BTS Services informatiques aux organisations- SISR

Session 2025

E5 – Support et mise à disposition de services informatiques

Coefficient 4

DESCRIPTION DE LA REALISATION PROFESSIONNELLE

NOM et prénom du candidat :

Nathan VANDENBOSSCHE

Contexte de la réalisation professionnelle

- ***Layer Bureautique et Informatique*** *est une entreprise spécialisée dans la gestion d'infrastructures IT et la virtualisation, offrant des services essentiels pour répondre aux besoins de performance, sécurité et continuité des systèmes informatiques de ses clients.*
- *La problématique principale est de créer un site web qui centralise mes réalisations professionnelles (RP) pour le passage de mon BTS SIO. Ce site inclut mon CV, une présentation de moi-même et un accès protégé à toutes mes RP.*
- *La solution choisie consiste à déployer une infrastructure locale avec plusieurs technologies : pfSense pour le réseau, Proxmox pour la virtualisation, Apache/WordPress pour l'hébergement web, et MariaDB pour la base de données.*

Intitulé de la réalisation professionnelle

Mise en place d'une Infrastructure et d'un site web

Période de réalisation : 04/11/24- 20/01/25 **Lieu :** Fleury-la-vallée

Modalité : ☒ Individuelle ☐ En équipe

Principale(s) activité(s) concernée(s) :

- METTRE A DISPOSITION DES UTILISATEURS UN SERVICE INFORMATIQUE
- DEVELOPPER LA PRESENCE EN LIGNE DE L'ORGANISATION
- GERER LE PATRIMOINE INFORMATIQUE
- ORGANISER SON DEVELOPPEMENT PROFESSIONNEL

Conditions de réalisation

- **Ressources présentes (situation avant la RP)**

  Je suis parti de zéro et ai récupéré du matériel auprès de mon entreprise et de connaissances : des PC transformés en serveurs, des Sophos convertis en pfSense, ainsi que des switchs et autres équipements nécessaires.

- **Résultats attendus (situation après la RP)**

  À la fin de cette RP, j'aurai une infrastructure stable pour héberger un site web accessible, sécurisé, et regroupant mon CV, mes RP et ma présentation, conçu pour une consultation fluide et professionnelle lors de mon BTS SIO.

- **Durée de réalisation**

  Ce projet a commencé tôt dans mon BTS SIO et a duré plusieurs mois, incluant la récupération du matériel et d'installation. Le site continue d'évoluer jusqu'à l'examen et sera amélioré au fil du temps en fonction de mes besoins.

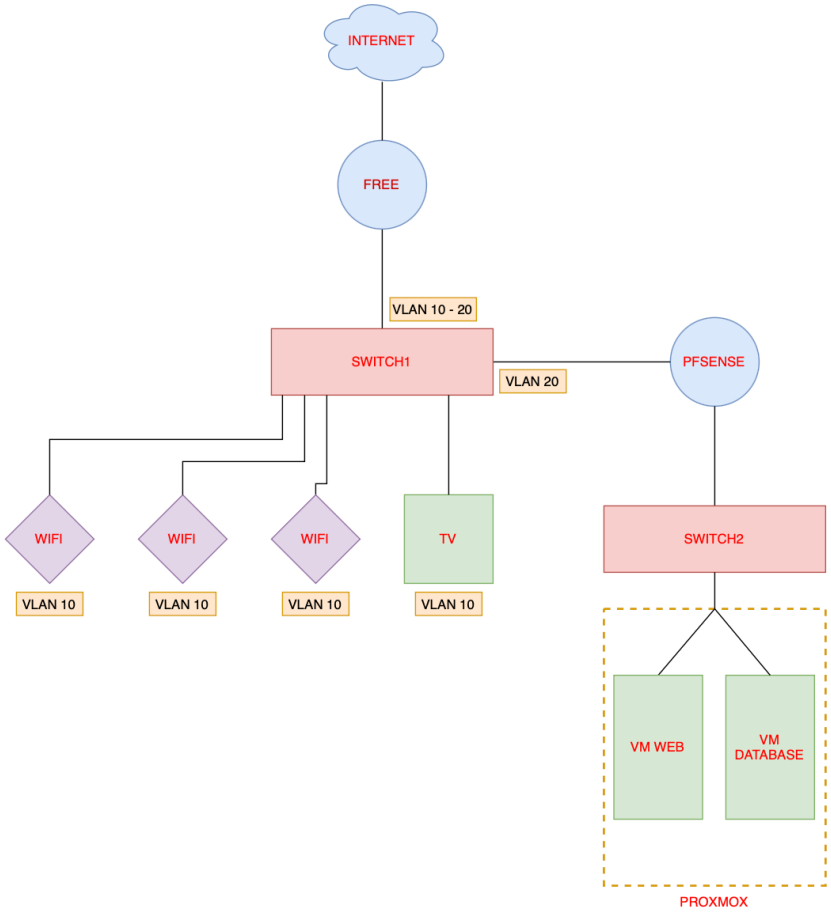
Modalités d'accès à cette réalisation professionnelle.

https://portfolio.vdb-pro.fr *mdp : Cyb3r-M@P89$*

Partie 1 – Procédure de mise en œuvre

Dans le cadre d'un projet personnel et de formation en BTS SIO, j'ai mis en place une **infrastructure réseau locale** pour héberger un **site web** directement à mon domicile. Tous les équipements, tels que le routeur, les serveurs et les switchs, sont installés chez moi, avec l'acquisition d'un **nom de domaine public** et la configuration d'un **DNS public**.

Pour créer cette **infrastructure locale**, j'ai dû récupérer du matériel grâce à ma société et à des connaissances. Cela m'a permis d'obtenir un ancien ordinateur transformé en serveur, un switch, des routeurs Sophos reconfigurés, ainsi que des onduleurs pour sécuriser l'alimentation. J'ai commencé par mettre en place un réseau local en intégrant le switch, ce qui m'a permis de gérer et sécuriser le trafic des paquets de données, tout en établissant une base solide pour l'ensemble de l'infrastructure.

