SELF-HOSTING > INSTALAR & DESPLEGAR GUÍAS >

Despliegue Unificado beta

Despliegue Unificado beta

> **Note**
>
> Esta solución está en beta y está destinada para uso personal. Los planes de negocio deben utilizar la opción de implementación estándar oficialmente respaldada.
>
> Mientras que la implementación unificada autoalojada de Bitwarden está en beta, aquellos que instalan unificada **no deberían** configurar procedimientos de mejora automáticos que extraigan las últimas imágenes disponibles. Bitwarden recomienda permitir algún tiempo para la estabilización de una versión antes de mejorar.
>
> Aprenda cómo reportar problemas .

Este artículo te guiará a través de la instalación y el lanzamiento de la implementación unificada y autoalojada de Bitwarden. Utilice este método de implementación para:

- Simplifique la configuración y optimice el uso de recursos (CPU, memoria) desplegando Bitwarden con una sola imagen de Docker.
- Utilice diferentes soluciones de base de datos como MSSQL, PostgreSQL, SQLite y MySQL/MariaDB.
- Ejecutar en la arquitectura ARM para sistemas alternativos como Raspberry Pi y servidores NAS.

Requisitos del sistema

La implementación unificada de Bitwarden requiere:

- Al menos 200 MB de RAM
- Almacenamiento 1GB
- Motor Docker 19+

Instalar Docker

La implementación unificada se ejecutará en su máquina utilizando un contenedor Docker. La implementación unificada se puede ejecutar con cualquier edición o plan de Docker. Evalúa qué edición es la mejor para tu instalación.

Instala Docker en tu máquina antes de proceder con la instalación. Consulte la siguiente documentación de Docker para obtener ayuda:

- Instala Docker Engine

Ejecuta Bitwarden unificado

El despliegue unificado se puede ejecutar utilizando el comando `docker run` (ver aquí) o utilizando Docker Compose (ver aquí). En cualquier caso, necesitarás especificar variables de entorno para el contenedor.

Especificar variables de entorno

La ejecución de la implementación unificada requerirá que se establezcan variables de entorno para el contenedor. Las variables de entorno se pueden especificar creando un archivo `settings.env`, del cual puedes encontrar un ejemplo en nuestro repositorio de GitHub, o utilizando la bandera `--env` si estás utilizando el método `docker run`. Varias variables opcionales están disponibles para su uso para una experiencia de implementación unificada más personalizada. Se pueden encontrar detalles adicionales sobre estas variables aquí.

Como mínimo, establezca valores para las variables que caen bajo la sección `# Ajustes Requeridos #` del ejemplo de archivo `.env`:

Variable	Descripción
DOMINIO_BW	Reemplace `bitwarden.sudominio.com` con el dominio donde se accederá a Bitwarden.
PROVEEDOR_BW_DB	El proveedor de base de datos que utilizarás para tu servidor Bitwarden. Las opciones disponibles son `sqlserver`, `postgresql`, `sqlite`, o `mysql`/`mariadb`.
SERVIDOR_BW_DB	El nombre del servidor en el que se está ejecutando su base de datos.
BW_DB_BASE_DE_DATOS	El nombre de tu base de datos de Bitwarden.
BW_DB_NOMBREDEUSUARIO	El nombre de usuario para acceder a la base de datos de Bitwarden.
CONTRASEÑA_BW_DB	La contraseña para acceder a la base de datos de Bitwarden.
ARCHIVO_BW_DB	Solo se requiere para `sqlite` si desea especificar la ruta al archivo de su base de datos. Si no se especifica, `sqlite` creará automáticamente un archivo `vault.db` en el volumen `/etc/bitwarden`.
ID_DE_INSTALACIÓN_BW	Una ID de instalación válida generada desde https://bitwarden.com/host/.
CLAVE_DE_INSTALACIÓN_BW	Una clave de instalación válida generada desde https://bitwarden.com/host/.

> **Note**
>
> A diferencia de la implementación estándar de Bitwarden, la implementación unificada no viene con una base de datos incluida. Puedes usar una base de datos existente, o crear una nueva como se documenta en este ejemplo, y en ambos casos debes ingresar información válida en las variables `BW_DB_...` documentadas aquí.
>
> El uso de proveedores de bases de datos que no sean MSSQL puede resultar en problemas de rendimiento, ya que el soporte para estas plataformas continúa trabajándose a lo largo de la beta. Por favor, use esta plantilla de problema para informar cualquier cosa relacionada con su despliegue unificado de Bitwarden y consulte esta página para seguir los problemas conocidos o unirse a la discusión.

