le cnam
Bretagne

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

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 Vautlwarden 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 que je loue déjà chez un autre hébergeur.

J'ai aperçu qu'un serveur Passbolt mettait à disposition une API. Par curiosité je vais tester cette API en tentant de mettre en place un CRUD par le biais d'une application NodeJs en ligne de commande. Ce n'est pas la technologie la plus adéquate pour ça, mais Javascript est ma zone de confort et comme le développement de l'application en elle-même n'est pas l'objet de ce dossier, NodeJs fera très bien l'affaire.

Ce que je vais mettre en place en précisant les objectifs

- Comment je vais m'y prendre

Comment je vais tester que j'ai atteint ou non mes objectifs

Schéma de principe

Par le bias d'un « dessin » et éventuellement d'une explication complémentaire je décrit mon lab. Je peux notamment faire apparaitre :

- Le ou les serveurs
- Le ou les applications
- Les IP ou nom
- Les flux et protocoles
- Les fichiers et/ou dossiers notables

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 crée 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 s
oftware-properties-common
```

On télécharge ensuite la clef GPG depuis le référentiel Docker. Cette clef permet de s'assurer que les paquets proviennent d'un 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/debi
an $(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
```

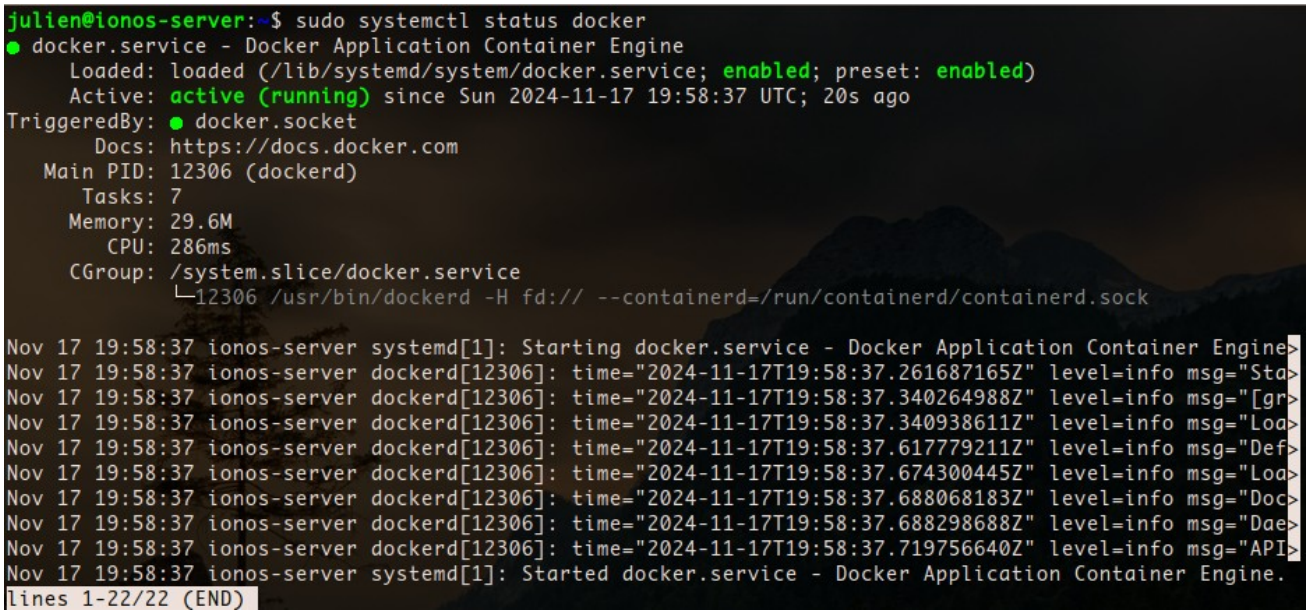
On accède à la stratégie de cache pour vérifier que les paquets d'installation proviennent bien de Docker.

```
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



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
```

