

UE SEC-105

Travaux pratiques

Mise en œuvre d'un outil ou  
d'un concept de sécurité

## Installation et utilisation de Vaultwarden



## Suivi du document

Date	Description	Auteur
28/10/2024	Création de l'exemple	Julien Le Coz

## Autorisation de partage

☒ J'autorise le partage du présent document aux autres élèves « Sec-105 » (la présente page sera supprimée pour garantir l'anonymisation du document)

## Composition de :

- premier auditeur CNAM : Julien Le Coz

## Table des matières

Description de l'objectif de sécurité visé.....	5
Schéma de principe.....	6
Implémentation.....	7
Conclusion.....	25
Sources.....	27
Sources WEB.....	27
Sources humaines.....	27
Annexes (facultatives).....	28

## Description de l'objectif de sécurité visé

A l'heure actuelle ma façon de gérer mes mots de passe est très problématique : je me contente de les stocker sur mon disque dur dans un fichier chiffré grâce à l'éditeur vim (algorithme Blowfish2)

Bien que cet algorithme soit à priori actuellement sûr, consulter ce fichier n'est pas très pratique et le copié-collé n'est pas possible. Cela se traduit par l'utilisation de mots de passe qui ne sont que moyennement complexes, et la réutilisation de certains de ces mots de passe. Bien sûr cela ne m'incite pas non plus à leur modification régulière. Qui plus est, mes mots de passe sont disponibles uniquement sur mon PC.

Je suis marié et j'ai 4 enfants. La gestion que fait ma femme de ses mots de passe est tout simplement catastrophique et les enfants perdent régulièrement leur accès à certains services où notent des mots de passe dans des carnets. Evidemment les mots de passe Netflix et autres nous sont régulièrement demandés vu que l'information ne leur est pas facilement accessible.

Je possède un PC sous Linux et ma femme un PC sous Windows. Nous avons tous un téléphone mobile, dont 5 appareils Android et un appareil Apple. Je recherche donc une solution qui soit utilisable quelque soit la plateforme pour stocker et diffuser ces informations de façon sécurisée et facile d'utilisation pour inciter les moins sensibles à ces enjeux à l'utiliser.

Dans mon entreprise nous utilisons Keypass pour stocker tous les mots de passe commun. La gestion des mots de passe personnels est à la charge de chacun, ce qui sous-entend l'utilisation d'au moins deux solutions différentes. Une solution permettant de gérer plusieurs comptes permettrait de rendre cette gestion plus claire et inciter tout le monde à utiliser systématiquement un gestionnaire de mot de passe.

Je vais donc chercher à configurer et déployer une instance de Vaultwarden sur un serveur virtuel d'Ionos de façon à ce que chaque membre de mon foyer puisse l'utiliser et tester ainsi l'intérêt et l'utilisation d'un gestionnaire de mot de passe de la façon la plus concrète qui soit. Il serait bon de sauvegarder toutes ces données, comme les options de sauvegarde automatique sont payantes, je vais plutôt envisager de la gérer moi-même en m'aidant d'un serveur loué chez OVH.

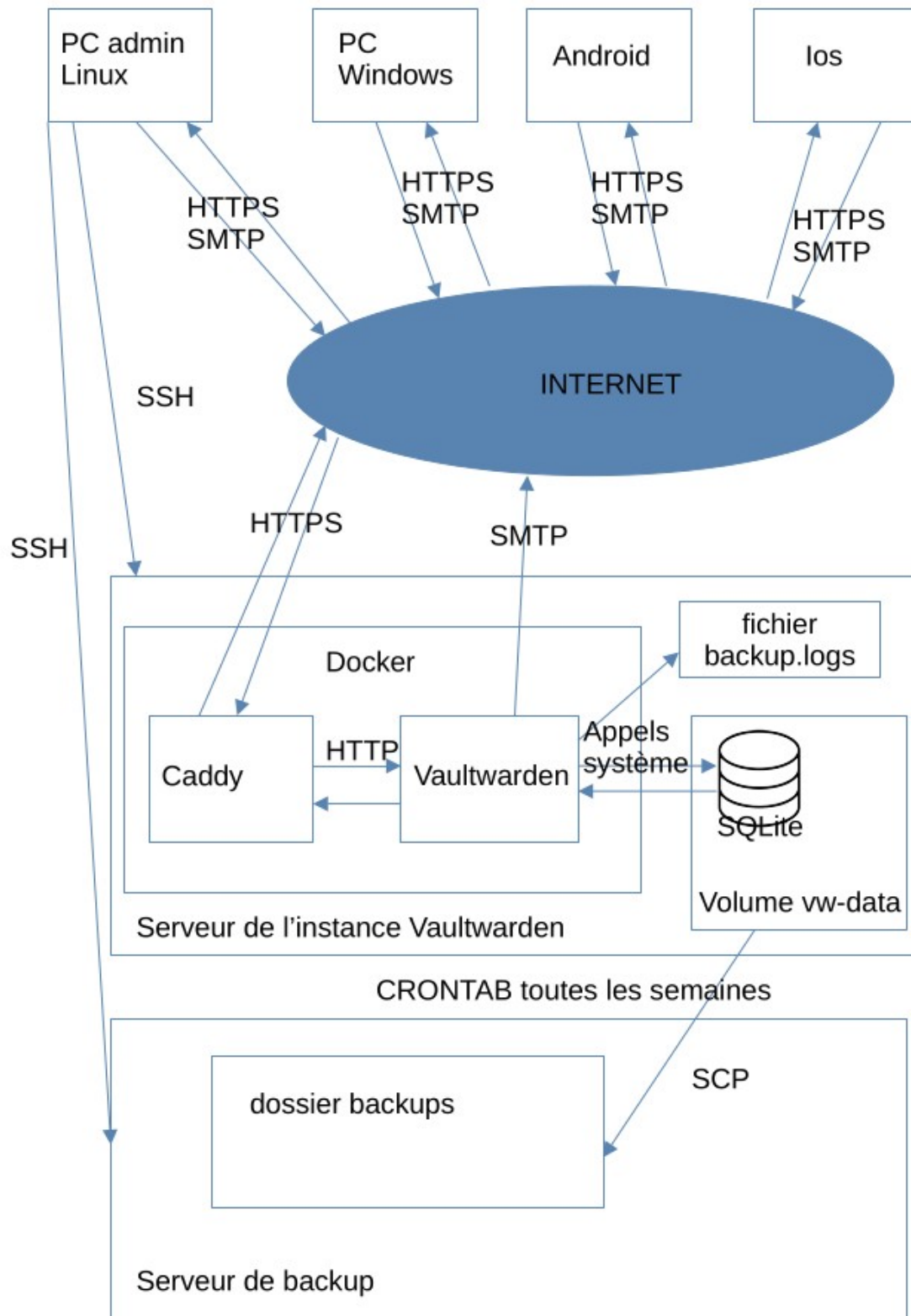
Au final je dois donc :

- Créer un serveur Vaultwarden
- Tester le fonctionnement de l'extension navigateur
- Gérer les envois de mails
- Tester la création d'organisations et le partage de mots de passe
- Assurer la sauvegarde et tester la restauration
- Tester le fonctionnement de l'application mobile

Dans un premier temps, l'objectif sera atteint lorsque toute ma famille délaissera ses notes sur le téléphone ou son carnet de mots de passe pour l'utilisation généralisée de Vaultwarden et quand plus personne n'aura besoin de nous demander les identifiants de tel ou tel compte.

