

SELF-HOSTING > INSTALL & DEPLOY GUIDES >

Unified Deployment beta

View in the help center:

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

Unified Deployment beta

Note

This solution is in beta and intended for personal use. Business plans should use the officially-supported, standard deployment option.

While the Bitwarden unified self-hosted deployment is in beta, those installing unified **should not** setup automatic upgrade procedures that pull the latest images available. Bitwarden recommends allowing some time for stabilization of a release before upgrading.

[Learn how to report issues.](#)

This article will walk you through installing and launching the [Bitwarden unified self-hosted deployment](#). Use this deployment method to:

- Simplify configuration and optimize resource usage (CPU, memory) by deploying Bitwarden with a single Docker image.
- Utilize different database solutions such as MSSQL, PostgreSQL, SQLite, and MySQL/MariaDB.
- Run on ARM architecture for alternative systems such as Raspberry Pi and NAS servers.

System requirements

Bitwarden unified deployment requires:

- At least 200 MB RAM
- Storage 1GB
- Docker Engine 26+

Install Docker

The unified deployment will run on your machine using a [Docker container](#). The unified deployment can be run with any Docker edition or plan. Evaluate which edition is best for your installation.

Install Docker on your machine before proceeding with installation. Refer to the following Docker documentation for help:

- [Install Docker Engine](#)

Run Bitwarden unified

The unified deployment can be run using the `docker run` command (see [here](#)) or using Docker Compose (see [here](#)). In either case, you'll need to specify environment variables for the container.

Quick start guide

Use `docker run` to launch Bitwarden on a Raspberry Pi:

Install and deploy the Bitwarden unified self-hosting option

<https://player.vimeo.com/video/799236723>

Specify environment variables

Running the unified deployment will require environment variables to be set for the container. Environment variables can be specified by creating a `settings.env` file, which you can find an example of in our [GitHub](#) repository, or by using the `--env` flag if you're using the `docker run` method. Several optional variables are available for use for a more personalized unified deployment experience. Additional details on these variables can be located [here](#).

At a minimum, set values for the variables that fall under the `# Required Settings #` section of the example `.env` file:

| Variable | Description |
|----------------|---|
| BW_DOMAIN | Replace <code>bitwarden.yourdomain.com</code> with the domain where Bitwarden will be accessed. |
| BW_DB_PROVIDER | The database provider you will be using for your Bitwarden server. Available options are <code>sqlserver</code> , <code>postgres</code> , <code>sqlite</code> , or <code>mysql/mariadb</code> . |
| BW_DB_SERVER | The name of the server on which your database is running. |

| Variable | Description |
|---------------------|---|
| BW_DB_DATABASE | The name of your Bitwarden database. |
| BW_DB_USERNAME | The username for accessing the Bitwarden database. |
| BW_DB_PASSWORD | The password for accessing the Bitwarden database. |
| BW_DB_FILE | Only required for <code>sqlite</code> if you would like to specify the path to your database file. If not specified, <code>sqlite</code> will automatically create a <code>vault.db</code> file under the <code>/etc/bitwarden</code> volume. |
| BW_INSTALLATION_ID | A valid installation ID generated from https://bitwarden.com/host/ . |
| BW_INSTALLATION_KEY | A valid installation key generated from https://bitwarden.com/host/ . |

Note

Unlike the Bitwarden standard deployment, unified deployment does not come out-of-the-box with a database. You can use an existing database, or create a new one as documented in [this example](#), and in both cases you must enter valid information in the `BW_DB_...` variables documented here.

Using non-MSSQL database providers may result in performance issues, as support for these platforms continues to be worked on throughout the beta. Please use [this issue template](#) to report anything related to your Bitwarden unified deployment and check out [this page](#) to track known issues or join the discussion.