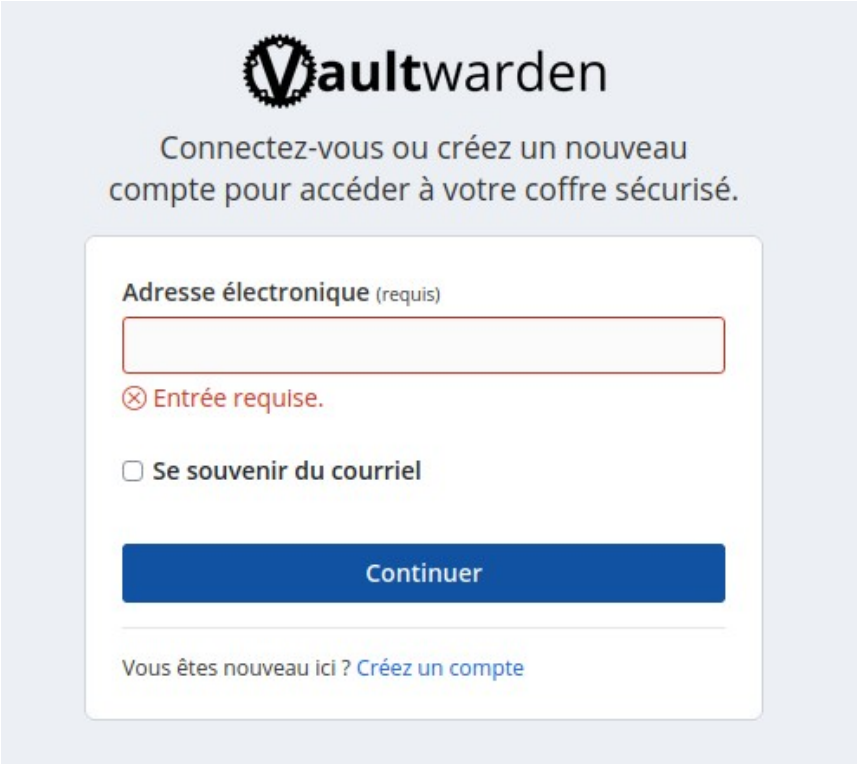
La configuration des conteneurs est terminée. C'est maintenant l'heure de vérité. On monte les conteneurs...

```
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
               PORTS
        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$
```

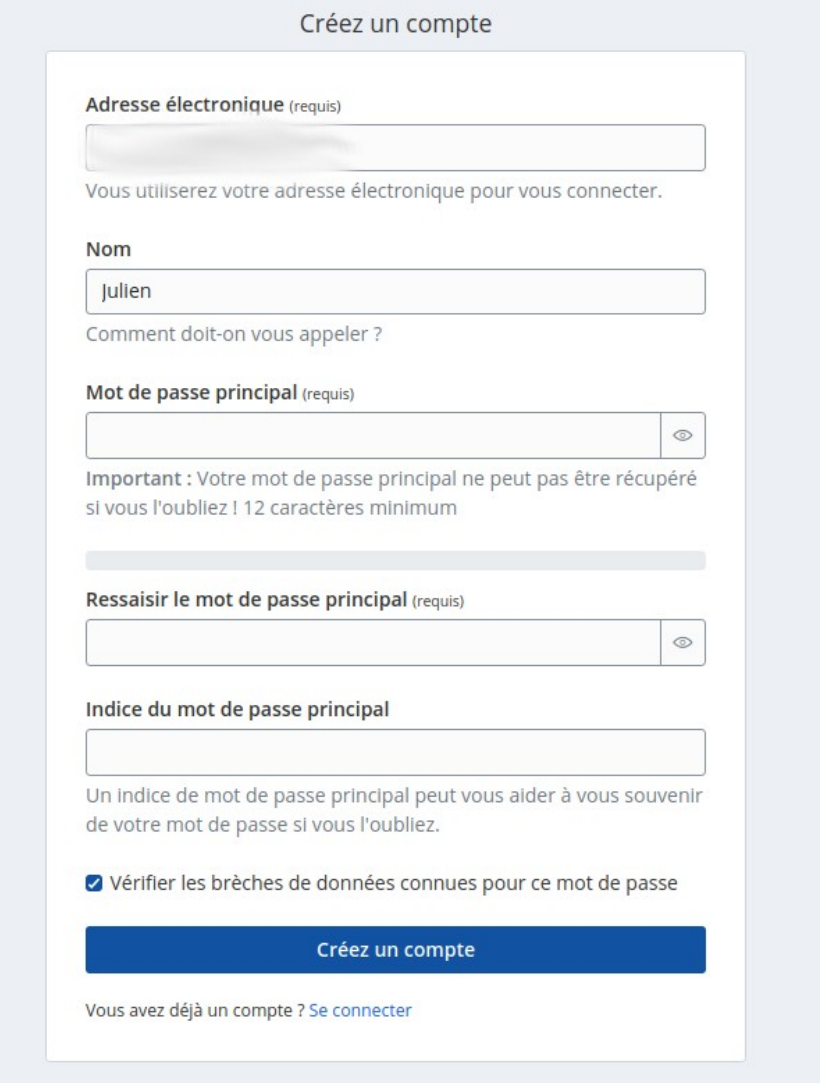
Tout semble correct ! Rendons nous sur l'URL dédiée à notre VPS pour voir ce que ça raconte :