En ce qui concerne mon entreprise, on verra ça dans un second temps, après le test maison grandeur nature !

## Schéma de principe



# Implémentation

Après avoir créé un compte utilisateur avec des droits sudo sur mon serveur virtuel Debian, je me suis assuré de pouvoir m'y connecter via SSH afin de pouvoir confortablement faire toutes les manipulations nécessaires à l'installation de Vaultwarden. Je me suis aussi créé un sous-domaine et une adresse mail dédiée.

## Installation :

Petite mise à jour du système et on commence par installer les paquets requis pour l'installation de Docker. (nécessaire pour l'installation de Vaultwarden)

```
julien@ionos-server:~$ sudo apt-get install docker-compose apt-transport-https ca-certificates gnupg2 software-properties-common
```

On télécharge ensuite la clef GPG depuis le référentiel Docker. Cette clef permet de s'assurer que les paquets proviennent d'une source fiable

```
julien@ionos-server:~$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

On ajoute le référentiel Docker aux sources du gestionnaire de paquets...

```
julien@ionos-server:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"
```

... et on fait une mise à jour pour s'assurer d'avoir les dernières informations concernant les paquets en question.

```
julien@ionos-server:~$ sudo apt-get update
```

On accède à la stratégie de cache pour vérifier que les paquets d'installation proviennent bien de Docker.

```
julien@ionos-server:~$ sudo apt-cache policy docker-ce
```

Et on peut enfin finir par installer Docker en lui-même !

```
julien@ionos-server:~$ sudo apt-get install docker-ce
```

Docker est installé, mais une petite vérification à posteriori n'est jamais inutile

```
julien@ionos-server:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-11-17 19:58:37 UTC; 20s ago
 TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 12306 (dockerd)
     Tasks: 7
    Memory: 29.6M
       CPU: 286ms
    CGroup: /system.slice/docker.service
            └─12306 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 17 19:58:37 ionos-server systemd[1]: Starting docker.service - Docker Application Container Engine.
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.261687165Z" level=info msg="Starting daemon"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.340264988Z" level=info msg="Graph driver"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.340938611Z" level=info msg="Loading"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.617779211Z" level=info msg="Default"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.674300445Z" level=info msg="Loading"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.688068183Z" level=info msg="Documentation"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.688298688Z" level=info msg="Daemon"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.719756640Z" level=info msg="API"
Nov 17 19:58:37 ionos-server systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-22/22 (END)
```

Nous avons donc bien une instance fonctionnelle de docker qui tourne sur le serveur.

Docker installé, nous avons déjà fait le plus gros du boulot : il ne nous reste plus qu'à lancer deux conteneurs.

- Le conteneur dans lequel tournera notre instance de Vaultwarden
- Le conteneur dans lequel nous allons faire tourner un reverse-proxy afin d'établir une

connexion via https pour protéger nos interactions avec l'instance de Vaultwarden des attaques type « man in the middle »

On commence par créer un dossier qui contiendra notre configuration puis à l'intérieur de ce dossier un fichier de configuration pour nos deux conteneurs (détails du docker-compose.yml en annexe). Nous allons utiliser des volumes afin de conserver hors du conteneur certaines des données consommées par notre instance de Vaultwarden. D'une part on s'assure ainsi de ne pas supprimer toutes ces données à chaque redémarrage du conteneur, d'autre part, comme nous le verrons plus tard cela sera pratique pour gérer tout ce qui a trait à la sauvegarde des données en question.

```
julien@ionos-server:~$ mkdir vaultwarden && cd vaultwarden && touch docker-compose.yml
```

Pour le reverse-proxy, nous allons utiliser Caddy qui a le gros avantage d'être facile à déployer et de gérer automatiquement le protocole HTTPS. Il nous faut un fichier de configuration, fichier qui sera utilisé par l'instance de Caddy (détails de Caddyfile en annexe)

```
julien@ionos-server:~/vaultwarden$ touch Caddyfile
```



La configuration des conteneurs est terminée. C'est maintenant l'heure de vérité. On monte les conteneurs...

```
julien@ionos-server:~/vaultwarden$ sudo docker-compose up -d
```

Et tout semble bien se passer. Ici aussi une petite vérification ne serait pas de trop...