Using docker run

The unified deployment can be run with the `docker run` command, as in the following example:

Bash

```
docker run -d --name bitwarden -v /$(pwd)/bwdata:/etc/bitwarden -p 80:8080 --env-file settings.env bitwarden/self-host:beta
```

The command featured above has several required options for the `docker run` command, including:

| Name, shorthand | Description |
|----------------------------|---|
| <code>--detach, -d</code> | Run the container in the background and print container ID. |
| <code>--name</code> | Provide a name for the container. <code>bitwarden</code> is used in the example. |
| <code>--volume, -v</code> | Bind mount a volume. At a minimum, mount <code>/etc/bitwarden</code> . |
| <code>--publish, -p</code> | Map container ports to the host. The example shows the port <code>80:8080</code> mapped. Port 8443 is required when configuring SSL. |
| <code>--env-file</code> | Path of the file to read environment variables from. Alternatively, use the <code>--env</code> flag to declare environment variables inline (learn more). |

Once you run the command, verify that the container is running and healthy with:

Bash

```
docker ps
```

Congratulations! Your unified deployment is now up and running at <https://your.domain.com>. Visit the web vault in your browser to confirm that it's working. You may now register a new account and log in.

Using Docker Compose

Running the unified deployment with Docker Compose will require Docker Compose version 1.24+. To run the unified deployment with Docker compose, create a `docker-compose.yml` file, for example:

Bash

```
---
version: "3.8"

services:
  bitwarden:
    depends_on:
      - db
    env_file:
      - settings.env
    image: bitwarden/self-host:beta
    restart: always
    ports:
      - "80:8080"
    volumes:
      - bitwarden:/etc/bitwarden

  db:
    environment:
      MARIADB_USER: "bitwarden"
      MARIADB_PASSWORD: "super_strong_password"
      MARIADB_DATABASE: "bitwarden_vault"
      MARIADB_RANDOM_ROOT_PASSWORD: "true"
    image: mariadb:10
    restart: always
    volumes:
      - data:/var/lib/mysql

volumes:
  bitwarden:
  data:
```

In the `docker-compose.yml` file, make any desired configurations including:

- Mapping volumes for logs and Bitwarden data.
- Mapping ports.

- [Configuring a database image.](#)^a

^aOnly setup a database in `docker-compose.yml`, as in the above example, if you want to **create a new database server** to use with Bitwarden. Sample configurations for MySQL, MSSQL, and PostgreSQL are included in our [example file](#).

Once your `docker-compose.yml` and `settings.env` file are created, start your unified server by running:

Bash

```
docker compose up -d
```

Verify that all containers are running correctly:

Bash

```
docker ps
```

Congratulations! Your unified deployment is now up and running at <https://your.domain.com>. Visit the web vault in your browser to confirm that it's working. You may now register a new account and log in.

Update your server

To update your unified deployment:

⇒ Docker run update

1. Stop the running Docker container:

Bash

```
docker stop bitwarden
```

2. Remove the Docker container:

Bash

```
docker rm bitwarden
```

3. Run the following command to pull the most recent Bitwarden unified image:

Bash

```
docker pull bitwarden/self-host:beta
```

4. Run the Docker container again:

Bash

```
docker run -d --name bitwarden -v /$(pwd)/bwdata:/etc/bitwarden -p 80:8080 --env-file settings.  
env bitwarden/self-host:beta
```

⇒ Docker Compose update

1. Stop the running Docker container:

Bash

```
docker compose down
```

2. Run the following command to pull the most recent Bitwarden unified image:

Bash

```
docker compose pull
```

3. Recreate any containers that need to be updated:

Bash

```
docker compose up -d
```

4. Verify that the containers are running:

Bash

```
docker compose ps
```

Environment variables

The unified deployment will operate by default without several of the standard Bitwarden services. This allows for increased customization and optimization of your unified deployment. Configure these services, and more optional settings, by editing various environment variables.

Note

Whenever you change an environment variable, the Docker container will need to be recreated. Learn more [here](#).