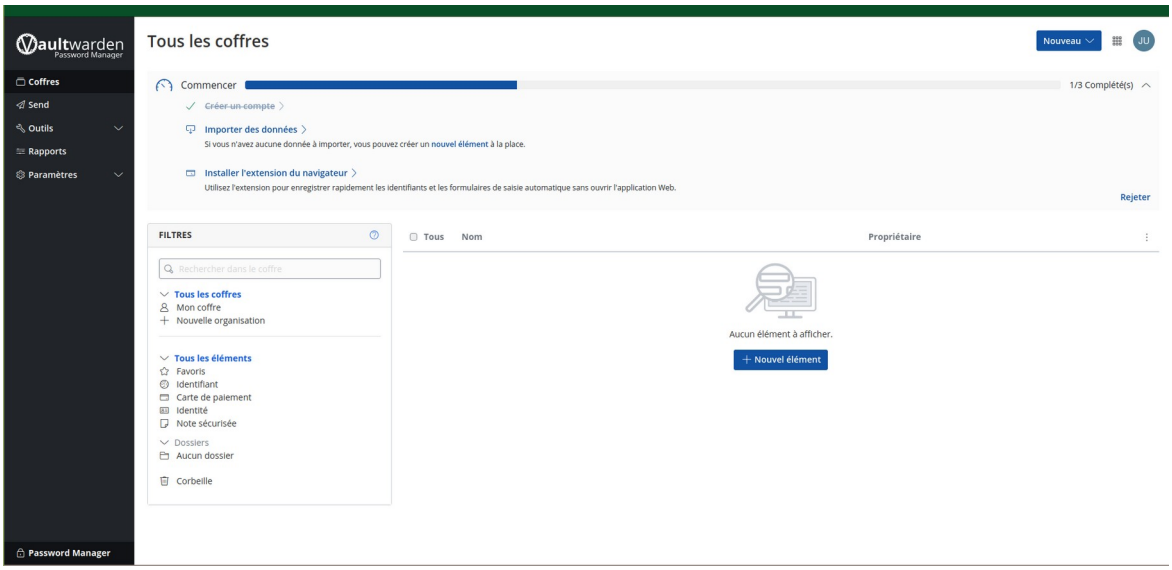
Et cette installation est un succès tout ce qu'il y a de plus manifeste !

Utilisation :

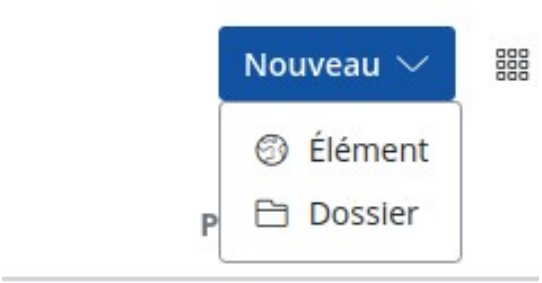
Comme la page web nous invite à créer un compte, nous allons donc nous y coller.

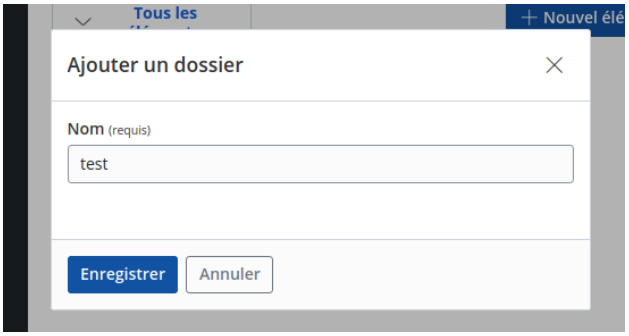


Un formulaire tout ce qu'il y a de plus classique, bien qu'une intention toute particulière soit bien évidemment donnée au master password, qui se doit d'être particulièrement robuste (longueur, complexité…). Une fois ce compte créé, nous allons pouvoir nous connecter pour accéder au « coffre » et tester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