```
julien@ionos-server:~/vaultwarden$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
NAMES
eef7e65bc713   vaultwarden/server:latest           "/start.sh"             15 seconds ago Up 14 seconds (hea
lth: starting) 80/tcp
vaultwarden
897a4f90e924   caddy:2                             "caddy run --config ..." 15 seconds ago Up 14 seconds
0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp, 443/udp, 201
9/tcp caddy
julien@ionos-server:~/vaultwarden$
```

Tout semble correct ! Rendons nous sur l'URL dédiée à notre VPS pour voir ce que ça raconte :



Et cette installation est un succès tout ce qu'il y a de plus manifeste !

## Utilisation :

Comme la page web nous invite à créer un compte, nous allons donc nous y coller.

### Créez un compte

**Adresse électronique** (requis)

Vous utiliserez votre adresse électronique pour vous connecter.

**Nom**

Comment doit-on vous appeler ?

**Mot de passe principal** (requis)

Important : Votre mot de passe principal ne peut pas être récupéré si vous l'oubliez ! 12 caractères minimum

**Ressaisir le mot de passe principal** (requis)

**Indice du mot de passe principal**

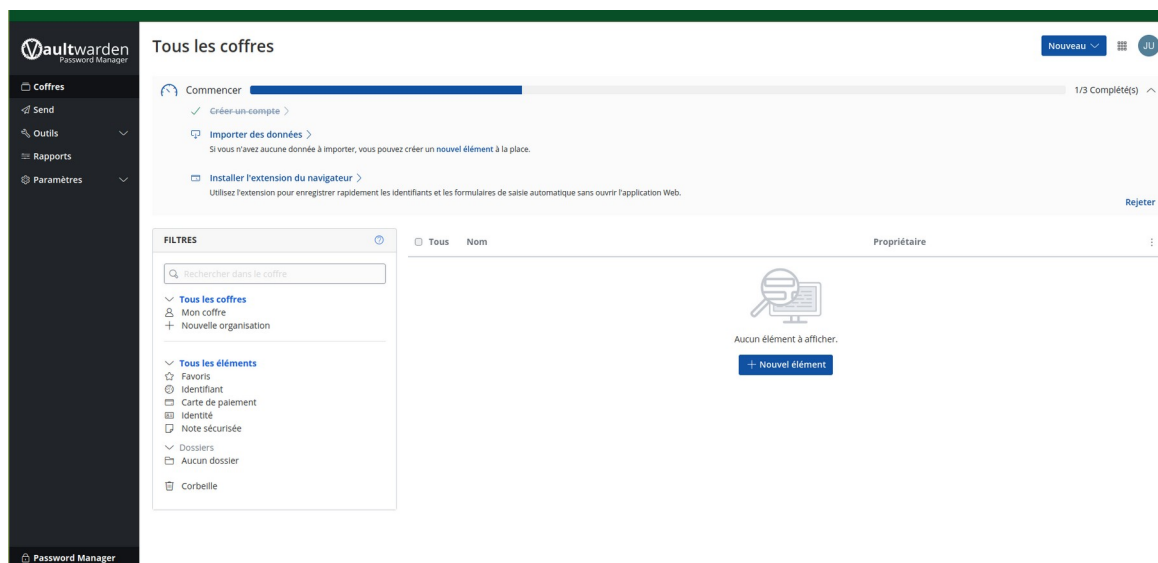
Un indice de mot de passe principal peut vous aider à vous souvenir de votre mot de passe si vous l'oubliez.

☒ Vérifier les brèches de données connues pour ce mot de passe

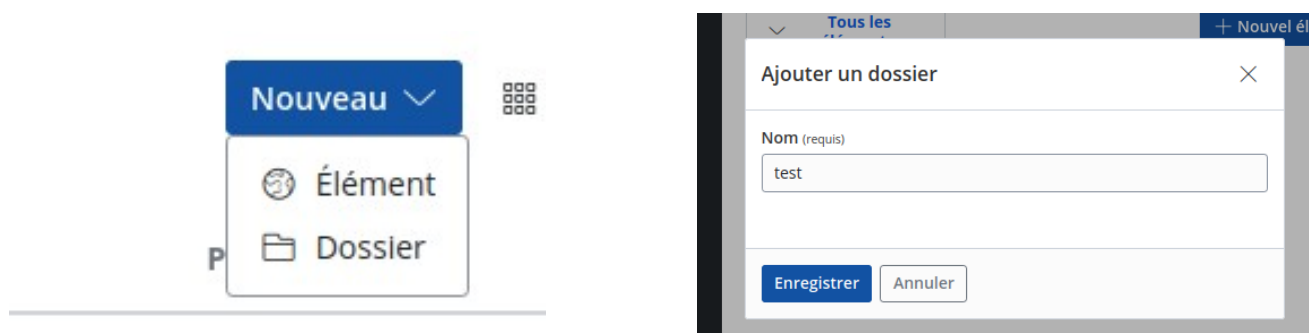
Créez un compte

Vous avez déjà un compte ? [Se connecter](#)

Un formulaire tout ce qu'il y a de plus classique, bien qu'une intention toute particulière soit bien évidemment donnée au master password, qui se doit d'être particulièrement robuste (longueur, complexité...). Une fois ce compte créé, nous allons pouvoir nous connecter pour accéder au « coffre » et tester les fonctionnalités de Vaultwarden.



Nous voici enfin dans le vif du sujet. Commençons par créer un nouveau dossier afin de stocker notre premier mot de passe.

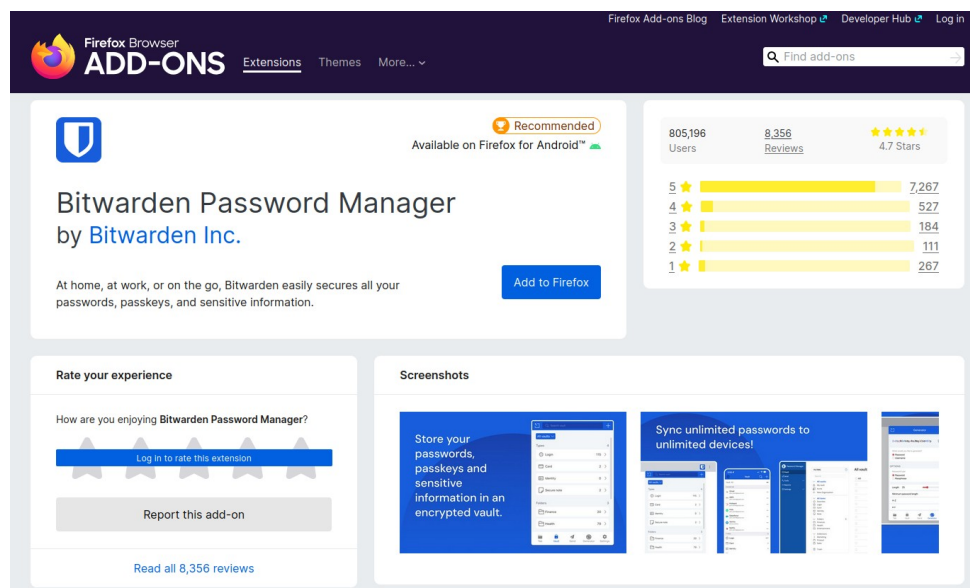


Une fois le dossier créé, nous pouvons y ajouter un nouvel élément, ici donc un mot de passe et un identifiant.

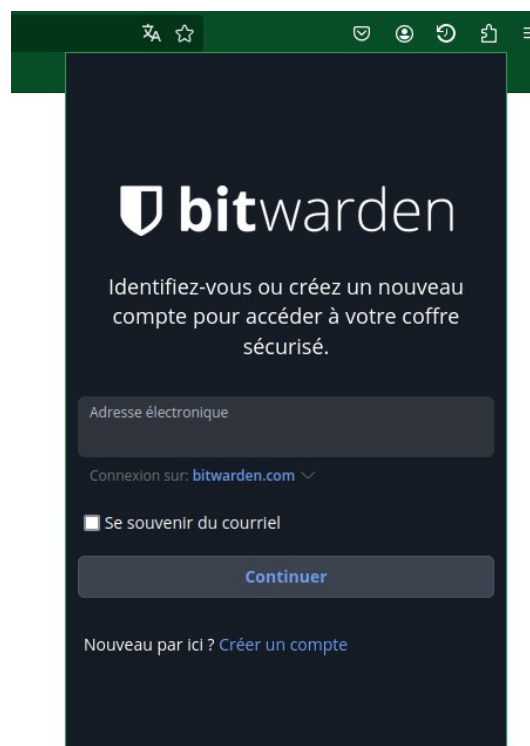
Bien évidemment ce mot de passe est uniquement prévu pour le test. D'ailleurs, je mentionne également une URL en localhost : c'est une page web custom tout ce qu'il y a de plus basique, servie par un serveur minimaliste.



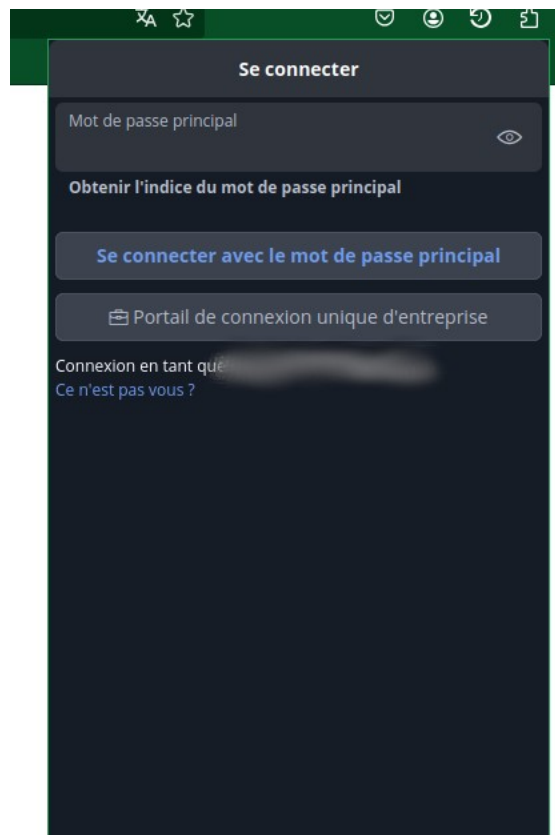
Je peux constater que mon premier élément est bien présent. Il est à présent l'heure d'installer l'extension Bitwarden pour navigateur (ici Firefox). Nous utilisons l'extension Bitwarden, vu que Vaultwarden est un fork de Bitwarden cette extension est compatible.



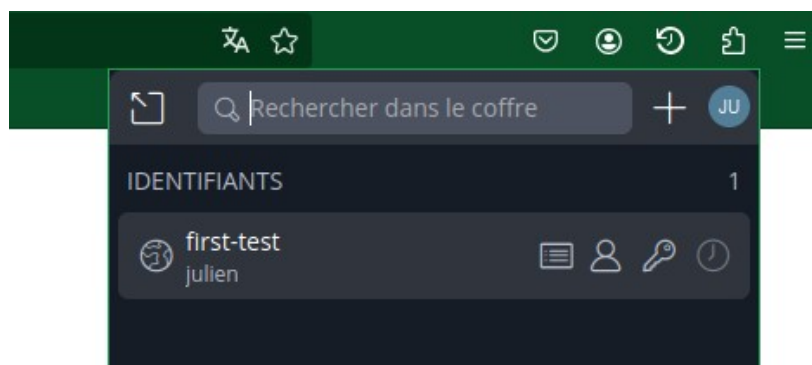
L'installation terminée, nous pouvons nous connecter à notre coffre via l'extension.



L'adresse électronique est l'adresse que j'ai renseignée lors de la création de mon compte. Attention à bien modifier la valeur au niveau de la connexion : je ne veux pas utiliser « bitwarden.com », mais bien mon propre domaine. Je vais donc choisir l'option « auto-hébergé » pour pouvoir le renseigner.



Je peux finalement renseigner mon master password pour me connecter à mon coffre, et constater la présence de l'élément que j'ai ajouté via l'interface graphique de mon instance de Vaultwarden



Je passe maintenant sur ma petite page web de test. Lorsque je clique sur le champ password, je vois bien que l'extension me propose des valeurs à utiliser.



Je clique et les deux champs sont remplis automatiquement. Le petit bouton spoil ne sert qu'à afficher en clair le mot de passe ajouté. On constate que c'est bien le même que précédemment renseigné dans le gestionnaire de mots de passe. L'extension fonctionne donc à merveille.

Identifiant

Mot de passe

## Configuration - sécurisation :

L'installation et l'utilisation sont une réussite. Mais il est possible de configurer un peu plus pour bénéficier de la fonctionnalité d'envoi de mails mais aussi sécuriser l'accès aux données.

Je vais donc configurer l'URL du site et la gestion des mails. Cela se fait dans la partie admin de Vaultwarden, partie admin atteignable à condition de spécifier un token dans les variables d'environnements. De retour sur mon serveur distant donc.

