

SELF-HOSTING > INSTALLATIONS- & BEREITSTELLUNGSANLEITUNGEN >

Linux Offline-Bereitstellung

Ansicht im Hilfezentrum:

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

Linux Offline-Bereitstellung

Dieser Artikel führt Sie durch das Verfahren zur Installation und Bereitstellung von Bitwarden auf Ihrem eigenen 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.
- Mit einem internetfähigen Gerät haben Sie die neueste `docker-stub.zip` Datei aus dem Bitwarden Server Repositorys [Releases-Seite](#) heruntergeladen und diese Datei auf Ihren Server übertragen.
- Ein offline SMTP-Server ist in Ihrer Umgebung eingerichtet und aktiv.

📌 Note

Bitwarden wird derzeit nur in einer Docker- und Docker Compose-Umgebung unterstützt. Die Installationsskripte von Bitwarden und manuelle Installationsartefakte lassen sich nicht genau in Kubernetes-Manifeste umwandeln, ohne umfangreiches Wissen über den Bitwarden-Stack und Kubernetes. Automatische Konvertierungen der Installationsartefakte werden derzeit nicht empfohlen und könnten zu einer defekten Bereitstellungsumgebung führen.

Systemspezifikationen

	Minimum	Empfohlen
Prozessor	x64, 1,4GHz	x64, 2GHz Dual-Core
Erinnerung	2GB RAM	4GB RAM
Speicher	12GB	25GB

	Minimum	Empfohlen
Docker-Version	Motor 19+ und Komponieren 1.24+	Motor 19+ und Komponieren 1.24+

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.

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

Wir empfehlen, Ihren Server mit einem dedizierten Bitwarden-Dienstkonto zu konfigurieren, von dem aus Sie Bitwarden installieren und ausführen können. Wenn Sie dies tun, wird Ihre Bitwarden-Instanz von anderen Anwendungen auf Ihrem Server isoliert.

Diese Schritte sind von Bitwarden empfohlene Best Practices, sind jedoch nicht erforderlich. Für weitere Informationen, siehe Docker's [Nachinstallations-Schritte für Linux](#) Dokumentation.

1. Erstellen Sie einen Bitwarden-Benutzer:

Bash

```
sudo adduser bitwarden
```

2. Legen Sie ein Passwort für den Bitwarden-Benutzer fest:

Bash

```
sudo passwd bitwarden
```

3. Erstellen Sie eine Docker-Gruppe (falls diese noch nicht existiert):

Bash

```
sudo groupadd docker
```

4. Fügen Sie den Bitwarden-Benutzer zur Docker-Gruppe hinzu:

Bash

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

5. Erstellen Sie ein Bitwarden-Verzeichnis:

```
Bash
```

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

6. Setzen Sie die Berechtigung für das `/opt/Bitwarden` Verzeichnis:

```
Bash
```

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

7. Setzen Sie den Bitwarden-Benutzerbesitz des `/opt/bitwarden` Verzeichnisses:

```
Bash
```

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

Konfigurieren Sie Ihre Maschine

⚠ Warning

Wenn Sie einen Bitwarden-Benutzer & Verzeichnis erstellt haben, führen Sie die folgenden Schritte als Bitwarden-Benutzer aus dem `/opt/bitwarden` Verzeichnis aus. **Installieren Sie Bitwarden nicht als Root**, da Sie während der Installation auf Probleme stoßen werden.

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

1. Erstellen Sie ein neues Verzeichnis namens `bwdata` und extrahieren Sie `docker-stub.zip` darin, zum Beispiel:

```
Bash
```

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

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 `ZUFÄLLIGES_DATENBANK_PASSWORT` durch ein sicheres Passwort, das Sie in einem späteren Schritt verwenden werden.
- `globalSettings__identityServer__certificatePassword`: Legen Sie ein sicheres Zertifikat-Passwort 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 [Konfigurieren Sie Push-Relay](#) 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. Aus `./bwdata`, generieren Sie eine `.pfx` Zertifikatsdatei für den Identität-Container und verschieben Sie diese in das zugeordnete Volumenverzeichnis (standardmäßig, `./bwdata/identity/`). Zum Beispiel, 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. Kopieren Sie `identity.pfx` in das Verzeichnis `./bwdata/ssl`.
5. Erstellen Sie ein Unterverzeichnis in `./bwdata/ssl`, benannt nach Ihrer Domain, zum Beispiel:

```
Bash
```

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

6. 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 bereitstellt.

7. In `./bwdata/nginx/default.conf`:

1. Ersetzen Sie alle Instanzen von `bitwarden.example.com` durch Ihre Domain, einschließlich im `Content-Security-Richtlinien` Header.
2. Setzen Sie die Variablen `ssl_certificate` und `ssl_certificate_key` auf die Pfade des Zertifikats und des privaten Schlüssels, die in **Schritt 6** angegeben sind.
3. Führen Sie eine der folgenden Aktionen durch, abhängig von Ihrer Zertifikateinrichtung:
 - Wenn Sie ein vertrauenswürdiges 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 Variable `ssl_trusted_certificate` aus.

8. In `./bwdata/env/mssql.override.env`, ersetzen Sie `RANDOM_DATABASE_PASSWORD` mit dem Passwort, das in **Schritt 2** erstellt wurde.

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

10. In `./bwdata/env/uid.env`, setzen Sie die UID und GID der Bitwarden Benutzer und Gruppe, die Sie früher erstellt haben, damit die Container unter ihnen laufen, zum Beispiel:

```
Bash
```

```
LOCAL_UID=1001
```

```
LOCAL_GID=1001
```

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 Datei `docker-compose.yml` in `docker-stub.zip` aufgelistet sind.
2. Speichern Sie jedes Bild in einer `.img` Datei, zum Beispiel:

Bash

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

```
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
406adf1a6c5c   bitwarden/sso:1.38.2               "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp bitwarden-sso
9e0e8cb75b29   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
feca5a34b02c   bitwarden/icons:1.38.2             "/entrypoint.sh"        3 minutes ago Up 3 minutes (healthy)   5000/tcp bitwarden-icons
bitwarden@bitwarden:/opt/bitwarden$
```

Healthy

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
```