Etablissement

Charles Carnus

Entreprise

Raynal & Roquelaure

Rapport de Stage

BTS CIEL

Luc Tournié

2025

Sommaire

Présentation de l’entreprise

Raynal & Roquelaure

Fondée en 1876 à Capdenac, dans l’Aveyron, Raynal et Roquelaure est une entreprise française historique de l’agroalimentaire, spécialisée dans la fabrication de plats cuisinés en conserve. Depuis plusieurs décennies, Raynal et Roquelaure s’est imposée comme l’un des leaders du marché des plats cuisinés appertisés en France. Raynal et Roquelaure fait aujourd’hui partie du groupe Cofigeo, un acteur majeur de l’agroalimentaire français

Chiffre d’affaires (2022) : 131,4 M €

Effectif total : environ 306 salariés

Cofigeo

Le groupe Cofigeo, dirigé par Mathieu Thomazeau, est basé à Issy-les-Moulineaux. Il a fortement développé son activité ces dernières années grâce à une stratégie de croissance externe et à l’intégration de marques patrimoniales de qualité.

Raynal et Roquelaure bénéficie ainsi du soutien industriel et logistique du groupe, tout en conservant son identité propre et son ancrage territorial, notamment grâce à ses sites de production implantés dans le sud-ouest de la France.

Activités Réalisés

Matériel de secours

Présentation

Le but de ce matériel de secours est de préparer des ordinateurs portables qui étaient utilisés mais qui on était remplacer par une nouvelle gamme est l'objectif est de pouvoir avec un stock préparer pour des utilité spécifique dans des services important par exemple la comptabilité.

En cas d'incident majeur, comme une cyberattaque ciblant le réseau informatique, ce dispositif permettrait aux services sensibles de continuer à travailler sans interruption. L'enjeu est donc de garantir la continuité d'activité et de limiter l'impact d'une éventuelle attaque sur l'ensemble de l'entreprise.

Préparation du matériels

Dans un premier temps, j'ai procédé à un inventaire du stock d'ordinateurs portables, en vérifiant le bon fonctionnement de chaque poste. Cette étape a permis d'écarter le matériel défectueux et de ne conserver que les équipements opérationnels.

Une fois ce tri effectué, j'ai utilisé la technique du PXE Boot afin de déployer des images Windows sur les ordinateurs sélectionnés. Ce procédé a permis d'automatiser la configuration et de gagner un temps considérable par rapport à une installation manuelle.

Explication du PXE Boot

Le PXE Boot (Preboot Execution Environment) est une méthode qui permet de démarrer un ordinateur à partir du réseau au lieu d'un disque dur ou d'une clé USB. Dans mon cas, j'ai branché les postes via un câble RJ45 connecté à un VLAN spécifique.

Lors du démarrage en PXE, le serveur DHCP attribue automatiquement une adresse IP à la machine, ce qui déclenche ensuite le lancement de l'outil Microsoft Deployment Toolkit (MDT). Cet environnement permet de déployer une image Windows personnalisée comprenant :

- l'intégration automatique dans l'Active Directory de l'entreprise,
- l'installation des logiciels nécessaires aux utilisateurs (chaque logiciel doit être sélectionné manuellement selon ce que l'on veut)
- la configuration des paramètres systèmes adaptés aux besoins internes.

Ainsi, au lieu de réaliser une installation complète et manuelle de Windows poste par poste, le processus devient entièrement automatisé et standardisé. Cela est un gain de temps significatif dans la préparation du matériel de secours.

Vaultwarden

Présentation de Vaultwarden

Vaultwarden est une version gratuite et open-source de Bitwarden, fonctionnant sur Docker. C'est un gestionnaire de mots de passe collaboratif qui permet notamment :

- le partage rapide et sécurisé de mots de passe,
- la génération de mots de passe complexes,
- l'accès aux données depuis plusieurs appareils (application mobile, poste informatique, navigateur ou extension web).

Cette solution est particulièrement adaptée aux entreprises ou équipes qui souhaitent gérer et partager des informations sensibles de manière centralisée et sécurisée.

Présentation de Docker

Docker est un système open-source qui permet de créer et de déployer des applications dans des conteneurs. Un conteneur regroupe le code, les dépendances et la configuration nécessaire pour exécuter une application.