```
julien@ionos-server:~/vaultwarden$ sudo docker-compose down
[sudo] password for julien:
Stopping caddy ... done
Stopping vaultwarden ... done
Removing caddy ... done
Removing vaultwarden ... done
Removing network vaultwarden_default
julien@ionos-server:~/vaultwarden$
```

J'arrête les conteneurs docker pour mettre à jour mon « docker-compose.yml ».

```
environment:
  - WEBSOCKET_ENABLED=true
  - ADMIN_TOKEN=ici_mon_token
volumes:
  - ./vw-data:/data
```

Cette modification de fichier aurait également pu être l'occasion de mettre les variables d'environnement « SIGNUPS\_ALLOWED », « INVITATIONS\_ALLOWED » et « SHOW\_PASSWORD\_HINT » à « false » pour respectivement empêcher les nouvelles inscriptions, les invitations et la possibilité de voir un indice concernant son master password (c'est toujours un peu risqué). N'ayant pas encore invité tous les futurs membres de mon instance je le ferais donc plus tard. Et concernant les indices, hélas, compte tenu de mes utilisateurs à venir il va falloir sans doute que je les laisse...



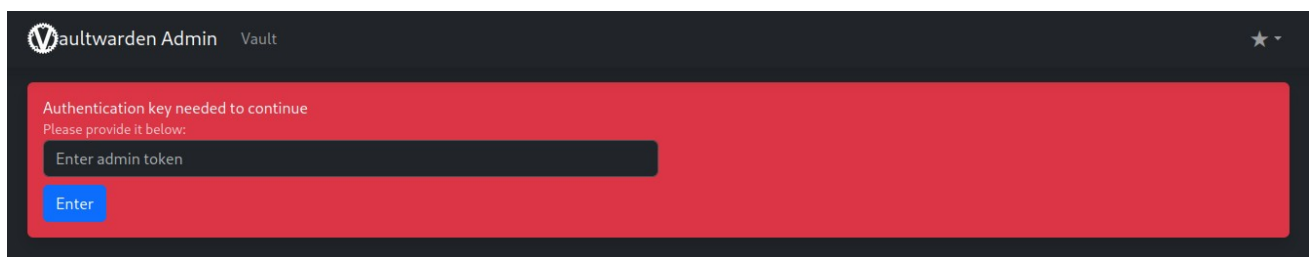
Ceci étant fait je relance les conteneurs