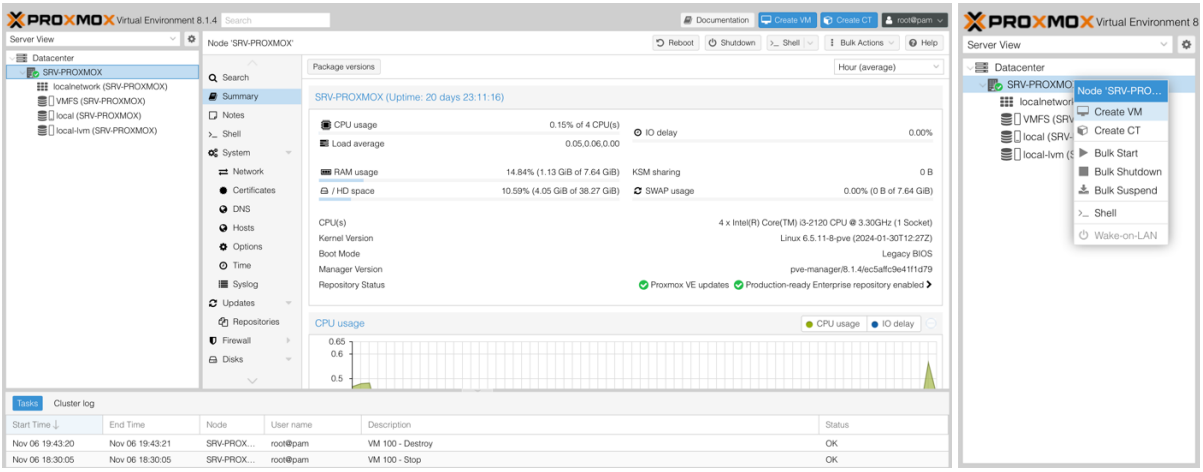


Le serveur Proxmox hébergera deux machines virtuelles sous Linux, qui rempliront des rôles spécifiques dans l'infrastructure, tout en servant de support pour mes sauvegardes. Quant au routeur Sophos, il sera reconfiguré pour devenir un pare-feu performant sous pfSense, renforçant ainsi la sécurité du réseau
Le Proxmox, pfSense, le switch et la box seront directement connectés à un onduleur afin d'assurer une alimentation continue et d'éviter toute coupure de courant.

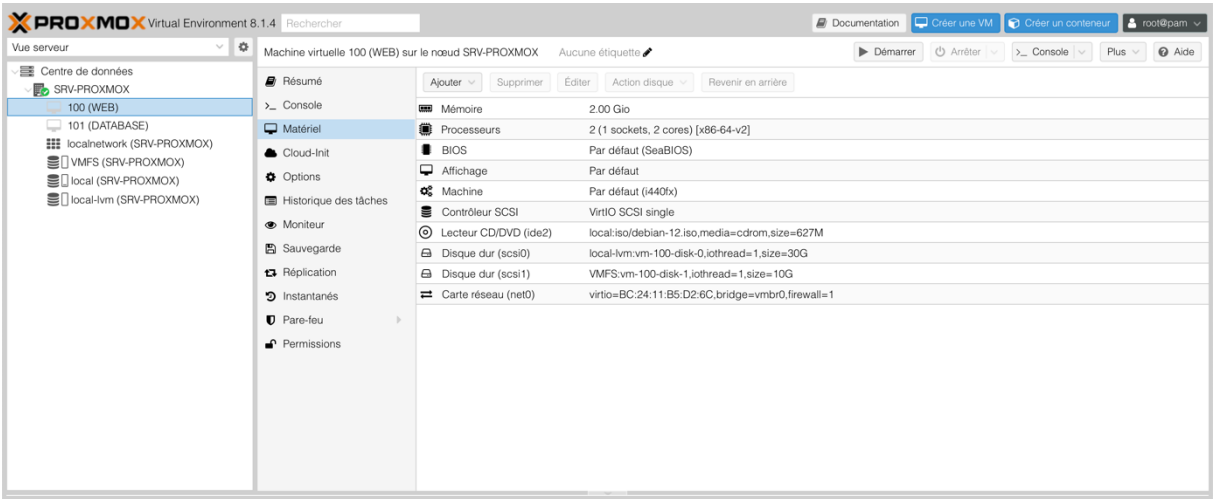
Création des VM dans PROXMOX

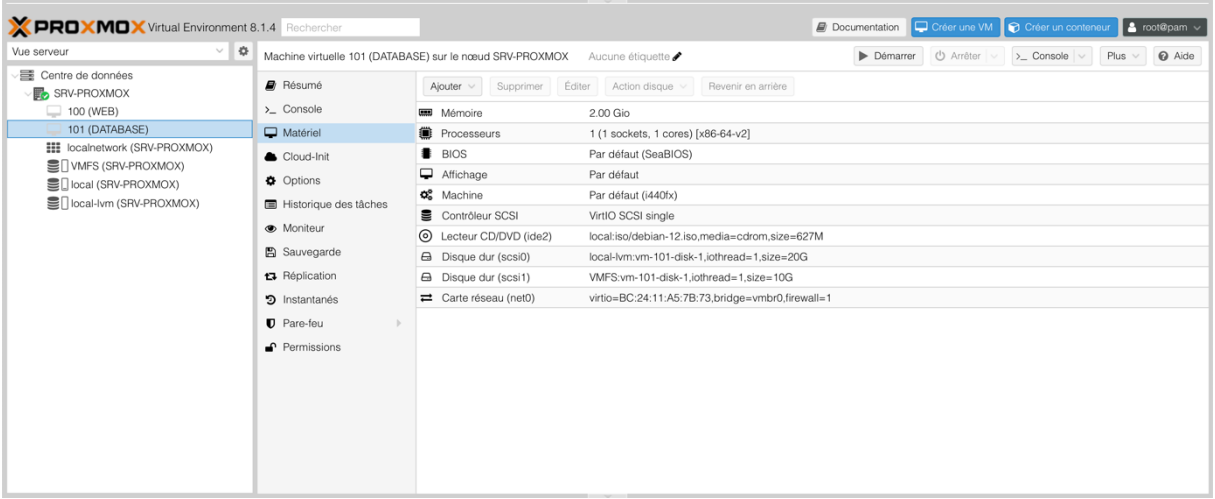
Suivez les étapes du guide d'installation Proxmox - [Installer un serveur Proxmox]

- Connectez-vous à l'interface web de Proxmox avec l'adresse IP et les identifiants administrateurs configurés lors de l'installation.
- Faites un clic droit sur le nom du serveur, par exemple **SRV-PROXMOX**.
- Sélectionnez **Create VM** dans le menu déroulant.



Les étapes détaillées de la création d'une VM sont volontairement omises ici pour privilégier une vue directe sur la configuration matérielle. Retrouvez ci-dessous les captures d'écran présentant les spécifications des VM créées :



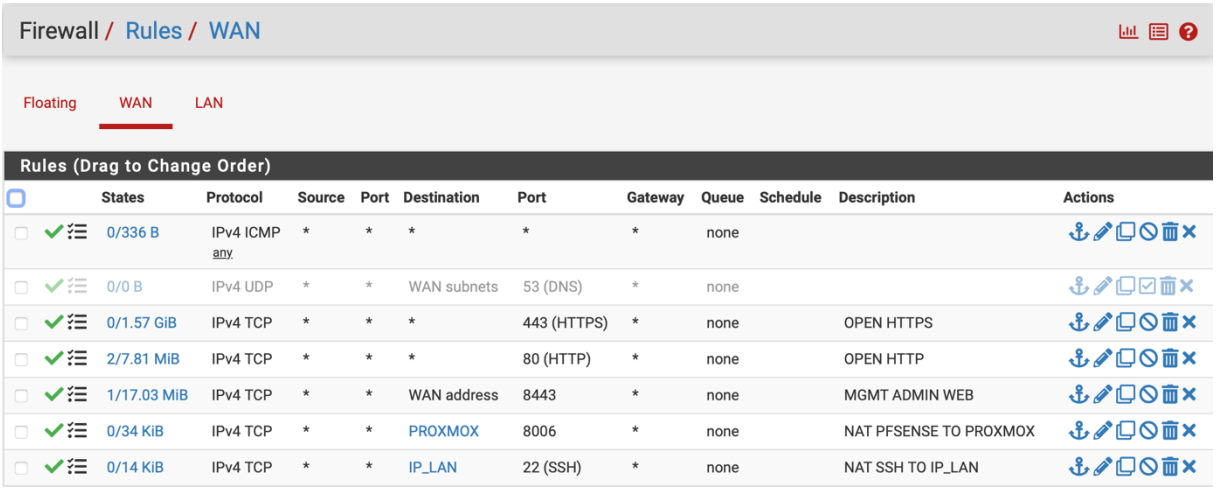


Après la création des VM dans Proxmox, j'ai installé pfSense sur le routeur Sophos, transformant ainsi ce dernier en un pare-feu performant et gratuit.

