Vaultwarden : héberger son gestionnaire de mots de passe avec Docker et Traefik

Vaultwarden est un fork Rust non-officiel de Bitwarden, 100 % compatible avec les applications officielles Bitwarden, consommant 10x moins de ressources. Déployé en Docker avec Nginx et Let's Encrypt, il offre un gestionnaire de mots de passe auto-hébergé, sécurisé et gratuit pour équipes et particuliers.

Confier ses mots de passe à un service cloud tiers, c'est un acte de foi. Pas toujours justifié. **Vaultwarden** résout ce problème en vous permettant d'héberger votre propre gestionnaire de mots de passe, **compatible avec l'ensemble des applications officielles Bitwarden** (iOS, Android, extensions Chrome/Firefox, desktop Windows/macOS/Linux), sur votre propre serveur, sous votre contrôle total. Contrairement au serveur Bitwarden officiel qui nécessite 2–4 Go de RAM et un environnement .NET, *Vaultwarden* (anciennement bitwarden_rs, disponible sur GitHub sous licence AGPL-3.0) est écrit en Rust et consomme environ 50 Mo de RAM — parfait pour un VPS entrée de gamme ou un Raspberry Pi 4. Pour les équipes de sécurité, les développeurs soucieux de leur souveraineté numérique et les organisations soumises à des exigences de résidence des données, **vaultwarden docker self-hosted** est le meilleur rapport fonctionnalités/coût/contrôle disponible en 2026. Ce guide couvre le déploiement complet avec docker-compose, la configuration Nginx en reverse proxy avec certificat Let's Encrypt automatique, la sécurisation de l'admin panel, la configuration du 2FA (TOTP, email OTP), le système de backup automatique et le hardening pour un usage en production.

À RETENIR

À retenir

Compatible Bitwarden 100 % : toutes les applications officielles Bitwarden fonctionnent avec Vaultwarden — iOS, Android, extensions navigateur, desktop.

Ultra léger : 50 Mo de RAM en fonctionnement normal, vs 2–4 Go pour le serveur Bitwarden officiel.

Docker obligatoire : Vaultwarden est distribué exclusivement via Docker Hub (vaultwarden/server) ; pas de paquet natif officiel.

Admin panel à protéger : le panel admin par défaut est accessible sur `/admin` sans authentification — à sécuriser immédiatement.

Backup critique : le répertoire `/data/` contient toute la base SQLite et les clés de chiffrement — un backup quotidien chiffré est non-négociable.

Vaultwarden vs Bitwarden officiel : quelles différences concrètes ?

La question revient à chaque fois : pourquoi utiliser Vaultwarden plutôt que d'héberger le serveur Bitwarden officiel ? La réponse tient en trois critères : les ressources système, la complexité de déploiement et les fonctionnalités.

Le serveur Bitwarden officiel est une application .NET complexe avec une dizaine de microservices (API, Identity, Admin, Icons, Notifications, SSO...), un serveur MSSQL ou une base de données dédiée, et des exigences matérielles de 2–4 Go de RAM minimum. Il est conçu pour le déploiement enterprise en cluster. Vaultwarden est un binaire Rust unique avec une base

SQLite embarquée (PostgreSQL et MySQL aussi supportés), qui expose la même API REST que le serveur officiel.

Critère	Vaultwarden	Bitwarden officiel
RAM requise	50–150 Mo	2–4 Go minimum
Base de données	SQLite (défaut), PostgreSQL, MySQL	MSSQL uniquement (officiel)
Support officiel	Non (projet communautaire)	Oui (Bitwarden Inc.)
Compatibilité apps	Complète (API identique)	Complète (référence)
2FA	TOTP, Email, Duo, WebAuthn	TOTP, Email, Duo, WebAuthn, YubiKey
SMTP requis	Fortement recommandé	Oui (obligatoire)
Coût licence	Gratuit (AGPL-3.0)	Gratuit (AGPL pour le code serveur)

Un point important à comprendre : Vaultwarden est un **projet communautaire non affilié à Bitwarden Inc.** Il ne reçoit pas de mises à jour de sécurité de Bitwarden Inc. et peut accuser un retard lors de changements d'API. En pratique, la compatibilité est maintenue rapidement par la communauté (retard habituel de 24–72h lors des mises à jour Bitwarden), mais pour des organisations avec des exigences de support formel, le serveur officiel est plus approprié.