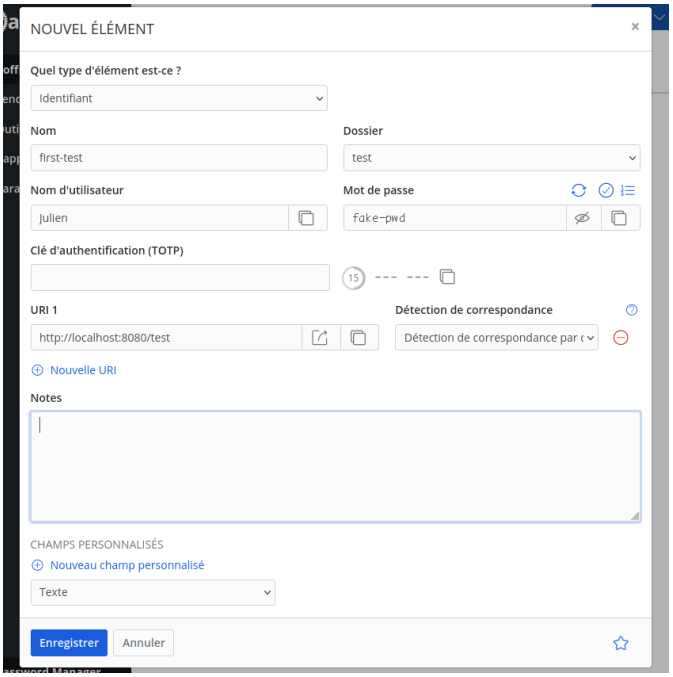


Nous voici enfin dans le vif du sujet. Commençons par créer un nouveau dossier afin de stocker notre premier mot de passe.



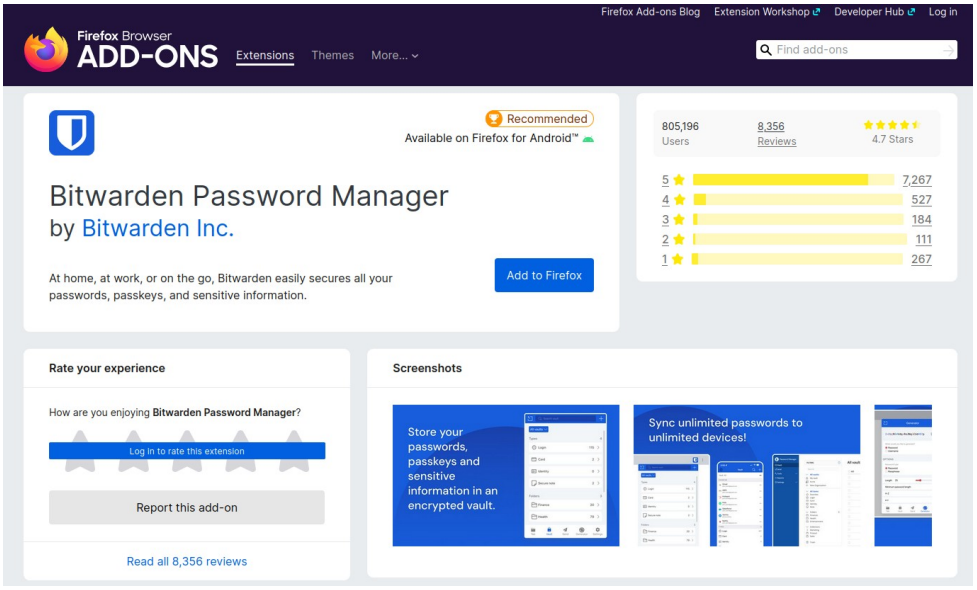


Une fois le dossier créé, nous pouvons y ajouter un nouvel élément, ici donc un mot de passe et un identifiant.

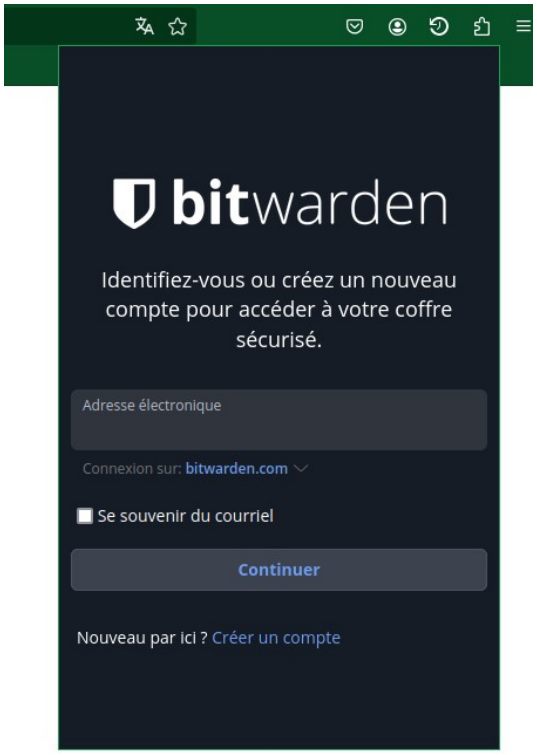


Bien évidemment ce mot de passe est uniquement prévu pour le test. D'ailleurs, je mentionne également une URL en localhost : c'est une page web custom tout ce qu'il y a de plus basique, servie par un serveur minimaliste.

