\ssl\bitwarden.example.com` bereit.

 **Note**

Dieses Verzeichnis ist dem NGINX-Container unter `\etc\ssl` zugeordnet. Wenn Sie kein vertrauenswürdiges SSL-Zertifikat bereitstellen können, stellen Sie die Installation mit einem Proxy vor, der den Bitwarden-Client-Anwendungen einen HTTPS-Endpunkt bietet.

8. In `.\bwdata\nginx\default.conf`:

1. Ersetzen Sie alle Instanzen von `bitwarden.example.com` durch Ihre Domain, einschließlich im `Content-Security-Policy` Kopfzeile.
2. Setzen Sie die Variablen `ssl_certificate` und `ssl_certificate_key` auf die Pfade des Zertifikats und des privaten Schlüssels, die in **Schritt 6** bereitgestellt wurden.

3. Führen Sie eine der folgenden Aktionen durch, abhängig von Ihrer Zertifikateinrichtung:

- Wenn Sie ein vertrauenswürdigen SSL-Zertifikat verwenden, setzen Sie die Variable `ssl_trusted_certificate` auf den Pfad zu Ihrem Zertifikat.
- Wenn Sie ein selbstsigniertes Zertifikat verwenden, kommentieren Sie die `ssl_trusted_certificate` Variablen aus.

9. In `.\bwdata\env\mysql.override.env`, ersetzen Sie `RANDOM_DATABASE_PASSWORD` mit dem Passwort, das in **Schritt 2** erstellt wurde.

10. In `.\bwdata\web\app-id.json`, ersetzen Sie `bitwarden.example.com` mit Ihrer Domain.

Bilder herunterladen & übertragen

Um Docker-Images für die Verwendung auf Ihrem Offline-Rechner zu erhalten:

1. Laden Sie von einer mit dem Internet verbundenen Maschine alle `bitwarden/xxx:latest` Docker-Bilder herunter, wie sie in der `docker-compose.yml` Datei in `docker-stub.zip` aufgelistet sind.

2. Speichern Sie jedes Bild in einer `.img` Datei, zum Beispiel:

Bash

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

3. Übertragen Sie alle `.img` Dateien auf Ihren Offline-Rechner.

4. Auf Ihrem Offline-Rechner laden Sie jede `.img` Datei, um Ihre lokalen Docker-Bilder zu erstellen, zum Beispiel:

Bash

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

Starten Sie Ihren Server

Starten Sie Ihren Bitwarden-Server mit dem folgenden Befehl:

Bash

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

Überprüfen Sie, ob alle Container korrekt laufen:

Bash

```
docker ps
```

```
PS C:\bitwarden> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS                    PORTS                                NAMES
7ded54e04647  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
dd96a838fb94  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
9e617bfe6f2e  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) 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) bitwarden-web
2772884733c6  bitwarden/notifications:1.43.0     "/entrypoint.sh"       36 minutes ago Up 36 minutes (healthy) 5000/tcp bitwarden-notifications
fa62d05a582  bitwarden/attachments:1.43.0       "/entrypoint.sh"       36 minutes ago Up 36 minutes (healthy) bitwarden-attachments
```

Liste der gesunden Behälter

Gratulation! Bitwarden läuft jetzt unter <https://your.domain.com>. Besuchen Sie den Web-Tresor in Ihrem Browser, um zu bestätigen, dass er funktioniert.

Sie können sich jetzt ein neues Konto registrieren und anmelden. Sie müssen SMTP-Umgebungsvariablen konfiguriert haben (siehe [Umgebungsvariablen](#)), um die E-Mail-Adresse für Ihr neues Konto zu verifizieren.

Nächste Schritte:

- Wenn Sie planen, eine Bitwarden Organisation selbst zu hosten, sehen Sie [eine Organisation selbst hosten](#) um zu beginnen.
- Für weitere Informationen siehe [FAQs zum selbst gehosteten](#).

Aktualisieren Sie Ihren Server

Die Aktualisierung eines selbst gehosteten Servers, der manuell installiert und bereitgestellt wurde, unterscheidet sich von dem [Standard-Aktualisierungsverfahren](#). Um Ihre manuell installierte Server-Aktualisierung durchzuführen:

1. Laden Sie das neueste [docker-stub.zip](#) Archiv von den [Veröffentlichungsseiten](#) auf GitHub herunter.
2. Entpacken Sie das neue [docker-stub.zip](#) Archiv und vergleichen Sie dessen Inhalt mit dem, was derzeit in Ihrem [bwdata](#) Verzeichnis ist, und kopieren Sie alles Neue in die bereits vorhandenen Dateien in [bwdata](#).
Überschreiben Sie Ihr bereits vorhandenes [bwdata](#)- Verzeichnis **nicht** mit dem Inhalt des neueren [docker-stub.zip](#)- Archivs, da dies alle von Ihnen durchgeführten benutzerdefinierten Konfigurationsarbeiten überschreiben würde.
3. Laden Sie die neuesten Containerbilder herunter und übertragen Sie sie auf Ihren Offline-Rechner [wie oben dokumentiert](#).
4. Führen Sie den folgenden Befehl aus, um Ihren Server mit Ihrer aktualisierten Konfiguration und den neuesten Containern neu zu starten:

Bash

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