Usando docker run

El despliegue unificado se puede ejecutar con el comando `docker run`, como en el siguiente ejemplo:

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

El comando destacado anteriormente tiene varias opciones requeridas para el comando `docker run`, incluyendo:

Nombre, abreviatura	Descripción
--desconectar , -d	Ejecuta el contenedor en segundo plano e imprime la ID del contenedor.
--nombre	Proporcione un nombre para el contenedor. `Bitwarden` se usa en el ejemplo.
--volumen , -v	Montar un volumen con bind. Como mínimo, monta `/etc/bitwarden`.
--publicar , -p	Mapear los puertos del contenedor al host. El ejemplo muestra el puerto `80:8080` mapeado. El puerto 8443 es necesario al configurar SSL.
--archivo-env	Ruta del archivo para leer las variables de entorno. Alternativamente, use la bandera `--env` para declarar variables de entorno en línea (aprende más).

Una vez que ejecute el comando, verifique que el contenedor está funcionando y está saludable con:

```Bash
docker ps
```

¡Felicidades! Su implementación unificada ya está en funcionamiento en `https://your.domain.com`. Visita la caja fuerte web en tu navegador para confirmar que está funcionando. Ahora puedes registrar una nueva cuenta e iniciar sesión.

Usando Docker Compose

Ejecutar la implementación unificada con Docker Compose requerirá la versión 1.24+ de Docker Compose. Para ejecutar la implementación unificada con Docker compose, crea un archivo `docker-compose.yml`, por ejemplo:

```bash
---
version: "3.8"

services:
  bitwarden:
    depends_on:
      - db
    env_file:
      - settings.env
    image: ghcr.io/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:
```

En el archivo `docker-compose.yml`, realice cualquier configuración deseada, incluyendo:

- Mapeando volúmenes para registros y datos de Bitwarden.
- Mapeando puertos.

- Configurar una imagen de base de datos. [a]

[a]Solo configure una base de datos en `docker-compose.yml`, como en el ejemplo anterior, si desea **crear un nuevo servidor de base de datos** para usar con Bitwarden. Las configuraciones de muestra para MySQL, MSSQL y PostgreSQL están incluidas en nuestro archivo de ejemplo.

Una vez que se han creado su `docker-compose.yml` y el archivo `settings.env`, inicie su servidor unificado ejecutando:

```
Bash

docker compose up -d
```

Verifique que todos los contenedores estén funcionando correctamente.

```
Bash

docker ps
```

¡Felicidades! Su implementación unificada ya está en funcionamiento en `https://your.domain.com`. Visita la caja fuerte web en tu navegador para confirmar que está funcionando. Ahora puedes registrar una nueva cuenta e iniciar sesión.

Actualiza tu servidor

Para actualizar tu despliegue unificado:

⇒Docker ejecutar actualizar

1. Detén el contenedor Docker en ejecución:

   ```
   Bash

   docker stop bitwarden
   ```

2. Elimina el contenedor Docker:

   ```
   Bash

   docker rm bitwarden
   ```

3. Ejecute el siguiente comando para extraer la imagen unificada más reciente de Bitwarden:

```Bash
docker pull ghcr.io/bitwarden/self-host:beta
```

4. Ejecuta el contenedor Docker de nuevo:

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

⇒Actualización de Docker Compose

1. Detén el contenedor Docker en ejecución:

```Bash
docker compose down
```

2. Ejecute el siguiente comando para extraer la imagen unificada más reciente de Bitwarden:

```Bash
docker compose pull
```

3. Recrea cualquier contenedor que necesite ser actualizado:

```Bash
docker compose up -d
```

4. Verifica que los contenedores estén funcionando:

```Bash
docker compose ps
```

Variables de entorno

La implementación unificada funcionará por defecto sin varios de los servicios estándar de Bitwarden. Esto permite una mayor personalización y optimización de su implementación unificada. Configure estos servicios, y más ajustes opcionales, editando varias variables de entorno.

> **Note**
> Siempre que cambies una variable de entorno, el contenedor Docker necesitará ser recreado. Aprende más aquí.

Puertos del servidor web

Variable	Descripción
BW_PORT_HTTP	Cambia el puerto utilizado para el tráfico HTTP. Por defecto, `8080`.
BW_PORT_HTTPS	Cambia el puerto utilizado para el tráfico HTTPS. Por defecto, `8443`.