```
julien@ionos-server:~/vaultwarden$ sudo docker-compose up -d
Creating network "vaultwarden_default" with the default driver
Creating vaultwarden ... done
Creating caddy ... done
julien@ionos-server:~/vaultwarden$
```

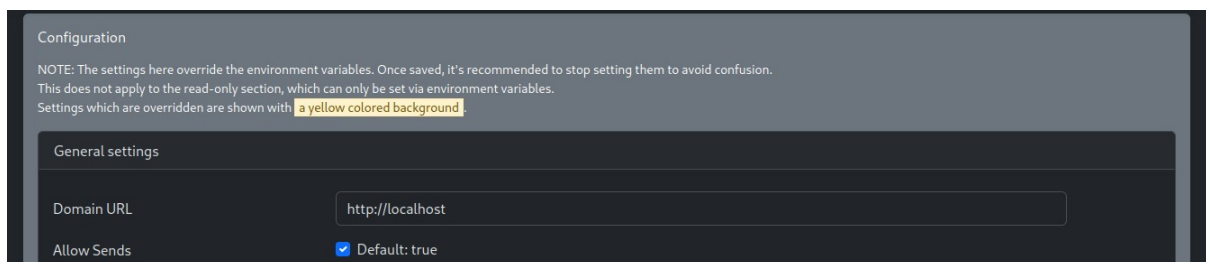
Bien que j'interdirais plus tard l'inscription de nouveaux utilisateurs, j'ai quand même testé la fonctionnalité...



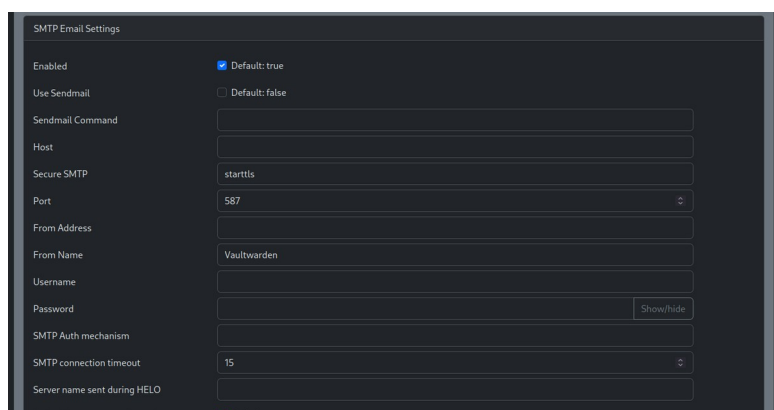
Je vais maintenant me connecter sur « [mon\_domain]/admin » pour y trouver l'interface d'admin.



Cette interface me demande bien un token, token que je lui ai fourni juste avant. Je m'en sers pour me connecter.



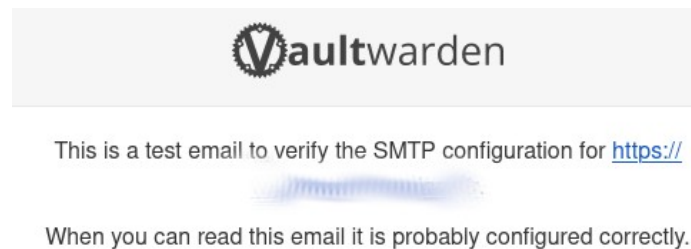
Je commence par configurer l'URL du domaine



Puis le SMTP (au terme d'une longue bataille, j'ai eu pas mal de soucis de pare-feu suite à des oublis malencontreux de ma part...)

Quitte à devoir ouvrir le port SMTP, j'en profite aussi pour modifier le port SSH. 22 est un peu trop classique...

Vaultwarden nous donne la possibilité bien pratique de faire un petit test en direct lorsque la configuration SMTP est terminée. Je teste donc et...



...je reçois (enfin) le mail qui me prouve que la configuration SMTP est parfaitement opérationnelle.

Au passage, comme je l'ai mentionné auparavant, l'utilisation des volumes de docker est sensé pouvoir rendre les datas de l'application persistantes. Dans les faits j'ai coupé les conteneurs et je les redémarrés, je vous fait grâce de la capture d'écran redondante, mais mon « fake-pwd » est toujours là, preuve que mes datas sont correctement conservées bien au chaud sur le disque dur de mon serveur.

## Création d'une organisation :

Je vais maintenant passer à la gestion d'un des objectifs principaux de mon implémentation : le partage de données.

Je commence par créer une organisation qui m'appartient.

### Informations générales

Nom de l'organisation (requis)

test-team

Courriel (requis)

Puis je crée un nouvel utilisateur. Ce nouvel utilisateur sera administrateur de l'organisation. Je le crée de façon classique et cette fois-ci, grâce à la configuration SMTP je reçois un mail à l'adresse associée. Cet utilisateur créé, je retourne dans mon propre espace et invite ce nouvel utilisateur par le biais de son adresse email.



## Inviter un membre

Rôle

Collections

Invitez un nouvel utilisateur dans votre organisation en saisissant l'adresse électronique de son compte Bitwarden ci-dessous. S'il n'a pas encore de compte Bitwarden, il lui sera demandé de créer un nouveau compte.

**Courriel** (requis)

Saisissez jusqu'à 20 courriels en les séparant par une virgule.

**Rôle du membre** ?

☒ **Utilisateur**

Un utilisateur normal avec accès aux collections de votre organisation.

☐ **Gestionnaire**

Les gestionnaires peuvent voir et gérer les collections de votre organisation qui leur ont été assignées.

☐ **Administrateur**

Les administrateurs peuvent voir et gérer tous les éléments, les collections et les utilisateurs de votre organisation.

☐ **Propriétaire**

L'utilisateur avec l'accès le plus élevé qui peut gérer tous les aspects de votre organisation.

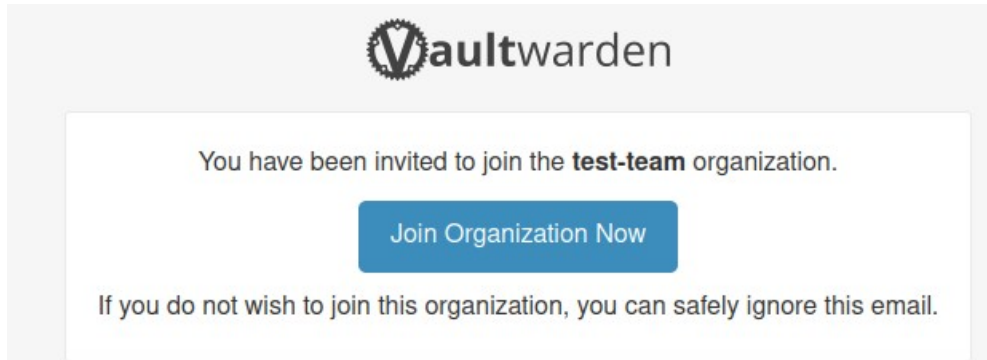
**Identifiant externe**

L'identifiant externe peut être utilisé comme référence ou pour lier cette ressource à un système externe tel qu'un répertoire utilisateur.

Enregistrer

Annuler

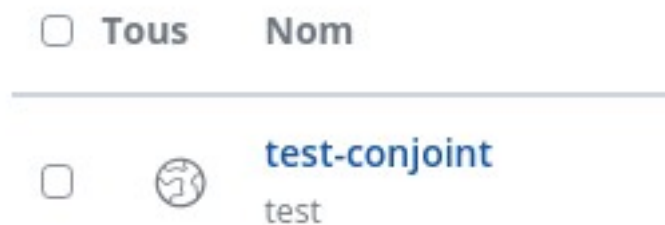
Une fois l'invitation lancée, mon utilisateur reçoit un mail pour le notifier de l'événement.



Bien que cet utilisateur ait été invité, il me faut encore, en tant que propriétaire de l'organisation, le confirmer.



Puisque mon utilisateur a été confirmé, je décide d'accéder à la page de cette organisation en cliquant en bas à gauche de l'interface sur « Admin Console ». Je crée un nouvel élément dans la collection par défaut.



Là aussi je vous fait grâce de l'image redondante : cet élément est bien sûr lui aussi présent du côté de mon nouvel utilisateur de test.

## Sauvegarde :

Toute cette installation commence à fonctionner de façon tout à fait correcte. Mais avant de lâcher mes utilisateurs sur l'outil, il faut penser à la sauvegarde : il serait dommage de perdre 2 ou 3 ans de mots de passe sur un crash serveur un peu méchant.

Je commence par créer un utilisateur « vaultwarden » sur mon serveur de backup, avec un home dédié et sans droits sudo. Dans ce home je crée un dossier « backups ».