Contrairement aux machines virtuelles, les conteneurs partagent le système hôte, ce qui les rend plus légers et plus rapides. Docker repose sur des images définies dans un fichier appelé Docker Compose, qui décrit les conteneurs et leur configuration. Cela facilite la synchronisation et la gestion de plusieurs conteneurs, tout en simplifiant l'installation d'outils comme Vaultwarden avec une utilisation minimale de ressources.

Pourquoi utiliser Vaultwarden ?

Plusieurs raisons expliquent le choix de Vaultwarden plutôt que Bitwarden ou d'autres solutions :

1 - Open-source et gratuit : contrairement à Bitwarden qui propose certaines fonctionnalités payantes, Vaultwarden est entièrement gratuit et modifiable.

2 - Auto-hébergement : grâce à Docker, Vaultwarden peut être hébergé sur les serveurs de l'entreprise, permettant un contrôle total des données et une meilleure confidentialité.

3 - Personnalisation et extensibilité : étant open-source, le logiciel peut être adapté et modifié selon les besoins. De plus, des conteneurs supplémentaires peuvent être ajoutés pour monitorer ou sécuriser Vaultwarden.

L’objectif

Le but de l'installation de cet solution est de remplacer le gestionnaire de mot de passe actuellement utiliser du nom de Keepass qui ne propose pas de pouvoir partager des mots de passe facilement, d’avoir une double authentification et d’y accéder depuis plus support comme on peut le voir sur ce tableau ci-dessous qui retranscrit les différences entre Vaultwarden et Keepass.

Critère	VAULTWARDEN	KEEPASS
Type de solution	Serveur auto-hébergé, Multi-utilisateur	Application local
Interface utilisateur	Interface web moderne + clients Bitwarden	Interface local Windows
Accès distant	Oui (web, mobile, extension navigateur, application de bureau)	Non (sauf synchronisation manuelle ou plugins)
Partage entre utilisateurs	Oui	Non
Déploiement en entreprise	Facilité via Docker	Limité souvent utilisé individuellement
Mode hors ligne	Non (nécessite un accès serveur)	Oui (totalement autonome)
Complexité de déploiement	Facilités via Docker	Facile (Exécutable autonome)
Utilisation recommandée	Environnement collaboratifs	Utilisation personnelle

Monitoring & Supervision

Lorsqu’on installe une solution auto-hébergée comme Vaultwarden, il est essentiel de disposer d’un espace permettant de surveiller et d’analyser facilement le fonctionnement du système. Cela permet de s’assurer que le gestionnaire de mots de passe fonctionne correctement et d’obtenir des informations sur la machine et les conteneurs associés.

Dans le cadre de la préparation de l'installation de Vaultwarden, j'ai mis en place un espace de monitoring composé de plusieurs conteneurs :

- Grafana : pour la visualisation et l'analyse des données de performance,
- Prometheus : pour la collecte et le stockage des métriques,
- Node Exporter : pour obtenir des informations sur l'état du système hôte,
- VWmetrics : pour des statistiques spécifiques à Vaultwarden.

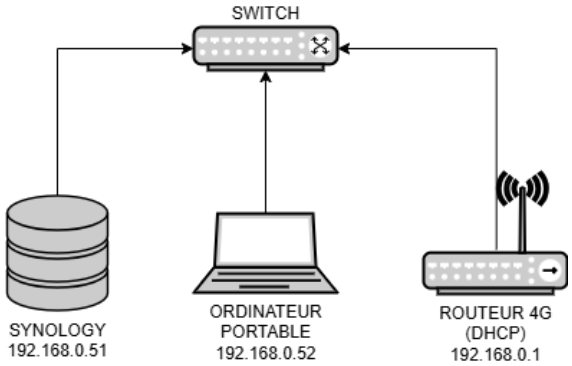
Il est également possible de notifier Portainer, qui permet de superviser l'ensemble des conteneurs via une interface graphique intuitive, facilitant la gestion et le suivi quotidien.