1. **Installation de pfSense sur le Sophos** :
   Insérez une clé USB pfsense dans le routeur Sophos, puis démarrez-le à partir de cette clé pour lancer l'installation de pfSense. L'installation est similaire à celle d'un système d'exploitation classique.
2. **Configuration réseau après installation :**
   Une fois pfSense installé, la première étape consiste à configurer le réseau. Il faut définir les interfaces réseau (WAN et LAN), configurer les adresses IP, et ajuster les paramètres de sécurité du pare-feu.

/!\ Par mesure de securité, je ne devoilerai pas les adresses IP du projet /!

Règles WAN



- ICMP :
  J'ai ouvert le port ICMP pour permettre les pings et tester les échanges réseau. Ce port sera fermé ultérieurement une fois les tests terminés, afin de renforcer la sécurité.
- Ports 80 (HTTP) et 443 (HTTPS) :
  Les ports **80 (HTTP)** et **443 (HTTPS)** sont laissés ouverts pour le bon fonctionnement du site web, avec une redirection automatique des requêtes HTTP vers HTTPS pour sécuriser les connexions.
- Port 8443 :
  Ce port est spécifiquement configuré pour permettre la gestion de pfSense via son interface web (mgmt).
- Règles NAT :
  Les ports nécessaires pour le NAT (Network Address Translation) ont été configurés pour que je puisse accéder à certaines ressources de mon réseau local depuis mon infrastructure. Cela facilite la gestion et l'accès distant tout en maintenant un contrôle précis des règles de sécurité.

Règles LAN

Firewall / Rules / LAN

Floating | WAN | LAN

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	0/517 KiB	*	*	*	LAN Address	8443 80 22	*	*		Anti-Lockout Rule	⚙
✓	0/0 B	IPv4 ICMP any	*	*	*	*	*	none		PING TO EXT	⚓ ✎ ⧉ ⊘ 🗑 ✕
✓	1/31.34 MiB	IPv4 UDP	LAN subnets	*	*	53 (DNS)	*	none			⚓ ✎ ⧉ ⊘ 🗑 ✕
✓	1/3.06 MiB	IPv4 UDP	LAN subnets	*	*	123 (NTP)	*	none			⚓ ✎ ⧉ ⊘ 🗑 ✕
✓	2/13.38 MiB	IPv4 TCP	LAN subnets	*	*	80 (HTTP)	*	none			⚓ ✎ ⧉ ⊘ 🗑 ✕
✓	0/222.68 MiB	IPv4 TCP	LAN subnets	*	*	443 (HTTPS)	*	none			⚓ ✎ ⧉ ⊘ 🗑 ✕

- ICMP :
  Autorisé pour permettre les diagnostics réseau et tester la connectivité entre les appareils.
- Port 53 (DNS) :
  Permet la résolution des noms de domaine en adresses IP pour les appareils du réseau local.
- Port 123 (NTP) :
  Utilisé pour la synchronisation des horloges des appareils avec des serveurs de temps.
- Ports 80 (HTTP) et 443 (HTTPS) :
  Autorisés pour permettre la navigation web sécurisée depuis le réseau local.

NAT Port Forward

Firewall / NAT / Port Forward

Port Forward | 1:1 | Outbound | NPt

Rules

	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
✓	WAN	TCP	*	*	WAN address	22001	VM_DATABASE	22 (SSH)	SSH TO DATABASE	✎ ⧉ 🗑
✓	WAN	TCP	*	*	WAN address	22002	VM_WEB	22 (SSH)	SSH TO WEB	✎ ⧉ 🗑
✓	WAN	TCP	*	*	WAN address	22010	PROXMOX	22 (SSH)	SSH TO PROXMOX	✎ ⧉ 🗑
✓	WAN	TCP	*	*	WAN address	8006	PROXMOX	8006		✎ ⧉ 🗑

- Redirection du port 22 (SSH) :
  J'ai configuré une redirection NAT pour le port **22** (SSH) vers des ports externes spécifiques pour chaque appareil afin de sécuriser et différencier les accès :
  - Le port **22001** redirige vers l'IP de la VM DATABASE.
  - Le port **22002** redirige vers l'IP de la VM WEB, et ainsi de suite.
- Port 8006 (Proxmox) :
  - Le port **8006** est directement redirigé vers **8006**, car il s'agit du port de gestion par défaut de Proxmox.

Pour configurer des machines virtuelles (VM) pour un serveur web hébergeant des applications comme **Apache2**, **WordPress**, **MariaDB** et **PHP**, l'allocation des ressources doit être bien équilibrée pour garantir des performances optimales. Vérifions ensemble quelques recommandations sur l'allocation des ressources.

1. Évaluer les besoins

Avant de dimensionner, réfléchissons aux points suivants :

- **Nombre d'utilisateurs simultanés :** un site à faible trafic n'aura pas les mêmes exigences qu'un site fortement fréquenté.
- **Complexité des pages :** un site avec beaucoup de contenu multimédia (images, vidéos) ou des plugins WordPress lourds peut nécessiter plus de mémoire et de CPU.
- **Type de base de données :** les requêtes SQL complexes et fréquentes demandent plus de mémoire et de CPU.

2. Répartition des services

Séparer les services peut optimiser les performances, mais cela dépend des ressources disponibles et du besoin de scalabilité :

- **VM Web :** pour héberger Apache2 et PHP.
- **VM Base de données :** pour héberger MariaDB (si le trafic est important ou pour des besoins de sécurité).
- **Sauvegarde externalisée :** optionnelle, pour effectuer des backups.

3. Configuration des ressources par VM

Service	vCPU	RAM	Stockage (SSD)	Stockage (HDD)	Détails
VM Web	2	2 Go	30 Go + 2 Go (SWAP)	10 Go (LIBRE)	Apache2, PHP, WordPress
VM Base de données	1	2 Go	20 Go + 1 Go (SWAP)	10 Go (LIBRE)	MariaDB
Sauvegardes			50 Go	100 Go	Sauvegardes régulières

5. Optimisations supplémentaires

Pour améliorer la performance, envisagez :

- **Mise en cache :** utilisez un plugin de cache pour WordPress (par ex. W3 Total Cache) pour réduire les temps de chargement.
- **CDN (Content Delivery Network) :** pour alléger la charge des serveurs et accélérer la distribution du contenu.
- **Optimisation de la base de données :** configuration de query_cache_size, indexation des tables, et analyse des requêtes lentes.
- **Sécurisation :** mettre en place des règles de pare-feu strict, surveiller les accès, et configurer des sauvegardes automatiques.