Prérequis : VPS, domaine et Docker

Pour un déploiement en production, vous avez besoin d'un **VPS Linux** avec au moins 512 Mo de RAM (recommandé : 1 Go), d'un **domaine ou sous-domaine** pointant vers l'IP du serveur, et de **Docker + Docker Compose** installés. Un VPS entrée de gamme chez Hetzner (CAX11 à 3,29 €/mois) ou OVH (Starter à 3,59 €/mois) est largement suffisant.

```
# Installer Docker et Docker Compose sur Debian 12 / Ubuntu 22.04
curl -fsSL https://get.docker.com | sh
sudo usermod -aG docker $USER

# Vérifier
docker --version && docker compose version
# Docker version 27.x.x
# Docker Compose version v2.x.x
```

Créez un répertoire dédié et les sous-répertoires nécessaires :

```
sudo mkdir -p /opt/vaultwarden/{data,nginx/conf,ssl}
sudo chown -R $USER:$USER /opt/vaultwarden
cd /opt/vaultwarden
```

Déploiement avec docker-compose : configuration complète

Voici le fichier docker-compose.yml de production, avec Vaultwarden, Nginx comme reverse proxy et Certbot pour Let's Encrypt :

```
 # /opt/vaultwarden/docker-compose.yml
version: '3.8'

services:
  vaultwarden:
    image: vaultwarden/server:latest
    container_name: vaultwarden
    restart: always
    volumes:
      - ./data:/data
    environment:
      # URL publique OBLIGATOIRE — doit correspondre exactement à votre domaine
      DOMAIN: "https://vault.exemple.fr"

      # Désactiver les inscriptions publiques (recommandé en production)
      SIGNUPS_ALLOWED: "false"

      # Activer les invitations par email (admin peut inviter des utilisateurs)
      INVITATIONS_ALLOWED: "true"

      # Token admin — générer avec : openssl rand -base64 48
      ADMIN_TOKEN: "VOTRE_TOKEN_FORT_48_CHARS_MINIMUM"

      # SMTP pour les emails (vérification, 2FA, invitations)
      SMTP_HOST: "smtp.exemple.fr"
      SMTP_FROM: "vault@exemple.fr"
      SMTP_PORT: "587"
      SMTP_SECURITY: "starttls"
      SMTP_USERNAME: "vault@exemple.fr"
      SMTP_PASSWORD: "MOT_DE_PASSE_SMTP"

      # Activer les logs détaillés
      LOG_LEVEL: "warn"
      EXTENDED_LOGGING: "true"

    # Ne pas exposer les ports directement — Nginx est le seul point d'entrée
    networks:
      - vaultwarden_net

  nginx:
    image: nginx:alpine
    container_name: vaultwarden_nginx
    restart: always
```

```
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./nginx/conf:/etc/nginx/conf.d:ro
      - ./ssl:/etc/letsencrypt:ro
      - ./nginx/certbot-webroot:/var/www/certbot:ro
    depends_on:
      - vaultwarden
    networks:
      - vaultwarden_net

  certbot:
    image: certbot/certbot
    container_name: vaultwarden_certbot
    volumes:
      - ./ssl:/etc/letsencrypt
      - ./nginx/certbot-webroot:/var/www/certbot
    entrypoint: "/bin/sh -c 'trap exit TERM; while :; do certbot renew --webroot -w /var/www/cert

networks:
  vaultwarden_net:
    driver: bridge
```