Installation d'essai

Pour l'installation d'essai et de pré configuration de Vaultwarden, j'ai utilisé le matériel suivant:

- NAS Synology D223j : qui peut accueillir Docker
- Ordinateur Portable : pour pouvoir accéder au NAS
- Routeur 4G : Pour pouvoir avec internet sur le NAS et installer les images Docker
- Switch : Pour pouvoir faire communiquer les différents éléments

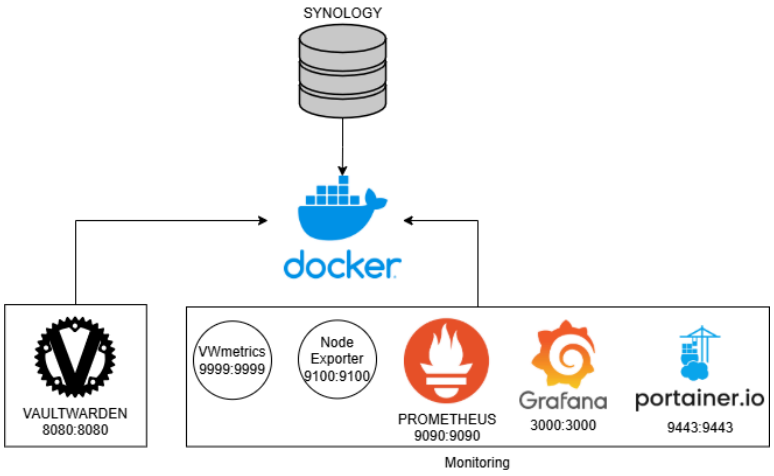
Voici une photo et un schéma de l'installation :



Une fois connecter au synology j'ai pu faire l'installation de Docker via le "centre de paquets" une fois Docker installer j'ai pu depuis le synology installer les différentes images :

- Vaultwarden : Gestionnaire de mot de passe
- Grafana : Dashboard du monitoring
- Vwmetrics : permet de récupérer des informations de Vaultwarden nombres d'utilisateurs, nombres de mots de passe…
- Node Exporter : permet de récupérer les informations du système
- Prometheus : permet de convertir les données de Vwmetrics et Node Exporter pour Grafana
- Portainer : Pour gérer les conteneurs

Voici ci-dessous un schéma du système Docker :

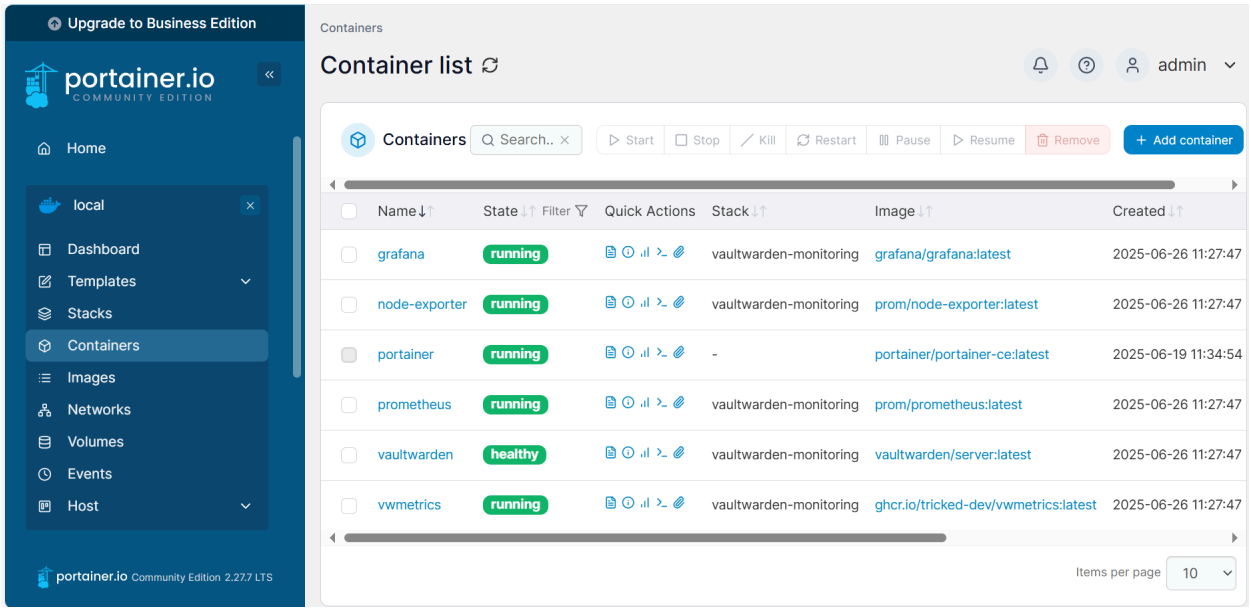


Pour le déploiement des conteneurs j'ai décidé de le faciliter en mettant tous les conteneurs dans le même docker-compose :