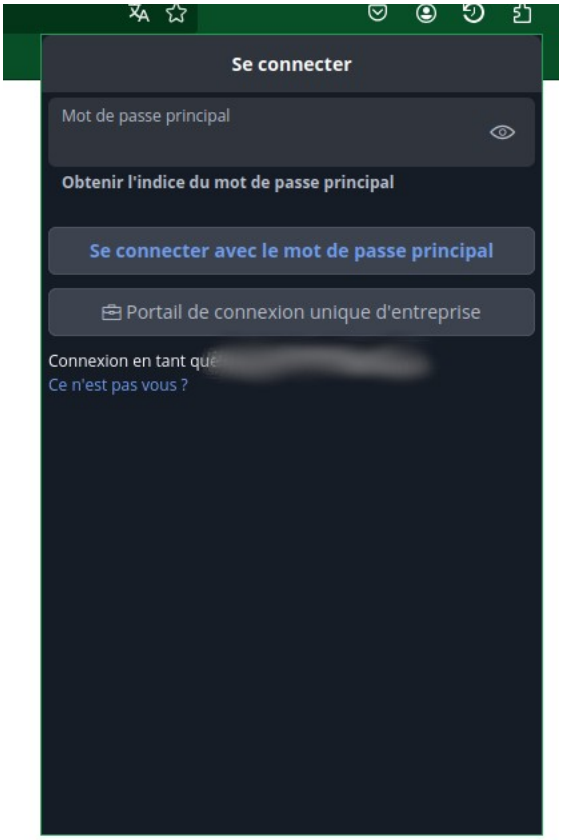
Je peux constater que mon premier élément est bien présent. Il est à présent l'heure d'installer l'extension Bitwarden pour navigateur (ici Firefox). Nous utilisons l'extension Bitwarden, vu que Vaultwarden est un fork de Bitwarden cette extension est compatible.



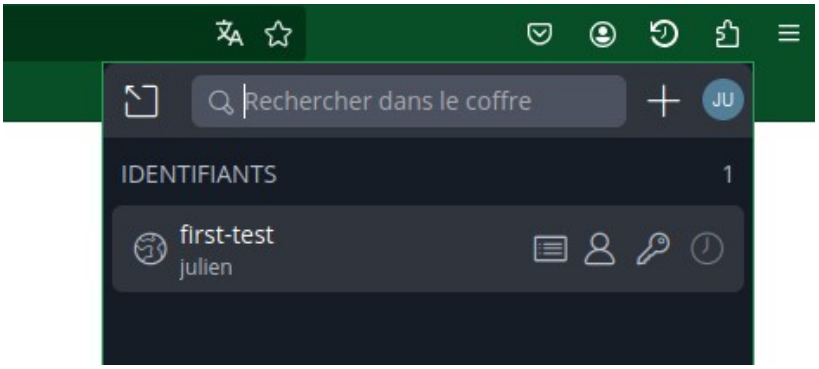
L'installation terminée, nous pouvons nous connecter à notre coffre via l'extension.



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.