Configuration Nginx : reverse proxy avec TLS Let's Encrypt

Créez la configuration Nginx dans `/opt/vaultwarden/nginx/conf/vaultwarden.conf` :

```
 # /opt/vaultwarden/nginx/conf/vaultwarden.conf

# Redirection HTTP → HTTPS
server {
    listen 80;
    server_name vault.exemple.fr;

    # Pour le renouvellement Let's Encrypt
    location /.well-known/acme-challenge/ {
        root /var/www/certbot;
    }

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl http2;
    server_name vault.exemple.fr;

    ssl_certificate     /etc/letsencrypt/live/vault.exemple.fr/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/vault.exemple.fr/privkey.pem;
    ssl_protocols       TLSv1.2 TLSv1.3;
    ssl_ciphers         HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;

    # En-têtes de sécurité
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
    add_header X-Content-Type-Options nosniff;
    add_header X-Frame-Options DENY;
    add_header Referrer-Policy no-referrer;

    # Taille maximale du corps (pour les pièces jointes Bitwarden)
    client_max_body_size 128M;

    # Proxy vers Vaultwarden
    location / {
        proxy_pass http://vaultwarden:80;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
```

```
    }

    # WebSocket pour les notifications en temps réel
    location /notifications/hub {
        proxy_pass http://vaultwarden:3012;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}
```

Obtenir le certificat Let's Encrypt initial (avant de lancer docker compose) :

```
 # Obtenir le certificat initial avec certbot en standalone
sudo certbot certonly --standalone -d vault.exemple.fr   --email admin@exemple.fr --agree-tos --n

# Ensuite démarrer les containers
cd /opt/vaultwarden && docker compose up -d

# Vérifier que tout fonctionne
docker compose ps
docker compose logs vaultwarden --tail=20
```

Sécuriser l'admin panel Vaultwarden

Par défaut, si vous définissez `ADMIN_TOKEN` dans la configuration, le panel admin est accessible sur `https://vault.exemple.fr/admin` avec ce token. C'est la première chose à sécuriser après le déploiement.

Le token admin doit être généré de façon cryptographiquement sûre :

```
 # Générer un token fort (48+ bytes en base64)
openssl rand -base64 48
# Résultat exemple : mK3oNR7pX2jL8qF1uZ5yAcWvHbEd4sTm6iYnPwRgJUkQx9hCVlBaOfIG/0+DZXe=

# Ou utiliser argon2 pour un hachage encore plus sécurisé (recommandé en prod)
# Vaultwarden accepte les hashs argon2 pour ADMIN_TOKEN depuis la version 1.29
echo -n "VOTRE_TOKEN_CLAIR" | argon2 "$(openssl rand -hex 16)" -id -t 3 -m 16 -p 4 -l 32 -e
```

En complément, limitez l'accès à `/admin` au niveau Nginx par IP autorisées :

```
 # Ajouter dans le bloc server 443 de vaultwarden.conf
location /admin {
    # Restreindre aux IPs de l'équipe d'administration
    allow 203.0.113.0/24;   # IP bureau
    allow 198.51.100.5;     # IP admin VPN
    deny all;               # Bloquer tout le reste

    proxy_pass http://vaultwarden:80;
    proxy_set_header Host $host;
}
```

Après configuration initiale (création des comptes, paramétrage SMTP), il est possible de désactiver complètement le panel admin en supprimant la variable `ADMIN_TOKEN` de l'environnement Docker. Pour en savoir plus sur la sécurisation des gestionnaires de mots de passe face aux attaques, notre article sur le hacking des password managers de navigateurs détaille les vecteurs d'attaque et les contre-mesures. À noter qu'en 2026, la supply chain npm a été ciblée : notre article sur l'incident Bitwarden CLI et Shai-Hulud rappelle l'importance de vérifier les sources.

Comment configurer le 2FA sur Vaultwarden ?

Vaultwarden supporte plusieurs méthodes de second facteur. La configuration se fait côté utilisateur dans l'application Bitwarden (pas dans l'admin panel). Pour les administrateurs, forcer le 2FA pour tous les utilisateurs se fait via la variable d'environnement `REQUIRE_DEVICE_EMAIL: "true"`.

TOTP (Google Authenticator, Authy, Bitwarden Authenticator) : méthode recommandée. Dans l'app Bitwarden → Paramètres → Sécurité → Authentification à deux facteurs → Authentificateur. Scannez le QR code avec votre app TOTP.

Email OTP : Vaultwarden envoie un code à 6 chiffres par email. Nécessite une configuration SMTP fonctionnelle. Méthode de secours pratique, moins sécurisée que TOTP (dépend de la sécurité de votre email).