Étapes suggérées

1. Installer la VM Database (MariaDB) :

- Installez MariaDB, créez la base de données pour WordPress, configurez les utilisateurs et définissez les permissions nécessaires.
- Testez localement la connexion à MariaDB pour confirmer que tout fonctionne correctement.

2. Configurer la VM Web (Apache, PHP, WordPress) :

- Installez Apache, PHP et WordPress.
- Connectez WordPress à la base de données MariaDB en utilisant les informations de connexion définies précédemment.
- Une fois que la connexion est confirmée, vous pourrez lancer WordPress et vérifier que le site fonctionne comme attendu.

Mise en place de la VM Database

Pour préparer un environnement WordPress stable, je commence par configurer la VM dédiée à la base de données. Cela permet d'assurer une structure solide dès le départ.

Étape 1 : Installer MariaDB

Installation de la VM sous Linux Debian 12 :

La VM est installée sans interface graphique pour réduire la consommation de ressources et privilégier les performances.

Installation de MariaDB :

Une fois la VM démarrée, voici les commandes exécutées pour installer et configurer

Configuration de MariaDB :

Mise à jour des paquets

```
sudo apt update && sudo apt upgrade -y
```

Installation de MariaDB

```
sudo apt install mariadb-server -y
```

Étape 2 : Sécuriser l'installation de MariaDB

Sécurisation de l'installation MariaDB

```
sudo mysql_secure_installation
```

Lors de l'exécution de ***mysql_secure_installation***, j'ai configuré un mot de passe pour l'utilisateur root, supprimé les utilisateurs anonymes, désactivé les connexions root distantes et supprimé les bases de test.

```
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
 ... Success!


You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
 ... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@Database:~#
```

Étape 3 : Créer une base de données et un utilisateur pour WordPress

Création de la base de données pour WordPress :

Une fois MariaDB installé, j'ai créé une base de données et un utilisateur dédié :

Connexion à MariaDB

```
sudo mysql -u root -p
```

Commandes MariaDB

```
CREATE DATABASE wordpress_db;
CREATE USER 'wordpress_user'@'%' IDENTIFIED BY
'mdp_securise';
GRANT ALL PRIVILEGES ON wordpress_db.* TO
'wordpress_user'@'ip_vm_web';
FLUSH PRIVILEGES;
EXIT;
```

```
root@Database:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 44
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE wordpress_db CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> CREATE USER 'wp_nathan'@'localhost' IDENTIFIED BY '[redacted]';
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress_db.* TO 'wp_nathan'@'localhost';
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]> EXIT;
Bye
root@Database:~#
```

Ces commandes permettent de créer une base de données nommée **wordpress_db**, un utilisateur dédié **wordpress_user** avec un mot de passe sécurisé, et de lui accorder les permissions nécessaires.

Configuration de MariaDB pour les connexions distantes :

J'ai modifié le fichier de configuration pour permettre les connexions depuis la VM WordPress :

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

J'ai remplacé la ligne suivante :

```
bind-address = 127.0.0.1
```

Par :

```
bind-address = <ip_vm_web>
```

Puis, redémarré MariaDB pour appliquer les modifications :

```
sudo systemctl restart mariadb
```

Vérifier que MariaDB fonctionne bien :

```
sudo systemctl status mariadb
```

Vérifier la connexion de WordPress à la base de données en accédant à l'installation de WordPress via un navigateur. Si tout est configuré correctement, WordPress pourra se connecter à la base de données et démarrer la procédure d'installation.

Conclusion :

Cette configuration garantit une base de données MariaDB sécurisée, prête à être utilisée par WordPress. L'étape suivante consistera à configurer la VM hébergeant WordPress pour qu'elle se connecte à cette base de données.

Installation de WordPress : Étapes détaillées

Mise à jour du système :

Assurez-vous que le système est à jour :

```
sudo apt update && sudo apt upgrade -y
```

Installation d'Apache2 :

Installez le serveur web **Apache2** :

```
sudo apt install apache2 -y
```

Activez **Apache2** pour qu'il démarre automatiquement :

```
sudo systemctl enable apache2
sudo systemctl start apache2
```

Installation de PHP et des extensions nécessaires :

WordPress nécessite PHP et certaines extensions spécifiques :

```
sudo apt install php php-mysql php-xml php-mbstring php-curl
php-zip php-gd libapache2-mod-php -y
```

Vérifiez que **PHP** est correctement installé :

```
php -v
```

Téléchargement de WordPress :

Téléchargez la dernière version de WordPress depuis le site officiel :

```
cd /tmp
wget -O https://wordpress.org/latest.tar.gz
```

Décompressez l'archive téléchargée :

```
tar xzvf latest.tar.gz
```

Déplacez le dossier **WordPress** vers le répertoire web d'Apache :

```
sudo mv wordpress /var/www/html/wordpress
```

Configuration des permissions :

Pour qu'Apache puisse lire et écrire dans le dossier WordPress, attribuez les bonnes permissions :

```
sudo chown -R www-data:www-data /var/www/html/wordpress
sudo find /var/www/html/wordpress -type d -exec chmod 755 {} \;
sudo find /var/www/html/wordpress -type f -exec chmod 644 {} \;
```

Configuration de WordPress :

Copiez le fichier de configuration par défaut de WordPress et éditez-le :

```
sudo cp /var/www/html/wordpress/wp-config-sample.php /var/www/html/wordpress/wp-config.php
sudo nano /var/www/html/wordpress/wp-config.php
```

Modifiez les lignes suivantes :

```
define('DB_NAME', 'wordpress_db');
define('DB_USER', 'wp_nathan');
define('DB_PASSWORD', 'MDP_SECURISÉ');
define('DB_HOST', 'IP_DE_VOTRE_DATABASE');
define('DB_CHARSET', 'utf8mb4');
define('DB_COLLATE', '');
```

Activation du module rewrite pour Apache2 :

WordPress nécessite le module **rewrite** d'Apache pour ses permaliens :

```
sudo a2enmod rewrite
sudo systemctl restart apache2
```

Configuration du VirtualHost Apache :

Créez un fichier de configuration Apache pour WordPress :

```
sudo nano /etc/apache2/sites-available/portfolio.conf
```