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
```

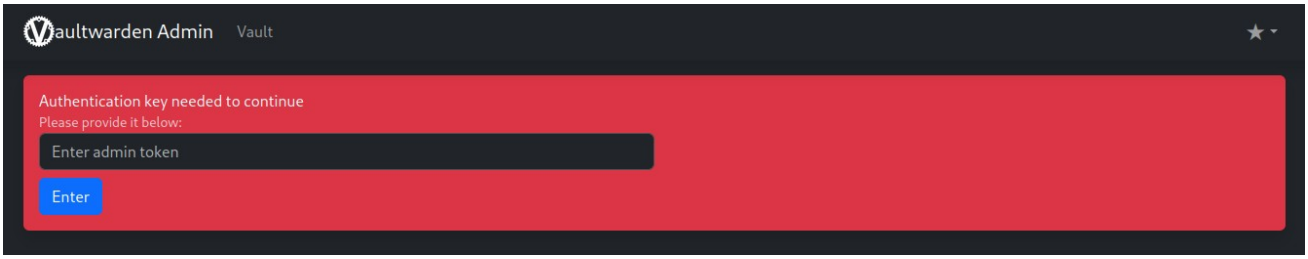
Cette modification de fichier aurait également pu être l'occasion de mettre les variables d'environnement « `SIGNUPS_ALLOWED` », « `INVITATIONS_ALLOWED` » et « `SHOW_PASSWORD_HINT` » à « `false` » pour respectivement empêcher les nouvelles inscriptions, les invitations et la possibilité de voir un indice 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

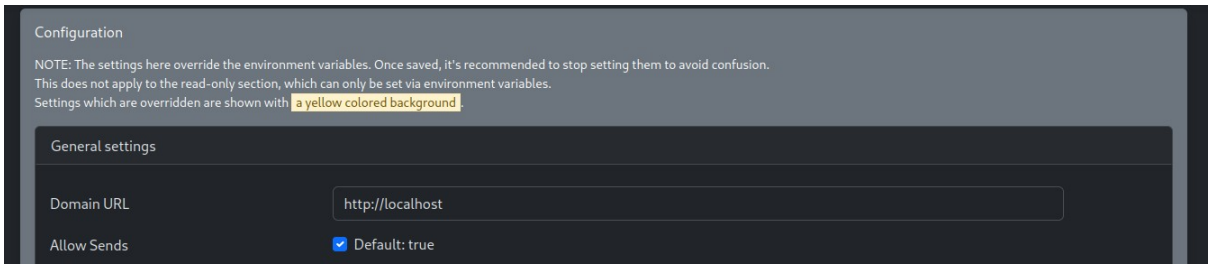
```
julien@ionos-server:~/vaultwarden$ sudo docker-compose up -d
Creating network "vaultwarden_default" with the default driver
Creating vaultwarden ... done
Creating caddy       ... done
julien@ionos-server:~/vaultwarden$
```

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

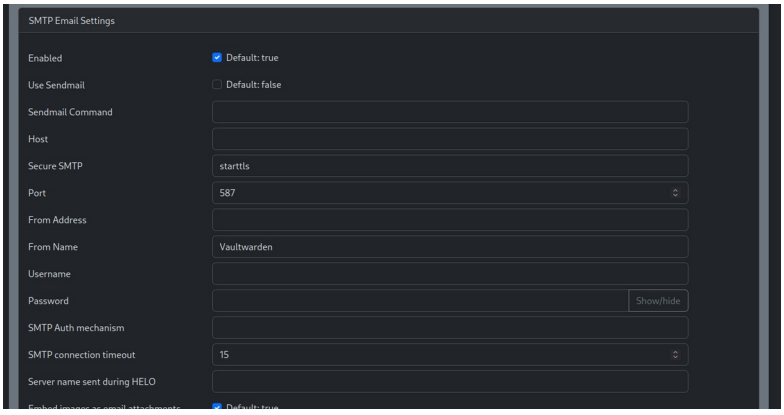