WebAuthn / FIDO2 : clés matérielles (YubiKey, Google Titan, Nitrokey). Niveau de sécurité le plus élevé. Configuration via l'interface web Bitwarden → Paramètres → 2FA → Clé de sécurité FIDO2. Pour aller plus loin sur les mécanismes d'authentification forte, notre article sur le bypass FIDO2 et les attaques sur l'authentification forte est utile pour comprendre les limites de chaque méthode.

```
 # Variables d'environnement pour forcer le 2FA
# Dans docker-compose.yml → environment :
REQUIRE_DEVICE_EMAIL: "true"  # Force vérification email sur nouveaux appareils
# Note : ne force pas le TOTP, mais ajoute une couche de vérification
```

Backup automatique : protection contre la perte de données

Le répertoire `/data/` de Vaultwarden contient absolument tout : la base SQLite (`db.sqlite3`), les pièces jointes, les configurations et les clés de chiffrement RSA. **Perdre ce répertoire = perdre tous les mots de passe.**

```
#!/bin/bash
# /opt/vaultwarden/scripts/backup.sh

BACKUP_DIR="/opt/backups/vaultwarden"
DATE=$(date +%Y%m%d_%H%M%S)
VAULTWARDEN_DATA="/opt/vaultwarden/data"
RETENTION_DAYS=30

# Créer le répertoire de backup
mkdir -p "$BACKUP_DIR"

# Stopper temporairement Vaultwarden pour un backup SQLite cohérent
docker compose -f /opt/vaultwarden/docker-compose.yml stop vaultwarden

# Copie du répertoire data
tar -czf "$BACKUP_DIR/vaultwarden_${DATE}.tar.gz" -C "$VAULTWARDEN_DATA" .

# Chiffrer le backup avec OpenSSL (AES-256)
openssl enc -aes-256-cbc -pbkdf2 -iter 100000   -in "$BACKUP_DIR/vaultwarden_${DATE}.tar.gz"   -o

# Supprimer le backup non chiffré
rm "$BACKUP_DIR/vaultwarden_${DATE}.tar.gz"

# Redémarrer Vaultwarden
docker compose -f /opt/vaultwarden/docker-compose.yml start vaultwarden

# Supprimer les backups de plus de 30 jours
find "$BACKUP_DIR" -name "*.enc" -mtime +$RETENTION_DAYS -delete

echo "Backup terminé : vaultwarden_${DATE}.tar.gz.enc"
```

```
# Planifier le backup quotidien à 3h du matin
echo "0 3 * * * root /opt/vaultwarden/scripts/backup.sh >> /var/log/vaultwarden-backup.log 2>&1"

# Rendre le script exécutable
chmod +x /opt/vaultwarden/scripts/backup.sh
```

Synchronisez les backups chiffrés vers un stockage externe : Backblaze B2 (très économique, 0,006 $/Go/mois), AWS S3, ou un second serveur via rsync. Un backup qui reste sur le même serveur que les données ne protège pas contre la perte du serveur lui-même.

Vaultwarden en production : hardening et monitoring

Quelques mesures supplémentaires pour durcir votre instance Vaultwarden en production.

Désactiver les inscriptions publiques (`SIGNUPS_ALLOWED: "false"`) est la première mesure après le déploiement. Sans ça, n'importe qui connaissant l'URL peut créer un compte. Utilisez les invitations admin pour ajouter des utilisateurs manuellement.

Configurer Fail2ban sur les logs Nginx pour bloquer les IPs qui tentent des attaques bruteforce sur la page de connexion :

```
 # /etc/fail2ban/filter.d/vaultwarden.conf
[Definition]
failregex = ^.*"POST /api/accounts/login.*" (401|429)
ignoreregex =

# /etc/fail2ban/jail.d/vaultwarden.conf
[vaultwarden]
enabled  = true
port     = http,https
filter   = vaultwarden
logpath  = /var/log/nginx/access.log
maxretry = 5
bantime  = 14400   # 4 heures
findtime = 600     # Dans les 10 dernières minutes
```

Monitoring de disponibilité : configurez un outil de monitoring externe (UptimeRobot gratuit, Freshping) pour recevoir une alerte si Vaultwarden devient inaccessible. La perte d'accès au gestionnaire de mots de passe en urgence est une situation critique.