Ajoutez le contenu suivant :

```
<VirtualHost *:80>
    ServerName portfolio.vdb-pro.fr
    ServerAlias www.portfolio.vdb-pro.fr
    ServerAdmin nathan.vandenbossche@vdb-pro.fr
    DocumentRoot /var/www/html/wordpress
    <Directory /var/www/html/wordpress>
        AllowOverride All
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
 GNU nano 7.2        /etc/apache2/sites-available/portfolio.conf
<VirtualHost *:80>
    ServerName portfolio.vdb-pro.fr
    ServerAlias www.portfolio.vdb-pro.fr
    DocumentRoot /var/www/html/wordpress
    <Directory /var/www/html/wordpress>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Activez ce site et rechargez Apache :

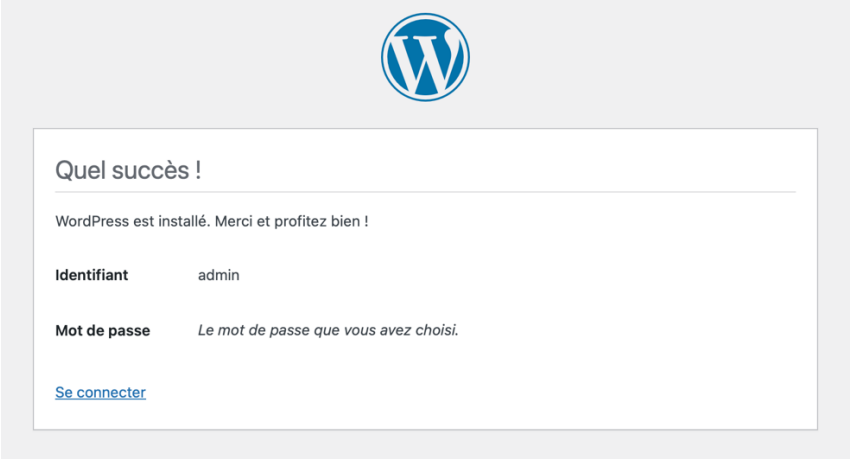
```
sudo a2ensite portfolio
sudo systemctl reload apache2
```

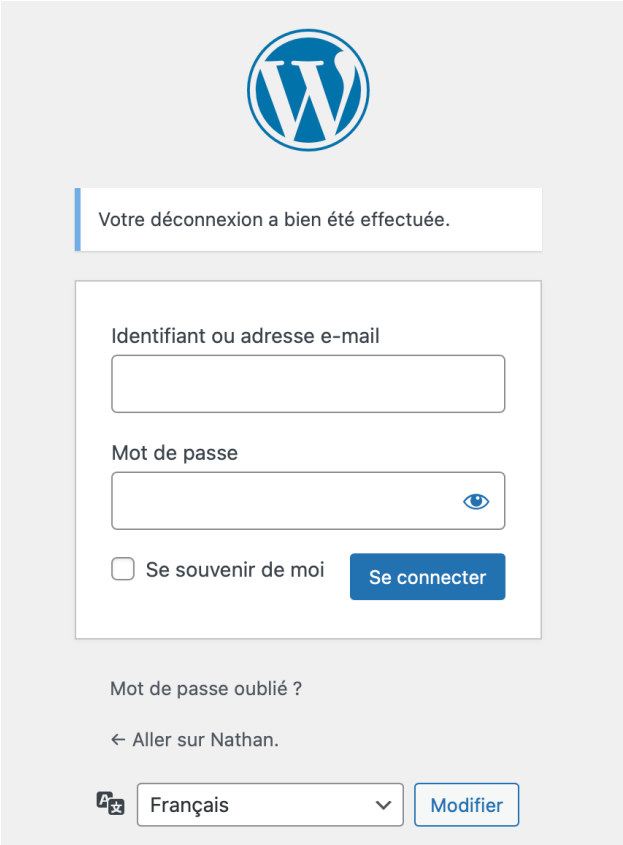
Terminer l'installation :

Accédez à votre site WordPress depuis votre navigateur pour finaliser la configuration :