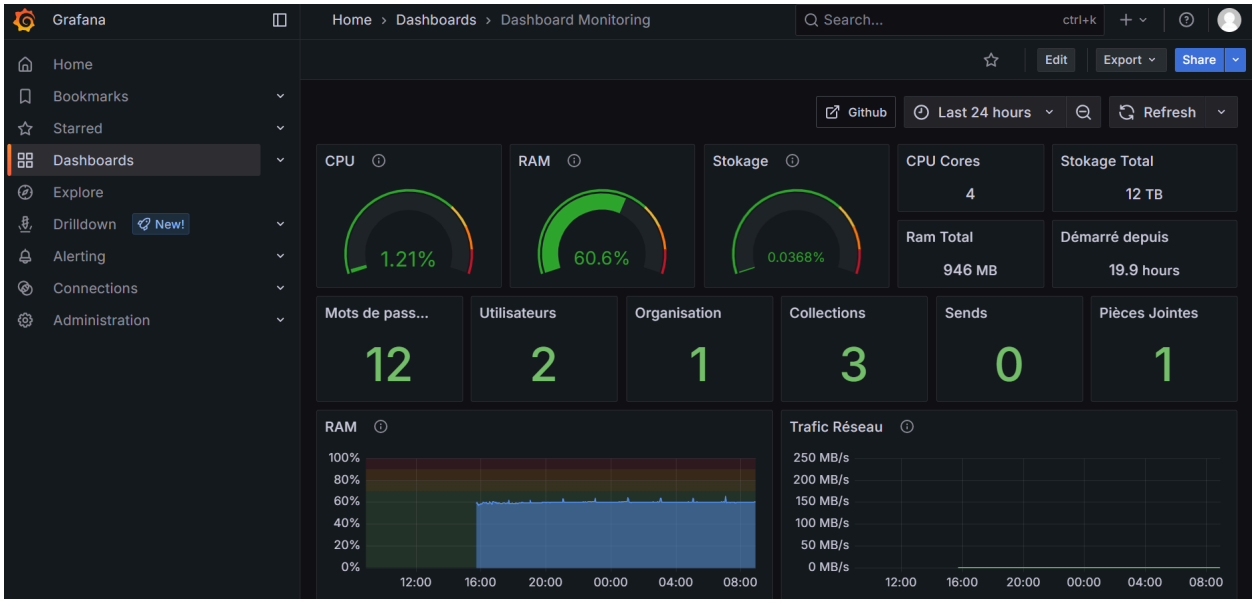
```
version: '3.8'  # Version de la syntaxe Docker Compose utilisée

services:
    vaultwarden:
        image: vaultwarden/server:latest  # Image officielle de Vaultwarden
        container_name: vaultwarden  # Nom du conteneur Docker
        restart: unless-stopped  # Redémarre le conteneur sauf si il est arrêté
manuellement
        ports:
            - "8080:80"      # Redirige le port 80 du conteneur vers le port 8080 de
l'hôte (interface web)
            - "3012:3012"    # Port pour les WebSockets (notifications en temps
réel)
        volumes:
            - ./vw-data:/data  # Monte le dossier local ./vw-data vers /data
        environment:
            - WEBSOCKET_ENABLED=true  # Active le support WebSocket
            - WEBSOCKET_PORT=3012     # Port utilisé pour WebSocket
            # Jeton administrateur sécurisé (exemple avec Argon2id hash)
            - ADMIN_TOKEN=
$$argon2id$$v=19$$m=65540,t=3,p=4$$JlqYxeszEo56RsCZmrllH+tDSJBER7kPVqoOmjcV8LQ$$m1Vm
zyGfLWTUzqGqfYTViwS3CfvA1WrpxqvWj59ySjk
            - SIGNUPS_ALLOWED=false    # Désactiver les inscriptions publiques
            - ADMIN_SESSION_LIFETIME=10000  # Durée de vie de session admin (en
secondes)
            - ORG_CREATION_USERS=admin@admin  # Utilisateurs autorisés à créer des
organisations
            - TRASH_AUTO_DELETE_DAYS=30  # Suppression automatique des éléments dans
la corbeille au bout de 30 jours
            - WEB_VAULT_ENABLED=true  # Active l'interface web de Vaultwarden
            - ORG_GROUPS_ENABLED=true # Active la gestion des groupes d'organisation
            - LOG_TIMESTAMP_FORMAT="%Y- %m- %d %H:%M:%S"  # Format des horodatages
dans les logs
            - ENABLE_DIRECTORY_CONNECTOR=true  # Active la connexion à un annuaire
externe (LDAP, etc.)

    node-exporter:
        image: prom/node-exporter:latest  # Exporter pour metrics système compatible
Prometheus
        container_name: node-exporter
        ports:
             - "9100:9100"  # Port exposé pour Prometheus scraper les metrics
        networks:
             - monitoring  # Connecté au réseau personnalisé "monitoring"

    vwmetrics:
        image: ghcr.io/tricked-dev/vwmetrics:latest  # Service de métriques pour
Vaultwarden
        container_name: vwmetrics
        restart: unless-stopped
        ports:
            - "9999:9999"  # Port exposé pour l'interface métriques
        command:
```

```
            - --database-url
            - sqlite:///data/db.sqlite3?mode=ro  # Base de données en lecture seule
            - --port
            - "9999"
            - --host
            - 0.0.0.0
        volumes:
            - /volume1/docker/volume1/docker/vaultwarden/vw-data:/data:ro  # Montre
le dossier en lecture seule
        networks:
            - monitoring

    prometheus:
        image: prom/prometheus:latest  # Serveur Prometheus pour collecter et
stocker les métriques
        container_name: prometheus
        ports:
            - "9090:9090"  # Port web de Prometheus
        volumes:
            - ./prometheus.yml:/etc/prometheus/prometheus.yml  # Configuration
Prometheus montée depuis local
        networks:
            - monitoring

    grafana:
        image: grafana/grafana:latest  # Interface web Grafana pour visualiser les
métriques
        container_name: grafana
        ports:
                - "3000:3000"  # Port web Grafana
        environment:
            - GF_SECURITY_ADMIN_PASSWORD=admin  # Mot de passe admin Grafana (à
changer a la première connexion)
        volumes:
                - grafana-storage:/var/lib/grafana  # Volume pour les données
Grafana
        networks:
                - monitoring

# Déclaration des volumes
volumes:
    grafana-storage:

# Déclaration des réseaux personnalisés utilisés par certains services
networks:
  monitoring:
    name: monitoring
```