Mises à jour : Vaultwarden suit le cycle de releases Bitwarden. Pour mettre à jour :

```
 cd /opt/vaultwarden
docker compose pull vaultwarden
docker compose up -d vaultwarden
# Tester l'accès après mise à jour
curl -sI https://vault.exemple.fr/ | head -3
```

Pour les organisations qui gèrent des infrastructures Docker en sécurité et souhaitent comprendre les risques d'évasion de containers, notre article sur les techniques d'évasion Docker et containerd est un complément utile à cette configuration.

Limites et cas où Vaultwarden n'est pas le bon choix

Soyons honnêtes : Vaultwarden n'est pas la bonne réponse dans tous les cas.

Support formel requis : si votre organisation a besoin d'un SLA, d'un support enterprise et de garanties de sécurité auditées, le cloud Bitwarden Teams/Enterprise ou le serveur Bitwarden officiel auto-hébergé sont plus appropriés. Vaultwarden est un projet communautaire sans engagement de maintenance. À titre de comparaison, Bitwarden Inc. publie annuellement un livre blanc de sécurité audité par un tiers indépendant — une garantie que Vaultwarden ne peut pas offrir.

SSO SAML/OIDC : l'intégration SSO (Entra ID, Okta, Google Workspace) nécessite la version Enterprise de Bitwarden Teams. Vaultwarden supporte partiellement OIDC via le module SSO communautaire, mais la compatibilité n'est pas garantie.

Haute disponibilité : Vaultwarden avec SQLite n'est pas adapté à un clustering actif-actif. Pour une disponibilité >= 99,9 %, utilisez PostgreSQL en backend et planifiez une architecture active-passive.

Organisations et collections : partager des mots de passe en équipe

Vaultwarden supporte les **Organisations** Bitwarden — la fonctionnalité qui permet à plusieurs utilisateurs de partager des mots de passe dans des collections sécurisées. C'est la fonctionnalité clé pour les équipes et les familles.

Pour créer une Organisation depuis l'interface web Vaultwarden :

Connectez-vous à votre instance Vaultwarden (`https://vault.exemple.fr`)

Cliquez sur votre profil → **Nouvelle Organisation**

Choisissez un nom et un plan (Free pour les organisations personnelles)

L'organisation est créée — créez des Collections (équivalent de dossiers partagés)

Invitez des membres par email (nécessite SMTP configuré)

Structure recommandée pour une PME de 10 personnes :

Collection "Infra" : credentials serveurs, switch, routeur, stockage (accès équipe IT)

Collection "Services SaaS" : comptes AWS, GitHub, Stripe, Mailchimp (accès élargi)

Collection "Clients" : accès portails clients, VPN partenaires (accès restreint par projet)

Collection "RH / Confidentiel" : accès SIRH, banque, compta (accès direction uniquement)

Les membres d'une Organisation peuvent avoir différents rôles : **Utilisateur** (lecture seule ou lecture/écriture selon la collection), **Gestionnaire** (peut gérer les collections), **Admin** (accès total à l'organisation), **Propriétaire**. La granularité par collection permet un contrôle d'accès précis sans nécessiter plusieurs instances Vaultwarden.

Accès d'urgence : récupérer l'accès en cas de perte du Master Password

La fonctionnalité d'**Accès d'urgence** (Emergency Access) de Bitwarden est supportée par Vaultwarden. Elle permet à un utilisateur de confiance de demander l'accès à votre coffre si vous êtes incapable de le déverrouiller (accident, décès, Master Password oublié).

```
 # Variables d'environnement Vaultwarden pour activer Emergency Access
EMERGENCY_ACCESS_ALLOWED: "true"  # Activé par défaut dans les versions récentes
```

Configuration côté utilisateur dans l'app Bitwarden :

Paramètres → Sécurité → Accès d'urgence

Ajouter un contact de confiance (collègue, conjoint) par email

Définir le délai d'attente avant accès automatique (1 à 90 jours)

Le contact reçoit une invitation email — doit accepter

Si vous perdez accès à votre compte, le contact de confiance peut demander l'accès d'urgence. Vous recevez une notification et pouvez refuser. Après le délai défini (pendant lequel vous pouvez bloquer la demande), le contact obtient automatiquement l'accès. C'est une protection essentielle pour les usages professionnels critiques — personne ne devrait être le single point of failure d'un coffre qui stocke les accès à l'infrastructure de production.