```
http://votre_ip/wordpress
```

Vous serez invité à configurer le nom de votre site, le compte administrateur, et les paramètres de base directement depuis l'interface WordPress.



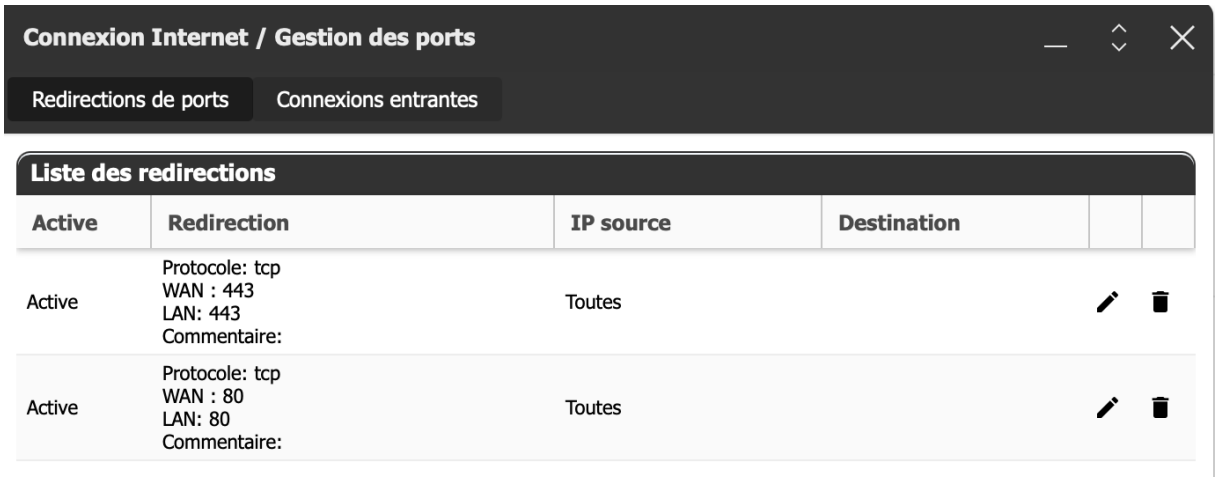


Avant la configuration du reverse proxy

1. Achat d'un nom de domaine public :
   - J'ai acheté le nom de domaine **portfolio.vdb-pro.fr** chez **OVH**.
2. Redirection du domaine vers l'IP publique :
   - J'ai configuré une redirection dans l'interface de gestion OVH pour que **portfolio.vdb-pro.fr** pointe vers l'**IP publique** de ma box Internet.

Domaine	TTL	Type	Cible
vdb-pro.fr.	0	MX	100 mx3.mail.ovh.net.
www.portfolio.vdb-pro.fr.	0	A	82. .162
portfolio.vdb-pro.fr.	0	A	82. .162

3. Configuration de la redirection des ports sur la box Free :
   - J'ai configuré ma box Free pour rediriger les **ports 443 (HTTPS)** et **80 (HTTP)** vers l'**IP WAN** de mon deuxième routeur, qui gère le réseau local.
   - Cette redirection permet d'acheminer les requêtes HTTP(S) depuis l'extérieur jusqu'à mon infrastructure locale.



Avec cette configuration de base, mon nom de **domaine public** est fonctionnel et redirige correctement les connexions externes vers mon réseau local, où j'ai ensuite mis en place le reverse proxy avec **HAProxy** pour gérer les requêtes.

Configuration de HAProxy sur pfSense en tant que reverse proxy

Pour rediriger l'accès **https://[nom_de_domaine]** vers ma **machine virtuelle WEB**, l'utilisation de **HAProxy** sur pfSense est une excellente solution. Voici les étapes détaillées pour mettre en place cette configuration :

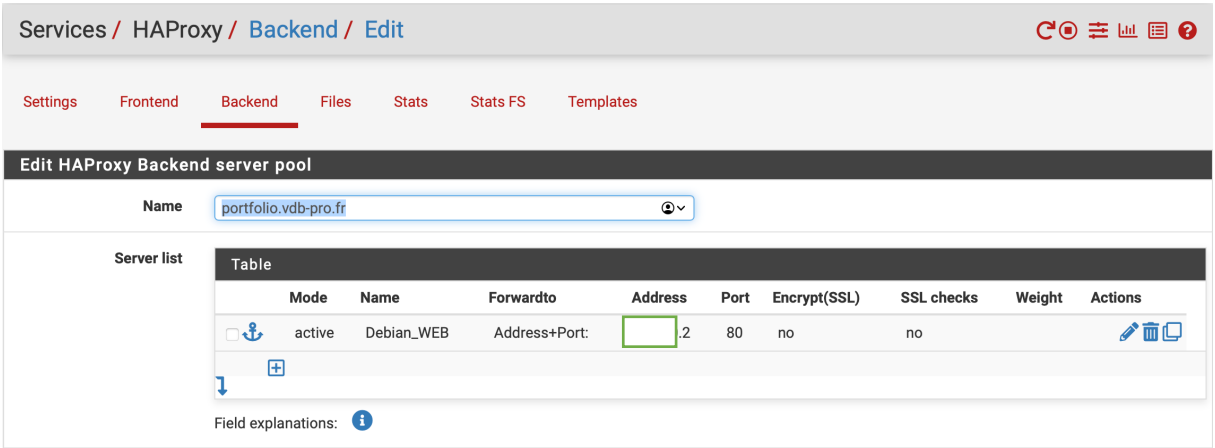
Installer HAProxy sur pfSense :

1. Accédez à l'interface Web de pfSense.
2. Allez dans **System > Package Manager > Available Packages**.
3. Recherchez **HAProxy** dans la barre de recherche.
4. Cliquez sur **Install** pour installer le package.

Configurer HAProxy pour la redirection :

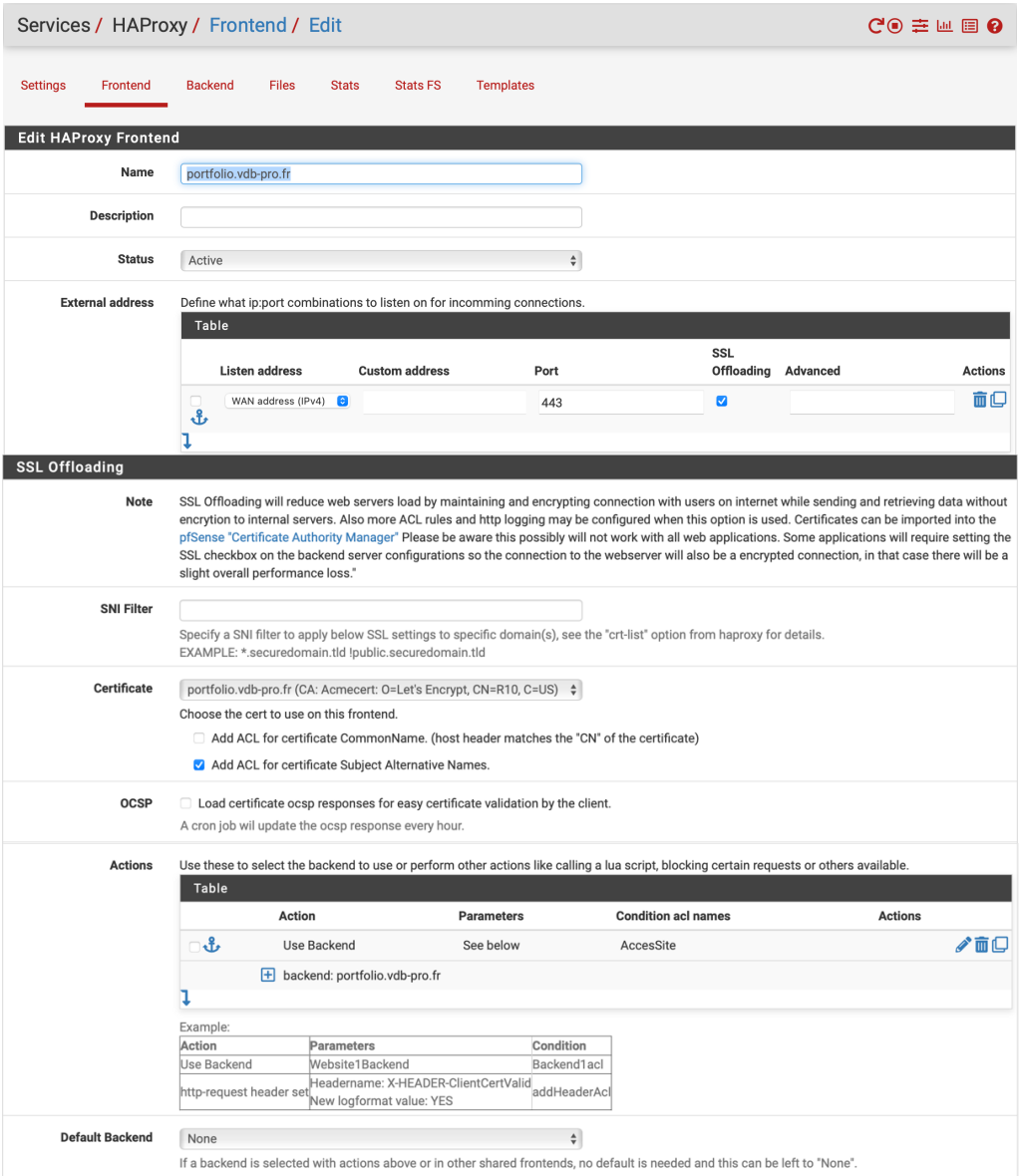
a. Ajouter une Backend (serveur cible)

1. Accédez à **Services > HAProxy > Backend**.
2. Cliquez sur **Add** pour créer un nouveau backend.