Je vais maintenant me connecter sur « [mon_domain]/admin » pour y trouver l'interface d'admin.



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



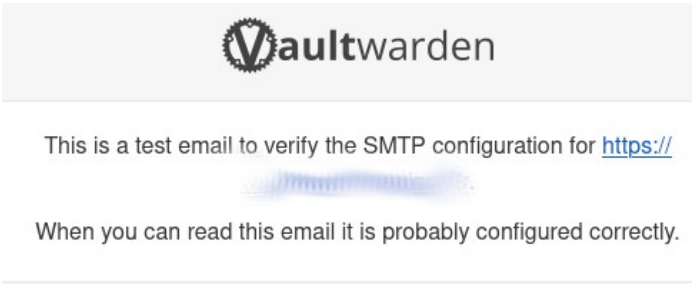
Je commence par configurer l'URL du domaine



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.

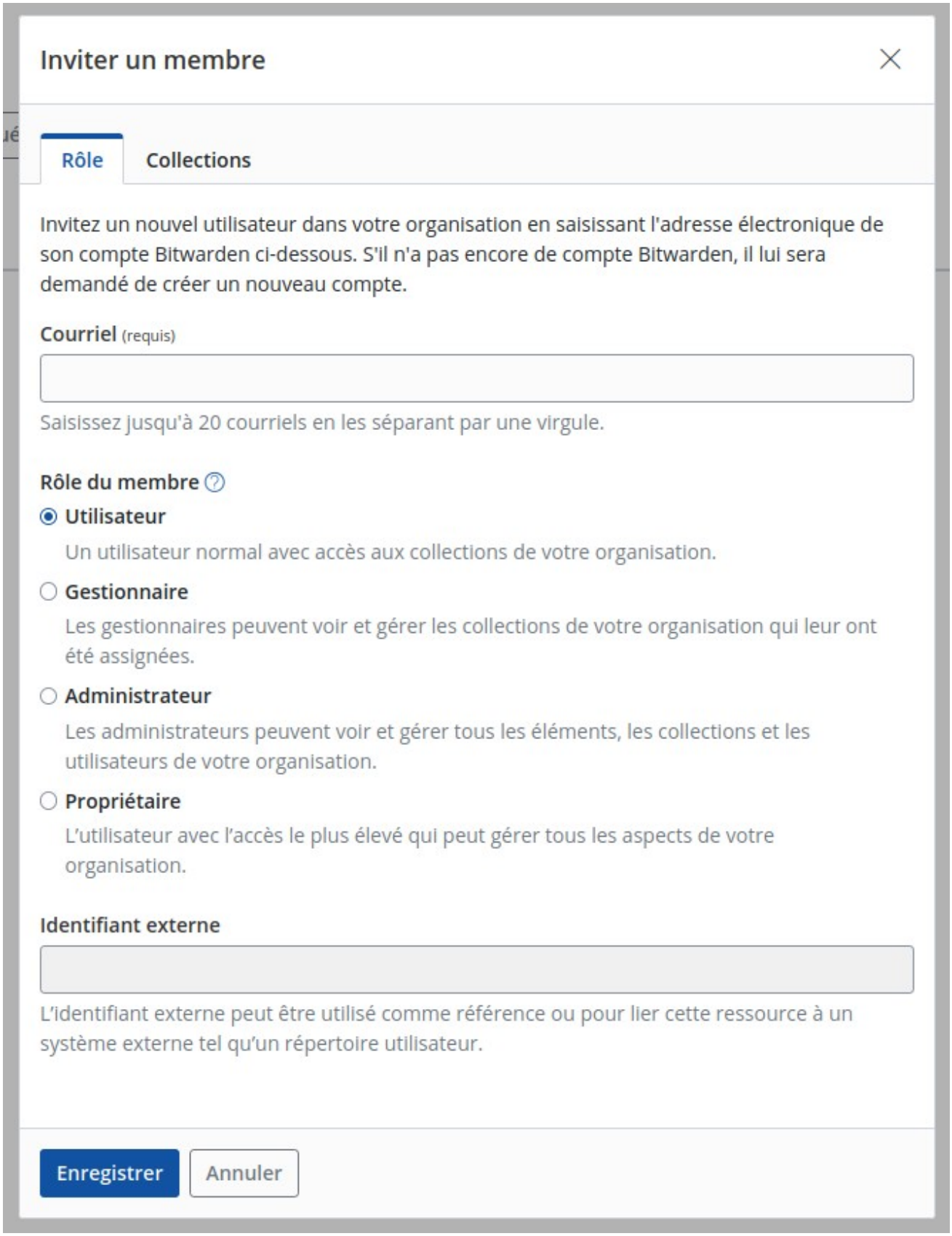
Au passage, comme je l'ai mentionné auparavant, l'utilisation des volumes de docker est sensé pouvoir rendre les datas de l'application persistantes. Dans les faits j'ai coupé les conteneurs et je les redémarrés, je vous fait grâce de la capture d'écran redondante, mais mon « fake-pwd » est toujours 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.

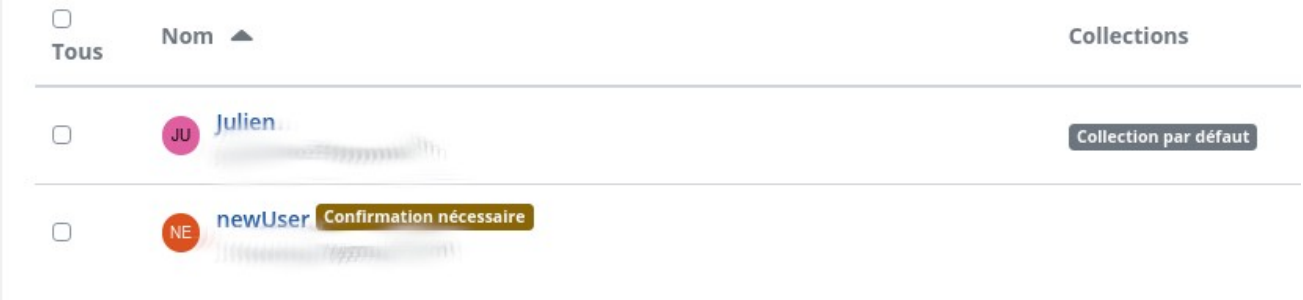
Je commence par créer une organisation qui m'appartient.