```
vaultwarden@~$ ls
vaultwarden@~$
```

Je vais avoir besoin de lancer une commande « scp » sans pouvoir renseigner le mot de passe SSH. Je vais donc créer des clefs SSH pour pouvoir me connecter depuis mon serveur vers mon serveur de backup sans mot de passe.

```
julien@ionos-server:~$ ssh-keygen -t ed25519
```

Une fois la paire de clefs générée, je copie ma clef publique sur le serveur de destination. J'en profite en passant pour configurer le pare-feu de mon serveur de backup (et le port SSH bien sûr)

```
julien@ionos-server:~$ ssh-copy-id -p 22 -i id_ed25519.pub vaultwarden@
```

Pour effectuer cette sauvegarde, j'ai fait le choix d'un script bash. (détails de save\_dump.sh en annexe).

Après avoir défini plusieurs variables utiles au fonctionnement, ce script effectue un dump de la base SQLite, l'envoie via scp au serveur de backup, supprime sur le serveur d'origine le fichier devenu inutile et consigne le nom du fichier ainsi que la date et l'heure dans un fichier de logs (ainsi que les erreurs éventuelles). Ici aussi on peut constater de nouveau l'intérêt des volumes docker : je peux réaliser ce dump et le manipuler comme si l'instance de Vaultwarden était directement installée sur mon disque dur.

Attention cependant aux droits : mon dump est généré à partir de fichiers présents dans un dossier dont le propriétaire est root (créé par Docker). Il faut donc que mon script bash soit lancé avec des droits root, script qui doit utiliser les variables d'environnements de root. De la même façon, la connexion SSH doit se faire par l'intermédiaire de l'utilisateur root, c'est donc mon utilisateur root qui va générer les clefs SSH. Pour finir c'est bien un crontab root qui déclenchera régulièrement la sauvegarde.

En parlant de crontab, il est temps de l'installer

```
julien@ionos-server:~/vaultwarden$ sudo apt-get install cron
```

Puis de le configurer.

```
julien@ionos-server:~/vaultwarden$ sudo crontab -e
```

Pour ce premier essai, la commande sera lancée à 2H. Par la suite je ne le lancerais qu'une fois par semaine, ce qui sera amplement suffisant.

```
## Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow command
0 12 * * * /home/julien/vaultwarden/save_dump.sh
```

Une nuit plus tard, je reviens et c'est courageusement que je me résigne à supprimer mon fidèle « fake-pwd »



Je dois maintenant pouvoir le récupérer. Je me connecte sur mon serveur pour voir s'il y a du nouveau.

```
julien@ionos-server:~/vaultwarden$ cat backup.logs
- 2024/11/23 02H00
backup_vaultwarden_20241123020001.db
julien@ionos-server:~/vaultwarden$
```

En effet un fichier « backup.logs » est apparu contenant deux lignes témoignant de l'action de mon cron dans la nuit. Tout porte à croire que tout s'est bien passé. Allons voir du côté du serveur de backup.

```
vaultwarden@ionos-server:~/backups$ ls
backup_vaultwarden_20241121120001.db
vaultwarden@ionos-server:~/backups$
```



Mon dossier initialement vide affiche bien maintenant un nouveau fichier. Je vais le transférer sur le serveur d'origine.

```
julien@ionos-server:~$ scp -P 22 vaultwarden@193.50.135.100:/home/vaultwarden/backups/backup_vaultwarden_20241121120001.db ./vaultwarden/backups/backup_vaultwarden_20241121120001.db
100% 260KB 1.4MB/s 00:00
julien@ionos-server:~$
```

Ce dump récupéré, il va falloir maintenant le tester (on ne peut pas garantir qu'un dump fonctionne à tous les coups)