Webserver ports

| Variable | Description |
|---------------|---|
| BW_PORT_HTTP | Change the port used for HTTP traffic. By default, 8080 . |
| BW_PORT_HTTPS | Change the port used for HTTPS traffic. By default, 8443 . |

SSL

Use these values to change certificate settings.

| Variable | Description |
|------------------|--|
| BW_ENABLE_SSL | Use SSL/TLS. true/false . Default false . SSL is required for Bitwarden to function properly. If you are not using SSL configured in the Bitwarden container you should front Bitwarden with a SSL proxy. |
| BW_SSL_CERT | The name of your SSL certificate file. The file must be located in the /etc/bitwarden directory within the container. Default ssl.crt . |
| BW_SSL_KEY | The name of your SSL key file. The file must be located in the /etc/bitwarden directory within the container. Default ssl.key . |
| BW_ENABLE_SSL_CA | Use SSL with certificate authority(CA) backed service. true/false . Default false . |
| BW_SSL_CA_CERT | The name of your SSL CA certificate. The file must be located in the /etc/bitwarden directory within the container. Default ca.crt . |
| BW_ENABLE_SSL_DH | Use SSL with Diffie-Hellman key exchange. true/false . Default false . |

| Variable | Description |
|------------------|--|
| BW_SSL_DH_CERT | The name of your Diffie–Hellman parameters file. The file must be located in the <code>/etc/bitwarden</code> directory within the container. Default <code>dh.pem</code> . |
| BW_SSL_PROTOCOLS | SSL version used by NGINX. Leave empty for recommended default. Learn more. |
| BW_SSL_CIPHERS | SSL ciphersuites used by NGINX. Leave empty for recommended default. Learn more. |

Note

If you are using an existing SSL certificate, you will have to enable the appropriate SSL options in `settings.env`. SSL files must be stored in `/etc/bitwarden`, which can be referenced in the `docker-compose.yml` file. These files must match the names configured in `settings.env`.

The default behavior is to generate a self–signed certificate if SSL is enabled and no existing certificate files are in the expected location (`/etc/bitwarden`).

Services

Additional services can be enabled or disabled for specific use cases, such as enterprise or team needs, by changing the following values:

| Variable | Description |
|------------------|---|
| BW_ENABLE_ADMIN | Do not disable this service. Learn more about Admin panel capabilities here . Default <code>true</code> . |
| BW_ENABLE_API | Do not disable this service. Default <code>true</code> . |
| BW_ENABLE_EVENTS | Enable or disable Bitwarden events logs for teams and enterprise event monitoring. Default <code>false</code> . |
| BW_ENABLE_ICONS | Enable or disable Bitwarden brand icons that are set with the login item URI's. Learn more here . Default <code>true</code> . |

| Variable | Description |
|-------------------------|---|
| BW_ENABLE_IDENTITY | Do not disable this service. Default true . |
| BW_ENABLE_NOTIFICATIONS | Enable or disable notification services for receiving push notifications to mobile devices, using login with device, mobile vault sync, and more. Default true . |
| BW_ENABLE_SCIM | Enable or disable SCIM for Enterprise organizations. Default false . |
| BW_ENABLE_SSO | Enable or disable SSO services for Enterprise organizations. Default false . |
| BW_ICONS_PROXY_TO_CLOUD | Enabling this service will proxy icon service requests to operate through cloud services in order to lower system memory load. If choosing to use this setting, BW_ENABLE_ICONS should be set to false in order to reduce container load. Default false . |

Mail

Configure SMTP settings for your unified deployment. Copy information from your chosen mail SMTP provider into the following fields:

| Variable | Description |
|------------------------------------|--|
| globalSettings__mail__replyToEmail | Enter the reply email for your server. |
| globalSettings__mail__smtp__host | Enter host domain for your SMTP server. |
| globalSettings__mail__smtp__port | Enter the port number from the SMTP host. |
| globalSettings__mail__smtp__ssl | If your SMTP host uses SSL enter true . Set value to false if your host uses TLS service. |

| Variable | Description |
|---|--------------------------|
| <code>globalSettings__mail__smtp__username</code> | Enter the SMTP username. |
| <code>globalSettings__mail__smtp__password</code> | Enter the SMTP password. |

Yubico API (YubiKey)

| Variable | Description |
|---|---|
| <code>globalSettings__yubico__clientId</code> | Replace value with ID received from your Yubico Key. Sign up for Yubico Key here . |
| <code>globalSettings__yubico__key</code> | Input the key value received from Yubico. |

Database configurations

Utilizing the variety of database options that are compatible with the unified deployment will require additional `.env` configurations.