Puis je créé un nouvel utilisateur. Ce nouvel utilisateur sera administrateur de l'organisation. Je le crée de façon classique et cette fois si, 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.



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

Bien que cet utilisateur ait été invité, il me faut encore, entant 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.

-Sauvegarde (tester dump)
-Utilisation avec appli
-Consommation API ?

API
https://bitwarden.com/help/vault-management-api/

Conclusion

Sécuriser, Yubikey toussa...

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 papier

-

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)

Lister ici les fichers de configuration, les version d'OS et applications utilisés ou tout autre éléments que vous jugez utiles.

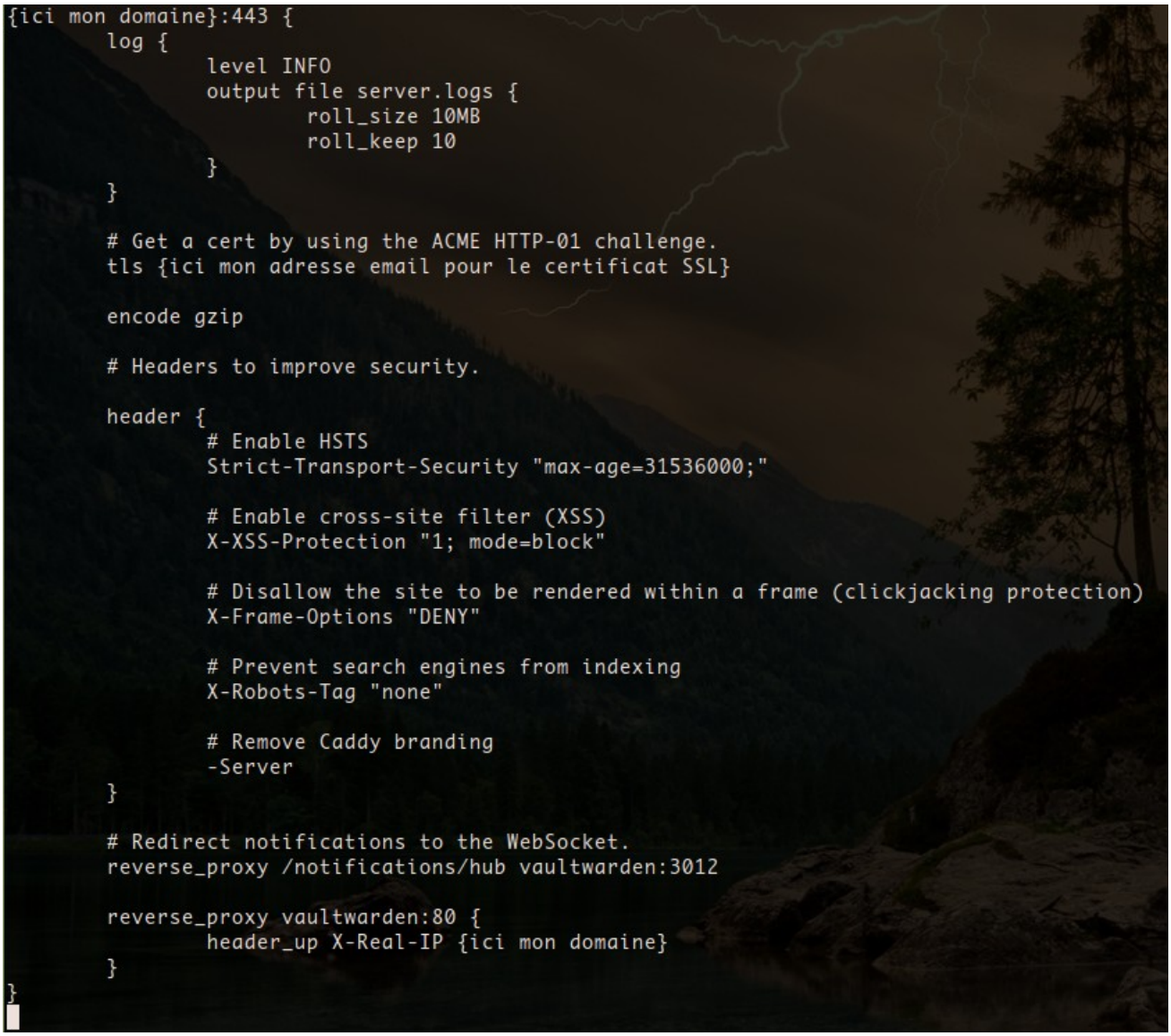
docker-compose.yml

```
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}
        }
}
```