```
julien@ionos-server:~/vaultwarden/vw-data$ sudo sqlite3 db.sqlite3 ".restore backup_vaultwarden_20241121120001.db"
julien@ionos-server:~/vaultwarden/vw-data$
```

Comment savoir si ça a fonctionné ou pas ? Allons voir ça directement dans le coffre-fort.

ÉDITER L'ÉLÉMENT

Nom: first-test

Dossier: test

Nom d'utilisateur: julien

Mot de passe: fake-pwd

Clé d'authentification (TOTP): 15

URI 1: https://equinoz.bzh

Détection de correspondance: Détection de correspondance par c

Notes:

CHAMPS PERSONNALISÉS

Nouveau champ personnalisé

Texte

Mis à jour: 21 nov. 2024, 12:45:28

Créé: 21 nov. 2024, 08:32:47

OPTIONS

Enregistrer Annuler

Et c'est avec émotion que je retrouve mon « fake-pwd » intact ! A 7€ mensuels le premier pack de sauvegarde, je rentabilise largement mon petit serveur de backup qui, lui, me permettra de faire autre chose !

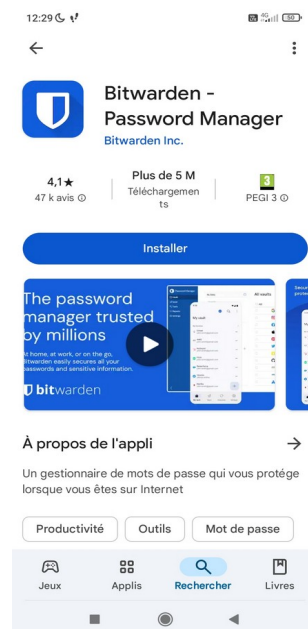
Dernier point concernant la sauvegarde : si je veux respecter la règle 3-2-1 qui stipule que je dois avoir au moins trois copies de mes données, deux des sauvegardes stockées sur des types de supports différents, et au moins une sauvegarde stockée hors site ou sur le cloud, je dois faire régulièrement une troisième sauvegarde sur un disque dur par exemple. Mes deux serveurs ne sont pas localisés au même endroit et ce n'est pas le même hébergeur.

## Utilisation de l'application mobile :

Un point crucial reste à explorer : l'utilisation de Vaultwarden sur appareil mobile. En effet, mes utilisateurs utilisant presque exclusivement leur téléphone, je dois pouvoir leur montrer que cette solution est facilement exploitable sur mobile.

Je vais avoir recours à la magnifique page web déjà utilisée dans la partie nommée « Utilisation ». Pour l'occasion, je vais l'héberger sur l'URL <https://equinoz.bzh>, où vous pourrez encore la trouver avec un peu de chance, si elle n'a pas déjà été remplacée par un autre test grossier. D'ailleurs vous pouvez remarquer que dans la dernière capture d'écran j'ai modifié l'URL associée à mon élément.

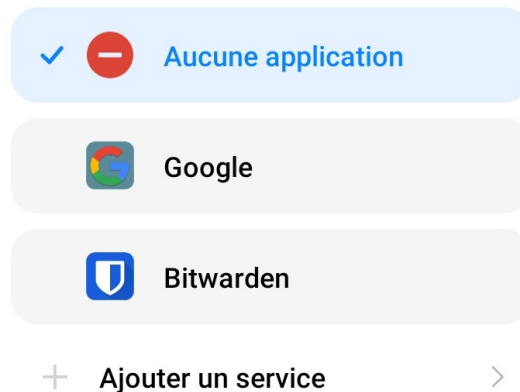
Tout comme l'extension pour navigateur, l'application utilisable avec Vaultwarden est l'application Bitwarden. Je la télécharge donc depuis le store Android.



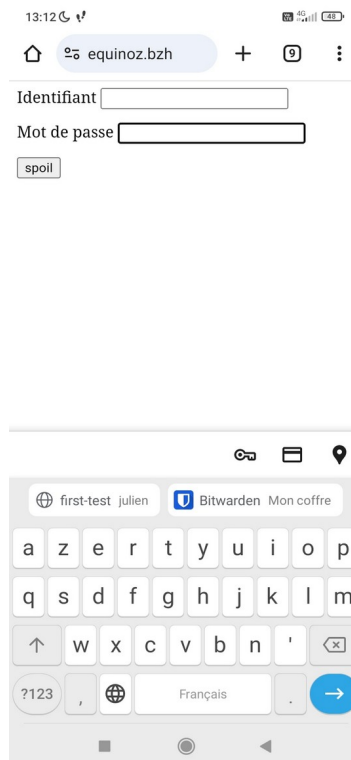
Je ne vais malheureusement pas pouvoir prendre de capture d'écran de l'application en elle-même. Ou plutôt heureusement : en toute cohérence, l'application interdit les captures pour des raisons évidentes de sécurité. Comme pour l'extension je me connecte avec mes identifiants en prenant garde de bien renseigner la connexion auto-hébergée.

Si je veux pouvoir disposer des informations de l'application au cours de ma navigation, je dois autoriser Vaultwarden à utiliser le service de saisie automatique Android. Cela se fait dans « Paramètres » en bas à droite de l'écran principal de l'app. Sur l'écran suivant, cliquez en restant appuyé sur « Services de saisie automatique », ce qui vous permettra de donner cette autorisation dans l'interface Android en cliquant sur « Bitwarden ».



## Service de saisie automatique








Maintenant que cela est fait, je me rend à l'URL <https://equinoz.bzh> et je peux constater que mon téléphone me fait une suggestion pour remplir les champs.



Je clique sur la proposition et je retrouve bien sûr mon indéboulonnable « fake-pwd »

13:12  

  equinoz.bzh   

---

Identifiant

Mot de passe



## Conclusion

Pour commencer j'avais initialement pensé à installer Passbolt... Mais j'ai pas mal galéré, l'installation m'a semblé loin d'être simple, et j'ai finalement laissé tomber après plusieurs heures de tentatives pour régler le problème de gestion des clefs GPG. Je me suis finalement rabattu sur Vaultwarden... qui s'est avéré être bien plus intéressant !

Je connaissais déjà Docker, mais il faut reconnaître que c'est quand même très efficace. Le plus long avec l'installation de Vaultwarden est l'installation de Docker. Une fois que c'est fait, monter le conteneur qui va bien n'est vraiment qu'une formalité ! Parallèlement, l'utilisation de Caddy en tant que reverse-proxy permet de configurer du TLS d'une façon si simple que je ne soupçonnais pas que ça puisse être aussi facile.

Qui plus est, j'ai aussi beaucoup apprécié que Vaultwarden fonctionne avec SQLite : une base MariaDB c'est encore des mots de passe à retenir, mots de passe qu'il est plutôt peu judicieux d'enregistrer sur la DB sur laquelle on est sensé avoir à intervenir en cas de pépin... Bien sûr avec les serveurs de l'instance Vaultwarden et de backup il y a des informations confidentielles à gérer, mais dans le cas de Passbolt ça aurait aussi été le cas de toutes façons. Qui plus est, la création d'un dump à partir de SQLite est vraiment très simple.

Ce qui m'a le plus posé problème au final c'est les à-côtés. J'ai eu beaucoup de mal à établir des connexions SSH sur un port autre que le 22, car j'avais oublié que j'avais configuré le pare-feu de mon propre appareil de façon à ne permettre le SSH que sur certains ports bien précis... La gestion des droits pour mon script bash a été l'occasion de nombreux tests et erreurs aussi. J'ai du chercher également quelles étaient les informations requises pour configurer le serveur SMTP. Bref, des erreurs classiques de réseau ou de droits utilisateurs, mais à aucun moment l'installation et la configuration de Vaultwarden en soi ne m'ont posé problème.

Le seul point que je trouve un peu dommage est le fait que ce fork de Bitwarden ne fournisse pas d'API, c'est quelque chose que j'aurais volontiers exploré, les possibilités offertes par le concept sont plutôt intéressantes. Il existe malgré tout un client CLI qui semble jouer un peu ce rôle-là, c'est un point à regarder de plus près à l'avenir.

Concernant mon installation, deux points me dérangent un peu, points pour lesquelles j'espère trouver tôt ou tard une solution plus satisfaisante :

- Le fait d'utiliser une clef SSH sans passphrase ne me plaît pas beaucoup. Il faudrait que je trouve un moyen de pouvoir automatiser cette commande SCP tout en utilisant malgré tout un mot de passe.
- Les variables de connexion directement dans le script bash ne semble pas non plus être une très bonne pratique. J'ai bien pensé à utiliser le fichier « /etc/profile » pour y consigner ces informations comme des variables d'environnement, mais ce fichier est consultable par tous les utilisateurs potentiels du système. Et même si de fait je suis seul sur ce système, l'idée ne me plaît pas du tout. Il est toujours possible de bloquer ce fichier en lecture, mais ce n'est pas sa vocation première. La solution qui consiste à conserver ces informations dans le script me semble moins gênante, au moins ce fichier n'est consultable que par l'utilisateur concerné. J'aimerais trouver un moyen de configurer cela dans un fichier à part, un peu comme les variables d'environnement que l'on garde uniquement en local comme quand on travaille sur un projet collaboratif via Git par exemple.

IL reste beaucoup à faire et à sécuriser sur mon instance. En dehors de la suppression des invitations et des inscriptions, j'ai aperçu qu'il est possible de gérer des Yubikey, de mettre en place de la double authentification, de configurer du TOTP, la page admin semble avoir beaucoup d'options à explorer... Il va falloir malgré tout trouver le bon équilibre entre sécurité et accessibilité : si mon système devient trop contraignant, je risque de perdre mes futurs utilisateurs !

Edit de dernière minute :

Le temps que je rédige le dossier, Vaultwarden est « en production » depuis déjà quelques temps, suffisamment pour faire un premier retour sur son utilisation au quotidien. Personnellement je me dit que j'aurais dû mettre ça en place bien avant tellement c'est pratique. Ma femme est totalement conquise, comme elle me disait avec humour pendant la période où elle consignait ses mots de passe « Je vais faire un peu de Bitwarden, c'est mon application préférée ! ». Ca représente pour elle un réel soulagement en terme de charge mentale. Mon fils l'a également totalement adoptée, et ma belle-fille n'a pas encore pris le temps mais a bien l'intention de s'en servir à terme. Reste les deux petites (14 ans) qui manifestent un intérêt plus réduit, elles comprennent l'intérêt, mais sans doute que le besoin n'est pas encore si présent pour elles à l'heure actuelle. (même si elles se souviennent bien que c'est là qu'elles trouveront les mots de passe des plateformes de streaming mis à jour !) En bref, je dirais que c'est un réel succès. Mon objectif me semble donc tout à fait atteint !

Côté pro, j'ai appris que notre sys-admin avait l'intention de déployer une instance de Passbolt sur le réseau de l'entreprise ce qui est une excellente nouvelle : je ne veux plus gérer mes mots de passe sans gestionnaire. La mise en place prend un peu de temps, je me fais donc un plaisir de le relancer régulièrement sur le sujet. En entreprise, et surtout une entreprise comme la nôtre, ce genre d'outil est tout bonnement indispensable !!

# Sources

## Sources WEB

### Vaultwarden

- <https://zatoufly.fr/creer-son-serveur-vaultwarden-avec-docker/>
- <https://belginux.com/vaultwarden/>
- <https://rdr-it.com/deployer-vaultwarden-avec-docker/>
- <https://wiki-tech.io/SelfHosted/Bitwarden>
- <https://docs.vultr.com/how-to-install-vaultwarden-on-ubuntu-20-04>

### Docker

- <https://www.ionos.fr/digitalguide/serveur/configuration/installer-docker-sur-debian-11/>

### Dump SQLite

- <https://blog.stephane-robert.info/docs/services/bdd/relationnelles/sqlite/#sauvegarde>

## Sources humaines !

- William, lead dev, qui m'a confirmé que ma façon d'envisager la gestion des sauvegardes était viable
- Steven, sys-admin, avec qui j'ai échangé à propos des forces, faiblesses et enjeux autour des gestionnaires de mots de passe

## Annexes (facultatives)

docker-compose.yml