Pour la gestion des identités et des accès dans un contexte plus large, notre article sur le Password Filter DLL et la défense des mots de passe explore les vecteurs d'attaque spécifiques à Active Directory, complémentaire à la sécurisation de votre gestionnaire de mots de passe.

Questions fréquentes

Vaultwarden est-il sécurisé pour un usage en entreprise ?

Pour les PME et les équipes techniques averties, oui — sous conditions. La sécurité de Vaultwarden repose sur votre infrastructure : TLS à jour, admin panel restreint, inscriptions désactivées, backups chiffrés quotidiens, mises à jour régulières. Le code Vaultwarden a fait l'objet d'audits informels par la communauté open-source, mais pas d'audit formel certifié comme celui que Bitwarden Inc. publie annuellement. Pour des données critiques en contexte réglementé (santé, finances), évaluez d'abord si l'absence d'audit formel est acceptable pour votre politique de sécurité.

Puis-je migrer depuis Bitwarden cloud vers Vaultwarden ?

Oui, la migration est straightforward. Dans votre compte Bitwarden cloud → Outils → Exporter le coffre → choisissez JSON chiffré. Sur votre instance Vaultwarden, connectez-vous → Outils → Importer des données → sélectionnez le format Bitwarden JSON. Tous vos mots de passe, notes sécurisées et pièces jointes sont importés. Les organisations Bitwarden nécessitent un

import séparé par organisation. Après import, désactivez votre compte Bitwarden cloud ou supprimez le coffre pour éviter la duplication.

Que se passe-t-il si mon serveur Vaultwarden est compromis ?

Les données dans Vaultwarden sont chiffrées côté client avec votre Master Password avant d'être envoyées au serveur — c'est la même architecture zero-knowledge que Bitwarden cloud. Un attaquant accédant à la base SQLite ne voit que des données chiffrées. La clé de déchiffrement n'est jamais stockée sur le serveur. La vraie menace dans un scénario de compromission serveur est l'injection d'un faux client JavaScript qui vole le Master Password avant chiffrement — d'où l'importance de vérifier les checksums des images Docker et de maintenir l'intégrité du serveur Nginx.

Comment configurer Vaultwarden pour plusieurs utilisateurs ou familles ?

Vaultwarden supporte les **Organisations** (collections partagées) et les **familles Bitwarden** de façon native. Une fois les comptes créés (par invitation admin), les utilisateurs peuvent partager des entrées dans des collections organisationnelles. Pour un usage familial (4–6 personnes), la configuration recommandée est : 1 Organisation "Famille", collections par usage (Banque, Maison, Streaming), partage sélectif par membre. La limite pratique est la mémoire du serveur : chaque utilisateur actif représente quelques Mo de RAM supplémentaire — négligeable pour des usages non enterprise.

Vaultwarden fonctionne-t-il offline / sans internet ?

L'application Bitwarden sur mobile et desktop dispose d'un cache local chiffré qui fonctionne offline pour lire les mots de passe existants. En revanche, les mises à jour (nouveaux mots de passe, modifications) nécessitent une connexion à votre instance Vaultwarden. Si votre serveur est inaccessible (panne réseau, maintenance), vous pouvez toujours accéder aux dernières données synchronisées depuis l'app. C'est une raison supplémentaire de maintenir Vaultwarden en haute disponibilité pour un usage critique.

Votre équipe a besoin d'un gestionnaire de mots de passe sécurisé et souverain ? Notre équipe peut déployer et durcir Vaultwarden sur votre infrastructure ou vous accompagner dans le choix de la solution adaptée à vos contraintes. Contactez notre équipe RSSI externalisé dès aujourd'hui.