3. Configurez les options suivantes :
    - **Name** : Donnez un nom à votre backend, par exemple, wordpress_backend.
    - **Server list** : Ajoutez votre serveur web hébergeant WordPress :
        - **Name** : wordpress_server.
        - **Address** : L'IP locale de votre serveur WordPress (par exemple, 192.168.1.10).
        - **Port** : Le port sur lequel votre serveur web écoute (généralement 443 pour HTTPS) mais pour ma part ce sera le 80 (HTTP).
    - Enregistrez la configuration.



Ajouter une Frontend (point d'entrée utilisateur) :

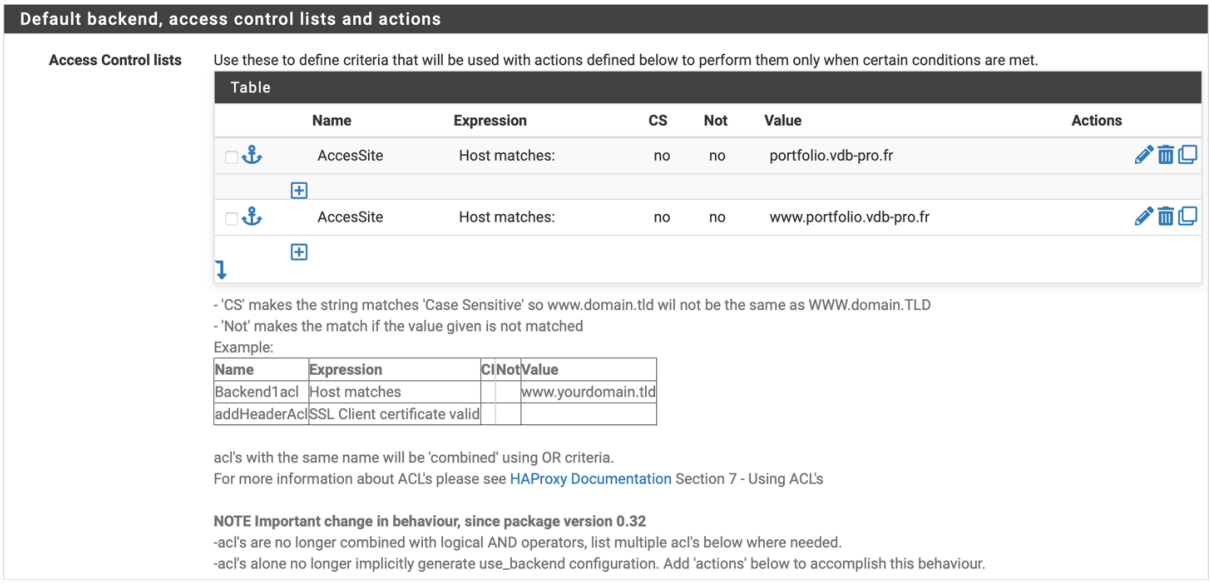
1. Accédez à **Services > HAProxy > Frontend**.
2. Cliquez sur **Add** pour ajouter une nouvelle frontend.
3. Configurez les options suivantes :
    - **Name** : Donnez un nom à votre frontend, par exemple, wordpress_frontend.
    - **External Address** : Sélectionnez votre WAN (adresse publique) ou une adresse IP spécifique si votre pfSense a plusieurs interfaces.
    - **Port** : Spécifiez le port d'écoute, généralement 443 pour HTTPS.
    - **SSL Offloading** : Activez cette option si vous voulez qu'HAProxy gère le chiffrement HTTPS.
    - **Default Backend** : Choisissez le backend précédemment créé (wordpress_backend).
4. Enregistrez la configuration.



Configurer la redirection de l'IP vers le nom de domaine :

Toujours dans la configuration de la **Frontend**, ajoutez une règle dans la section **Actions** pour effectuer la redirection :

1. **Condition ACL** : Ajoutez une condition pour détecter les connexions vers l'IP publique :
    - **Name :** AccesSite
    - **Expression :** Host matches:
    - **Value :** [votre nom de domaine public]



Ajouter un Frontend HTTP pour la Redirection :

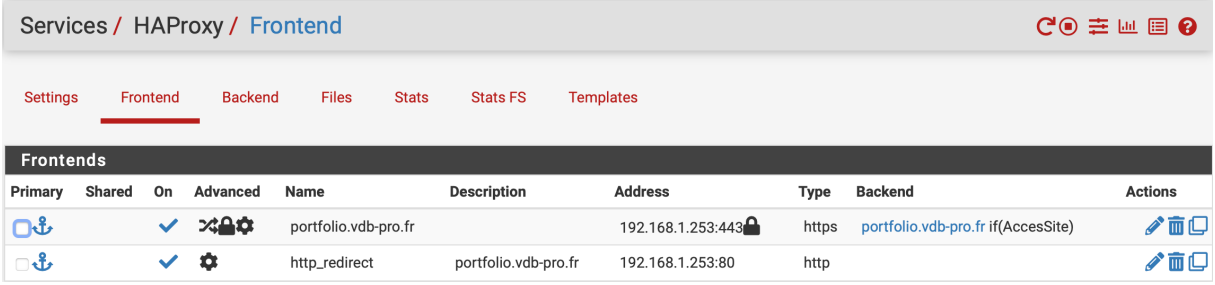
Créez un **frontend** dédié uniquement à rediriger HTTP vers HTTPS.

Dans **pfSense > Services > HAProxy > Frontend**, ajoutez un nouveau frontend :

- **Name** : http_redirect
- **Listen Address** : WAN Address:80 (écoute sur le port 80).
- **Action** :
    - Sélectionne **Advanced pass-through** ou ajoute une règle :