```
version: '3'

services:
  vaultwarden:
    image: vaultwarden/server:latest
    container_name: vaultwarden
    restart: always
    environment:
      - WEBSOCKET_ENABLED=true
    volumes:
      - ./vw-data:/data

  caddy:
    image: caddy:2
    container_name: caddy
    restart: always
    ports:
      - 80:80
      - 443:443
    volumes:
      - ./Caddyfile:/etc/caddy/Caddyfile:ro
      - ./caddy-config:/config
      - ./caddy-data:/data
    environment:
      - DOMAIN={ici mon domaine}
      - EMAIL={ici mon adresse email pour le certificat SSL}
      - LOG_FILE=/data/access.log
```

## Caddyfile

```
{ici mon domaine}:443 {  
    log {  
        level INFO  
        output file server.logs {  
            roll_size 10MB  
            roll_keep 10  
        }  
    }  
  
    # Get a cert by using the ACME HTTP-01 challenge.  
    tls {ici mon adresse email pour le certificat SSL}  
  
    encode gzip  
  
    # Headers to improve security.  
    header {  
        # Enable HSTS  
        Strict-Transport-Security "max-age=31536000;"  
  
        # Enable cross-site filter (XSS)  
        X-XSS-Protection "1; mode=block"  
  
        # Disallow the site to be rendered within a frame (clickjacking protection)  
        X-Frame-Options "DENY"  
  
        # Prevent search engines from indexing  
        X-Robots-Tag "none"  
  
        # Remove Caddy branding  
        -Server  
    }  
  
    # Redirect notifications to the WebSocket.  
    reverse_proxy /notifications/hub vaultwarden:3012  
  
    reverse_proxy vaultwarden:80 {  
        header_up X-Real-IP {ici mon domaine}  
    }  
}
```



## save\_dump.sh

```
#!/bin/bash

# Identifiants du serveur de destination
NAME="vaultwarden"
IP=ip_serveur_destination
PORT=port_ssh
ORIGIN_PATH="/home/julien/vaultwarden"
ORIGIN_FILE="/vw-data"
TARGET_PATH="/home/vaultwarden/backups"
FILE_LOGS="backup.logs"

# Date, chemin de la base de données, du fichier de sauvegarde et du fichier de logs
DATE=$(date +%Y%m%d%H%M%S)
DB_PATH=$ORIGIN_PATH$ORIGIN_FILE"/db.sqlite3"
BACKUP_PATH=$ORIGIN_PATH"/backup_vaultwarden_"$DATE".db"
LOGS_FILE=$ORIGIN_PATH"/"$FILE_LOGS

# Logs
echo "- $(date +%Y)/$(date +%m)/$(date +%d) $(date +%H)H$(date +%M)" >> $LOGS_FILE

# Exécuter la sauvegarde et rediriger la sortie d'erreur vers le fichier de logs
sqlite3 $DB_PATH ".backup $BACKUP_PATH" >> $LOGS_FILE 2>&1

# Envoi du fichier de backup au serveur de secours via SSH et rediriger la sortie d'erreur vers le fichier de logs
echo "backup_vaultwarden_"$DATE".db" >> $LOGS_FILE
scp -P $PORT $BACKUP_PATH $NAME@$IP:$TARGET_PATH >> $LOGS_FILE 2>&1

# Suppression du fichier de backup, inutile ici
rm $BACKUP_PATH
```