Une fois l'installation et la stack lancer, j'ai vérifier de que tous les conteneurs était bien lancée grâce à portainer :



Une fois les tests effectués sur Vaultwarden que tout fonctionne normalement. Je me suis attaqué à la création du dashboard sur Grafana pour pouvoir afficher les informations relevé :



Annuaire Active Directory

Après avoir installé Vaultwarden et configuré le système de monitoring, il restait à exploiter les informations de l'Active Directory (emails, noms, etc.) afin de créer des comptes automatiquement.

Le problème était que le réseau sur lequel Vaultwarden était installé est isolé du réseau principal, ce qui empêchait l'accès à l'Active Directory de l'entreprise. Pour tester la création de comptes à partir de l'annuaire, j'ai utilisé un serveur isolé avec un hyperviseur (VMware ESXi 7)

Mise en service du serveur ESXI

Le serveur utilisé est un Dell R730, déjà présent dans une baie mais non exploité. J'ai commencé par :

- Ajouter une carte réseau Gigabit pour améliorer les débits,
- Mettre à jour le BIOS et l'iDrac, dont les dernières mises à jour dataient de 2021,
- Installer VMware ESXi 7 pour gérer les machines virtuelles.

Création de la machine virtuel Debian (Docker)

Pour l'OS, j'ai choisi Debian 12, car il supporte plusieurs architectures (amd64, aarch64, ppc64le…) et est léger en ressources.

Après l'installation de Debian, j'ai installé Docker ainsi que les conteneurs nécessaires. Cependant, Vaultwarden ne fonctionnait pas correctement car il n'était pas en HTTPS. J'ai donc mis en place un reverse proxy avec Apache, permettant à Vaultwarden de fonctionner de manière sécurisée.