SSL

Utilice estos valores para cambiar los ajustes del certificado.

Variable	Descripción
HABILITAR_BW_SSL	Usa SSL/TLS. `verdadero`/`falso`. Predeterminado `falso`. Se requiere SSL para que Bitwarden funcione correctamente. Si no está utilizando SSL configurado en el contenedor de Bitwarden, debería poner Bitwarden al frente con un proxy SSL.
CERTIFICADO_BW_SSL	El nombre de tu archivo de certificado SSL. El archivo debe estar ubicado en el directorio `/etc/bitwarden` dentro del contenedor. Predeterminado `ssl.crt`.
BW_SSL_KEY	El nombre de tu archivo de clave SSL. El archivo debe estar ubicado en el directorio `/etc/bitwarden` dentro del contenedor. Predeterminado `ssl.key`.
BW_ENABLE_SSL_CA	Use SSL con servicio respaldado por autoridad de certificación (CA). `verdadero`/`falso`. Por defecto `falso`.

Variable	Descripción
BW_SSL_CA_CERT	El nombre de tu certificado CA SSL. El archivo debe estar ubicado en el directorio `/etc/bitwarden` dentro del contenedor. Predeterminado `ca.crt`.
BW_ENABLE_SSL_DH	Usa SSL con intercambio de claves Diffie-Hellman. `verdadero`/`falso`. Predeterminado `falso`.
BW_SSL_DH_CERT	El nombre de tu archivo de parámetros Diffie-Hellman. El archivo debe estar ubicado en el directorio `/etc/bitwarden` dentro del contenedor. Por defecto `dh.pem`.
BW_SSL_PROTOCOLS	Versión de SSL utilizada por NGINX. Dejar vacío para la opción predeterminada recomendada. Más información.
BW_SSL_CIPHERS	Cifrados SSL utilizados por NGINX. Dejar vacío para la opción predeterminada recomendada. Más información.

Note

Si está utilizando un certificado SSL existente, tendrá que habilitar las opciones SSL apropiadas en `settings.env`. Los archivos SSL deben almacenarse en `/etc/bitwarden`, que se puede referenciar en el archivo `docker-compose.yml`. Estos archivos deben coincidir con los nombres configurados en `settings.env`.

El comportamiento predeterminado es generar un certificado autofirmado si SSL está habilitado y no hay archivos de certificado existentes en la ubicación esperada (`/etc/bitwarden`).

Servicios

Los servicios adicionales pueden habilitarse o deshabilitarse para casos de uso específicos, como las necesidades de la empresa o de los equipos, cambiando los siguientes valores:

Variable	Descripción
HABILITAR_ADMIN	**No desactive este servicio.** Aprende más sobre las capacidades del panel de administrador aquí. Predeterminado `verdadero`.

Variable	Descripción
HABILITAR_API	**No desactive este servicio.** Predeterminado `verdadero`.
HABILITAR_EVENTOS	Habilitar o deshabilitar los registros de eventos de Bitwarden para los equipos y el monitoreo de eventos de la empresa. Predeterminado `falso`.
HABILITAR_ICONOS	Habilitar o deshabilitar los iconos de marca de Bitwarden que se establecen con los URI de inicio de sesión del elemento. Aprende más aquí. Predeterminado `verdadero`.
HABILITAR_IDENTIDAD	**No desactive este servicio.** Predeterminado `verdadero`.
HABILITAR_NOTIFICACIONES	Habilite o deshabilite los servicios de notificación para recibir notificaciones push en dispositivos móviles, utilizando el inicio de sesión con dispositivo, sincronización de caja fuerte móvil, y más. Predeterminado `verdadero`.
HABILITAR_BW_SCIM	Habilitar o deshabilitar SCIM para organizaciones de Empresa. Predeterminado `falso`.
HABILITAR_BW_SSO	Habilitar o deshabilitar servicios SSO para organizaciones de Empresa. Predeterminado `falso`.
BW_ICONOS_PROXY_A_LA_NUBE	Habilitar este servicio hará que las solicitudes de servicio de iconos se realicen a través de servicios en la nube para reducir la carga de memoria del sistema. Si elige usar este ajuste, `BW_ENABLE_ICONS` debe establecerse en `falso` para reducir la carga del contenedor. Predeterminado `falso`.