⇒MySQL/MariaDB

In `settings.env`:

Bash

```
# Database
BW_DB_PROVIDER=mysql
BW_DB_SERVER=db
BW_DB_DATABASE=bitwarden_vault
BW_DB_USERNAME=bitwarden
BW_DB_PASSWORD=super_strong_password
```

⇒MSSQL

In `settings.env`:

Bash

```
# Database
BW_DB_PROVIDER=sqlserver
BW_DB_SERVER=db
BW_DB_DATABASE=bitwarden_vault
BW_DB_USERNAME=bitwarden
BW_DB_PASSWORD=super_strong_password
```

⇒ SQLite

In `settings.env`:

Bash

```
# Database
BW_DB_PROVIDER=sqlite
BW_DB_FILE=/path/to/.db
```

Assigning the `sqlite` value will create a `vault.db` file in the `/etc/bitwarden` volume automatically. `BW_DB_FILE` is only required if you would like to specify the path to a different database file.

⇒ PostgreSQL

In `settings.env`:

Bash

```
# Database
BW_DB_PROVIDER=postgresql
BW_DB_SERVER=db
BW_DB_DATABASE=bitwarden_vault
BW_DB_USERNAME=bitwarden
BW_DB_PASSWORD=super_strong_password
```

Other

| Variable | Description |
|--|---|
| <code>globalSettings__disableUserRegistration</code> | Enable or disable user account registration capabilities. |

| Variable | Description |
|----------------------------|--|
| globalSettings__hibpApiKey | Enter the API key provided by Have I Been Pwnd. Register to receive the API key here . |
| adminSettings__admins | Enter admin email addresses. |
| BW_REAL_IPS | Define real IPs in <code>nginx.conf</code> in a comma separated list. Useful for defining proxy servers that forward the client IP address. Learn more . |
| BW_CSP | Content-Security-Policy parameter. Reconfiguring this parameter may break features. By changing this parameter, you become responsible for maintaining this value. |
| BW_DB_PORT | Specify a custom port for database traffic. If unspecified, the default will depend on your chosen database provider. |

Restart the container

To restart your Docker container after changing environment variables, run the following commands from the Bitwarden unified deployment directory:

⇒ Docker run

1. Stop the running Docker container:

```
Bash
```

```
docker stop bitwarden
```

2. Remove the Docker container:

```
Bash
```

```
docker rm bitwarden
```

3. Run the Docker container again:

Bash

```
docker run -d --name bitwarden -v /$(pwd)/bwdata/:/etc/bitwarden -p 80:8080 --env-file settings.env bitwarden/self-host:beta
```

⇒ Docker Compose

1. Stop the running Docker container:

Bash

```
docker compose down
```

2. Recreate the containers:

Bash

```
docker compose up -d
```

3. Ensure that the containers are running properly with:

Bash

```
docker compose ps
```

Memory usage

By default, the Bitwarden container will consume memory that is available to it, often being more than the minimum needed to run. For memory conscious environments, you can use `docker -m` or `--memory=` to limit the Bitwarden container's memory usage.

| Name, shorthand | Description |
|--|---|
| <code>--memory=</code> , <code>-m</code> | The maximum amount of memory the container can use. Bitwarden requires at least 200m. See the Docker documentation to learn more. |

To control memory usage with Docker Compose, use the `mem_limit` key:

Bash

```
services:  
  bitwarden:  
    env_file:  
      - settings.env  
    image: bitwarden/self-host:beta  
    restart: always  
    mem_limit: 200m
```

Reporting issues

While the Bitwarden unified deployment remains in beta release, we encourage you to report issues and give feedback via GitHub. Please use [this issue template](#) to report anything related to your Bitwarden unified deployment and check out [this page](#) to track known issues or join the discussion.

Additional resources

- If you are planning to self-host a Bitwarden organization, see [self-host an organization](#) to get started.

For more information on Bitwarden's standard self-hosted deployment see:

- [Install and Deploy - Linux](#)
- [Install and Deploy - Windows](#)
- [Install and Deploy - Manual](#)