Création de la machine virtuel Windows Server

Pour Windows Server, j'ai choisi la version 2025, la plus récente, car l'entreprise prévoit de migrer vers cette version. Cela m'a permis de tester la synchronisation avec Vaultwarden sur un environnement proche de la future infrastructure.

Sur cette machine, j'ai créé un Active Directory de test avec :

- un groupe administrateur et un compte admin,
- un groupe utilisateur avec deux comptes utilisateurs.

Synchronisation des utilisateurs entre l'Active Directory et Vaultwarden

Pour pouvoir effectuer la synchronisation il me fallait utiliser Bitwarden Directory Connector une fois dessus et m'être connecté a l'organisation que j'avais créer sur vaultwarden je me suis retrouvé bloqué. En fait dans la documentation de Vaultwarden il est indiqué que la synchronisation avec l'active directory marche grâce au logiciel Bitwarden Directory connector prévu à cet effet comme indiqué ci dessous :

Features

A nearly complete implementation of the Bitwarden Client API is provided, including:

- Personal Vault
- Send
- Attachments
- Website icons
- Personal API Key
- Organizations
  - Collections, Password Sharing, Member Roles, Groups, Event Logs, Admin Password Reset, Directory Connector, Policies
- Multi/Two Factor Authentication
  - Authenticator, Email, FIDO2 WebAuthn, YubiKey, Duo
- Emergency Access
- Vaultwarden Admin Backend
- Modified Web Vault client (Bundled within our containers)

Après avoir passé un certain temps à identifier le problème et en me renseignant directement sur le Git de Vaultwarden, j'ai appris que l'utilisation de Bitwarden Directory Connector pouvait poser des difficultés avec Vaultwarden.

Il existe cependant un conteneur Docker spécifique, considéré comme une extension de Vaultwarden, qui permet de réaliser la synchronisation avec l'Active Directory. Malheureusement, il ne me restait que deux jours de stage, ce qui ne m'a pas permis de configurer correctement ce conteneur ni de tester la synchronisation. Le temps restant étant insuffisant pour lire la documentation complète, comprendre le fonctionnement et paramétrer le système, j'ai décidé de documenter l'ensemble de l'installation et de la configuration réalisée.

Documentation Vaultwarden

Compte tenu du temps limité, j'ai préparé une documentation détaillée de tout ce que j'avais mis en place :

- Explications sur chaque dossier et fichier du système,
- Procédures d'installation pas à pas pour que la personne qui prendra la suite puisse comprendre rapidement le fonctionnement du système,
- Liste des problèmes non résolus, notamment la synchronisation de l'Active Directory avec Vaultwarden.
- Cette documentation permet de faciliter la reprise du projet par un autre collaborateur et d'éviter de perdre du temps à analyser le fonctionnement du système ou à retrouver les problèmes rencontrés.

Conclusion

Bilan du stage

Au cours de mon stage au sein du service informatique chez Raynal et Roquelaure, j'ai eu l'opportunité d'approfondir mes connaissances et de développer de nouvelles compétences. Ce stage m'a permis d'enrichir mon expérience pratique et de mieux comprendre le fonctionnement d'une infrastructure informatique en entreprise. J'ai notamment pu travailler sur des environnements utilisant Docker, ce qui m'a permis de renforcer mes compétences dans la gestion et la mise en place de conteneurs. J'ai appris à mieux appréhender l'outil, également à résoudre certaines problématiques rencontrées dans ce contexte.

En parallèle, j'ai eu l'occasion de découvrir et d'observer ce qu'implique une grande installation réseau au sein d'une entreprise. Cela m'a offert une vision plus concrète des enjeux liés à la configuration, à la sécurité et à la maintenance d'une infrastructure de grande envergure, bien différente de ce que l'on peut rencontrer dans un cadre scolaire ou personnel. En somme, ce stage a été une expérience très formatrice, qui m'a permis à la fois de consolider mes acquis, de progresser sur des aspects techniques comme Docker, et de mieux comprendre l'organisation et la gestion d'un service informatique en entreprise.

Annexes

Secure the ADMIN_TOKEN

> [!WARNING]
> This feature is available since 1.28.0+.
>
> Using environment variables is preferred.
> But if you updated settings via the admin interface you need to update the admin token via the same web interface!
> Please **do not** edit the `config.json` manually since that could cause issues if done wrong!
>
> To log into the admin page after securing the token, you instead use the password provided during token creation.