Correo

Configure los ajustes SMTP para su despliegue unificado. Copia la información de tu proveedor de correo SMTP elegido en los siguientes campos:

Variable	Descripción
globalSettings__mail__responderAEmail	Ingrese el correo electrónico de respuesta para su servidor.
configuracionesGlobales__correo__smtp__anfitrión	Ingrese el dominio del host para su servidor SMTP.
globalSettings__mail__smtp__puerto	Ingrese el número de puerto del host SMTP.
configuracionesGlobales__correo__smtp__ssl	Si tu host SMTP usa SSL, ingresa `verdadero`. Establezca el valor en `falso` si su anfitrión utiliza el servicio TLS.
configuracionesGlobales__correo__smtp__nombreDeUsuario	Ingrese el nombre de usuario SMTP.
globalSettings__mail__smtp__contraseña	Ingrese la contraseña SMTP.

API de Yubico (YubiKey)

Variable	Descripción
configuraciónGlobal__yubico__idCliente	Reemplace el valor con la ID recibida de su Yubico Key. Regístrate para la llave Yubico aquí.
configuraciónGlobal__yubico__clave	Ingrese el valor de la clave recibida de Yubico.

Configuraciones de base de datos

La utilización de la variedad de opciones de base de datos que son compatibles con la implementación unificada requerirá configuraciones adicionales de `.env`.

⇒MySQL/MariaDB

En `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

En `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

En `settings.env`:

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

Asignar el valor de `sqlite` creará automáticamente un archivo `vault.db` en el volumen `/etc/bitwarden`. `BW_DB_FILE` solo es necesario si desea especificar la ruta a un archivo de base de datos diferente.

⇒PostgreSQL

En `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
```

Otro

Variable	Descripción
configuraciónGlobal__deshabilitarRegistroDeUsuario	Habilitar o deshabilitar las capacidades de registro de cuenta de usuario.
globalSettings__hibpApiKey	Ingrese la clave API proporcionada por Have I Been Pwnd. Regístrate para recibir la clave de la API aquí.
configuraciónDeAdministrador__administradores	Ingrese las direcciones de correo electrónico del administrador.
BW_REAL_IPS	Defina IPs reales en `nginx.conf` en una lista separada por comas. Útil para definir servidores proxy que reenvían la dirección IP del cliente. Más información.
BW_CSP	Parámetro de Política de Seguridad de Contenido. Reconfigurar este parámetro puede romper funcionalidades. Al cambiar este parámetro, te conviertes en responsable de mantener este valor.
BW_DB_PORT	Especifique un puerto personalizado para el tráfico de la base de datos. Si no se especifica, el valor predeterminado dependerá de su proveedor de base de datos elegido.

Reinicia el contenedor

Para reiniciar su contenedor Docker después de cambiar las variables de entorno, ejecute los siguientes comandos desde el directorio de despliegue unificado de Bitwarden:

⇒Docker ejecutar

1. Detén el contenedor Docker en ejecución:

```Bash
docker stop bitwarden
```

2. Elimina el contenedor Docker:

```Bash
docker rm bitwarden
```

3. Ejecuta el contenedor Docker de nuevo:

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

⇒Docker Compose

1. Detén el contenedor Docker en ejecución:

```Bash
docker compose down
```

2. Recrear los contenedores:

```Bash
docker compose up -d
```

3. Asegúrate de que los contenedores estén funcionando correctamente con:

```Bash
docker compose ps
```

Uso de memoria

Por defecto, el contenedor de Bitwarden consumirá la memoria que esté disponible para él, a menudo siendo más de la mínima necesaria para funcionar. Para entornos conscientes de la memoria, puedes usar docker `-m` o `--memory=` para limitar el uso de memoria del contenedor Bitwarden.

Nombre, abreviatura	Descripción
--memoria=, -m	La cantidad máxima de memoria que puede usar el contenedor. Bitwarden requiere al menos 200m. Consulta la documentación de Docker para aprender más.

Para controlar el uso de memoria con Docker Compose, usa la clave `mem_limit`:

```
Bash

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

Informando problemas

Mientras que el despliegue unificado de Bitwarden permanece en versión beta, te animamos a informar sobre problemas y dar retroalimentación a través de GitHub. Por favor, use esta plantilla de problema para informar cualquier cosa relacionada con su despliegue unificado de Bitwarden y consulte esta página para seguir los problemas conocidos o unirse a la discusión.

Recursos adicionales

- Si planeas autoalojar una organización Bitwarden, consulta autoalojar una organización para comenzar.

Para obtener más información sobre la implementación estándar autoalojada de Bitwarden, consulte:

- Instalar y Desplegar - Linux
- Instalar y Desplegar - Windows
- Instalar y Desplegar - Manual