Previously the `ADMIN_TOKEN` could only be in a plain text format.
You can now hash the `ADMIN_TOKEN` using Argon2 by generating a PHC string.
This can be generated by using a built-in `hash` command within Vaultwarden, or use the `argon2` CLI tool.
Within the vaultwarden application we have two presets, one using the Bitwarden defaults, and one using the OWASP recommendations. Some examples on how to generate an Argon2id PHC hash.

Using argon2

You can also use the `argon2` CLI available on most Linux Distros.

```
# Using the Bitwarden defaults
echo -n "MySecretPassword" | argon2 "$(openssl rand -base64 32)" -e -id -k 65540 -
t 3 -p 4
# Output:
$argon2id$v=19$m=65540,t=3,p=4$bXBGMENBZUVzT3VUSFErTzQzK25Jck1BN2Z0amFuWjdSdVlIQVZ
qYzAzYz0$T9m73OdD2mz9+aJKLuOAdbvoARdaKxtOZ+jZcSL9/N0

# Using the OWASP minimum recommended settings
echo -n "MySecretPassword" | argon2 "$(openssl rand -base64 32)" -e -id -k 19456 -
t 2 -p 1
# Output:
$argon2id$v=19$m=19456,t=2,p=1$cXpKdUxHSWhlaUs1QVVsSStkbTRPQVFPSmdpamFCMHdvYjVkWTV
KaDdpYz0$E1UgBKjUCD2Roy0jdHAJvXihugpG+N9WcAaR8P6Qn/8
```

How to prevent variable interpolation in docker-compose.yml

When [[using Docker Compose]] and you configure the `ADMIN_TOKEN` via the `environment` directive you need to escape all five occurrences of the dollar sign `$` in the generated argon2 PHC string using two dollar signs `$$` in order to prevent variable interpolation:

```
  environment:
    ADMIN_TOKEN:
$$argon2id$$v=19$$m=19456,t=2,p=1$$UUZxK1FZMkZoRHFQRlVrTXZvS0E3bHpNQW55c2dBN2NORzd
sa0Nxd1JhND0$$cUoId+JBUsJutlG4rfDZayExfjq4TCt48aBc9qsc3UI
```

How to prevent variable interpolation in `docker-compose.yml`

When [[using Docker Compose]] and you configure the `ADMIN_TOKEN` via the `environment` directive you need to escape all five occurrences of the dollar sign `$` in the generated argon2 PHC string using two dollar signs `$$` in order to prevent variable interpolation:

```
  environment:
    ADMIN_TOKEN:
$$argon2id$$v=19$$m=19456,t=2,p=1$$UUZxK1FZMkZoRHFQRlVrTXZvS0E3bHpNQW55c2dBN2NORzd
sa0Nxd1JhND0$$cUoId+JBUsJutlG4rfDZayExfjq4TCt48aBc9qsc3UI
```

This can be done automatically e.g. using sed by adding `| sed 's#\$#\$\$#g'` to the end of the `argon2` command line above.

Otherwise you'll get warning messages and the variable will not be set correctly:

```
WARNING: The argon2id variable is not set. Defaulting to a blank string.
WARNING: The v variable is not set. Defaulting to a blank string.
WARNING: The m variable is not set. Defaulting to a blank string.
...
```

> [!NOTE] This is not the case when using a `.env` file for `docker-compose.yaml`
> As shown below. In this case just use the single `$` variant.
> The same for using the docker/podman cli using `-e ADMIN_TOKEN`.

```
/docker-data
├── .env
├── docker-compose.yaml
├── vaultwarden/data
```

.env:

Make sure you use single quotes in the `.env` file used by docker-compose.

```
VAULTWARDEN_ADMIN_TOKEN='$argon2id$v=19$m=65540,t=3,p=4$MmeK.....'
```

Page Admin

> [!IMPORTANT] It's heavily recommended to activate HTTPS before enabling this feature, to avoid possible MITM attacks.

This page allows a server administrator to view all the registered users and to delete them. It also allows inviting new users, even when registration is disabled.

To enable the admin page, you need to set an authentication token. This token can be anything, but it's recommended to use a long, randomly generated string of characters, for example running `openssl rand -base64 48`.

Keep this token secret, this is now the password to access the admin area of your server! Which is why you should secure the admin token.

To set the token, use the `ADMIN_TOKEN` variable:

```
docker run -d --name vaultwarden \
  -e ADMIN_TOKEN=some_random_token_as_per_above_explanation \
  -v /vw-data/:/data/ \
  -p 80:80 \
  vaultwarden/server:latest
```

After this, the page will be available in the `/admin` subdirectory.

The first time you save a setting in the admin page, `config.json` will be generated in your `DATA_FOLDER`. Values in this file will take precedence over the corresponding environment variable.

Note that config changes in the admin page do not take effect until you click the `Save` button. For example, if you are testing SMTP settings, and you change the `SMTP Auth mechanism` setting and then click `Send test email` to test the change, this won't work as expected -- since you didn't click `Save`, the `SMTP Auth mechanism` change won't have taken effect.

Note: After changing the `ADMIN_TOKEN`, any admins that are currently logged in will still be able to use their existing login sessions until expiration. The admin session lifetime is configurable, with a default of 20 minutes.

Disabling the admin page

In order to disable the admin page you have to unset the `ADMIN_TOKEN` and restart Vaultwarden.

Note: Removing the environment variable `ADMIN_TOKEN` won't disable the admin page if the value is persisted in the `config.json` file mentioned above. **To disable admin page**, make sure no `ADMIN_TOKEN` environment variable is set, and no `"admin_token"` key exists in `config.json`, if that file exists.

CARNUS Enseignement supérieur RODEZ

ATTESTATION DE STAGE

BTS CIEL

ORGANISME D'ACCUEIL

Nom ou dénomination sociale : Raynal et Roquelaure

Adresse : Avenue Raynal et Roquelaure
12 700 CAPDENAC GARE

Tél. : 05 65 [illegible] 23 60

Certifie que

LE OU LA STAGIAIRE

Nom : TOURNIE Prénom : Luc Sexe : F ☐ M ☒ Né(e) le : 03/06/2006

Adresse : 655 Rue François Fabié, Livinhac le haut

Tél. : 06 12 50 38 11 Email : luc.tournie@gmail.com

Etudiant en (intitulé de la formation de l'enseignement supérieur suivi par le ou la stagiaire) :
BTS CIEL

Au sein de (nom de l'établissement d'enseignement supérieur ou de l'organisme de formation) :
Charles Carnus

A effectué un stage prévu dans le cadre de ses études

Durée du stage : 32 jours

Date de début et de fin du stage : Du 2 Juin 2025 **Au** 18 Juillet 2025

Représentant une durée totale de 7 **nombre de semaines** ~~/ de mois~~ (rayer la mention inutile).

La durée totale du stage est appréciée en tenant compte de la présence effective du stagiaire dans l'organisme, sous réserve des droits à congés et autorisations d'absence prévus à l'article L. 124-13 du code de l'éducation (art. L. 124-18 du code de l'éducation). Chaque période au moins égale à 7 heures de présence consécutives ou non est considérée comme équivalente à un jour de stage et chaque période au moins égale à 22 jours de présence consécutifs ou non est considérée comme équivalente à un mois.

MONTANT DE LA GRATIFICATION VERSEE AU STAGIAIRE

Le stagiaire a perçu une gratification de stage pour **un montant total de** 0 €

L'attestation de stage est indispensable pour pouvoir, sous réserve du versement d'une cotisation, faire prendre en compte le stage dans les droits à retraite. La législation sur les retraites (loi n°2014-40 du 20 janvier 2014) ouvre aux étudiants **dont le stage a été gratifié** la possibilité de faire valider celui-ci dans la **limite de deux trimestres**, sous réserve du **versement d'une cotisation**. **La demande est à faire par l'étudiant dans les deux années** suivant la fin du stage et sur présentation **obligatoire de l'attestation de stage** mentionnant la durée totale du stage et le montant total de la gratification perçue. Les informations précises sur la cotisation à verser et sur la procédure à suivre sont à demander auprès de la Sécurité sociale (code de la Sécurité sociale art. L.351-17 – code de l'éducation art. D. 124-9).

FAIT À Capdenac Gare **LE** 6 à 18 Juillet 2025

Nom, fonction et signature du représentant de l'organisme d'accueil

Thibaut Gallajoli
Responsable Infrastructures et Cybersécurité