```
http-request redirect scheme https code 301 if !{ ssl_fc }
```

Cela redirige toutes les requêtes HTTP reçues sur le port 80 vers HTTPS.

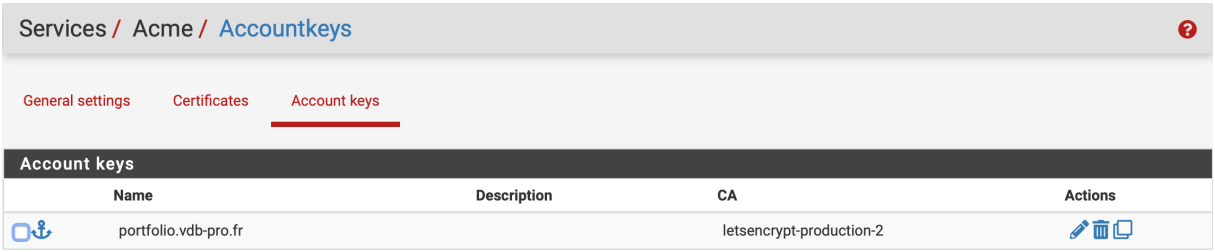


Générer un certificat SSL Let's Encrypt et le configurer pour HAProxy

Génération des clés ACME

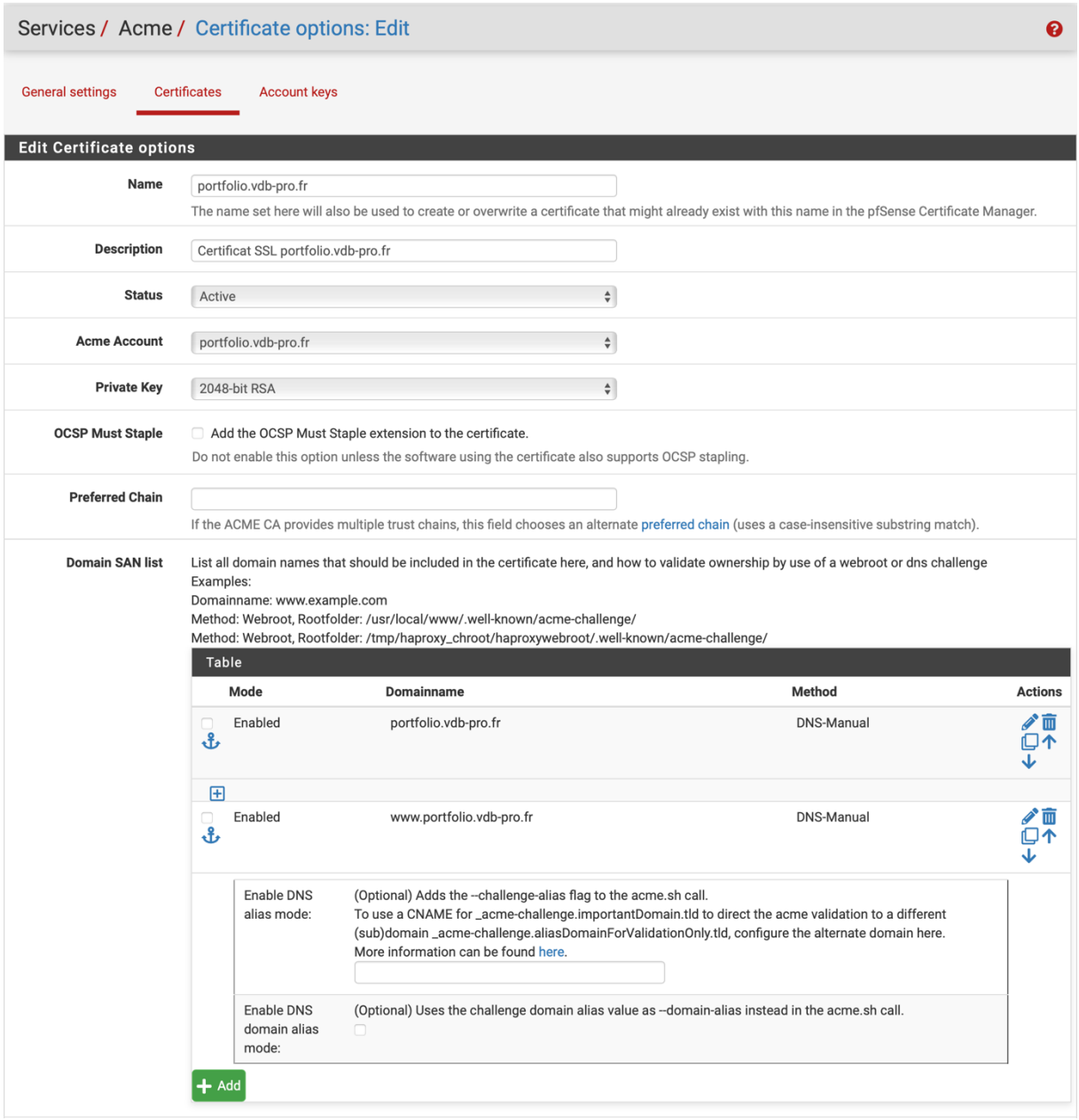
1. Accéder à l'onglet "Account Keys" :
    - Connectez-vous à l'interface de pfSense.
    - Allez dans **Services > Acme Certificates** et cliquez sur l'onglet **Account Keys**.
    - Cliquez sur le bouton **Add** pour ajouter une nouvelle clé.
2. Configurer la clé d'authentification :
    - Donnez un nom à la clé, par exemple : cert-mon-domaine.
    - Sélectionnez l'option **"Let's Encrypt Production ACME v2"** dans la liste déroulante **ACME Server**.
    - Cliquez sur **Create new account key** pour générer la clé d'authentification. Le champ **Account key** sera automatiquement rempli.
3. Enregistrer la clé :
    - Cliquez sur le bouton **Register ACME account key**.
    - Une icône de validation devrait s'afficher à côté du bouton.
    - Cliquez sur **Save** pour sauvegarder la clé.

Vous verrez maintenant la clé enregistrée dans la liste des **Account Keys**.

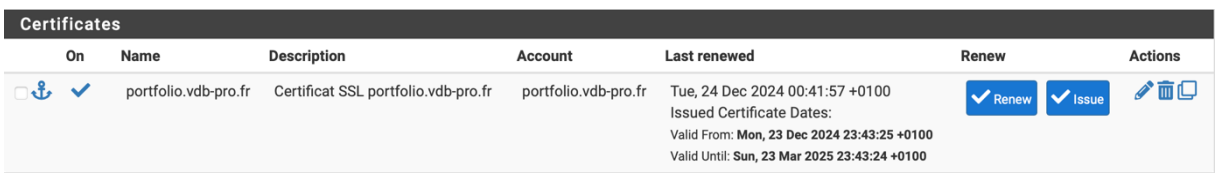


Demande d'un certificat Let's Encrypt

1. Accéder à l'onglet "Certificates" :
    - Rendez-vous dans l'onglet **Certificates** de la même section.
2. Créer un certificat :
    - Cliquez sur **Add** pour ajouter un nouveau certificat.
    - Renseignez les champs suivants :
        - **Name** : Saisissez un nom pour le certificat, par exemple : mon-domaine.
        - **Acme Account** : Sélectionnez la clé ACME créée à l'étape précédente.
        - **Domain SAN list** :
            - Cliquez sur **Add**.
            - Renseignez le champ **Domainname** avec le nom de domaine pour lequel vous souhaitez générer un certificat SSL (par exemple, mondomaine.com).
            - Sélectionnez la méthode **DNS-Manual**

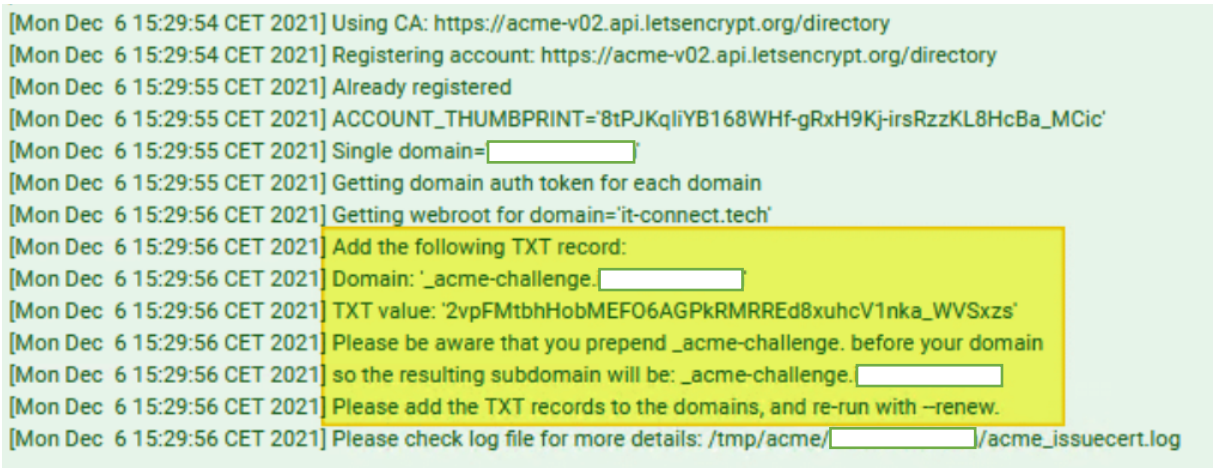


3. Configuration du renouvellement automatique :
   - Par défaut, le certificat est configuré pour se renouveler automatiquement tous les **60 jours** (dans l'option **Certificate renewal after** en bas de la page).
4. Sauvegarder et générer le certificat :
   - Cliquez sur **Save**.
   - Dans la liste des certificats, localisez le certificat nouvellement créé et cliquez sur le bouton **Issue**.

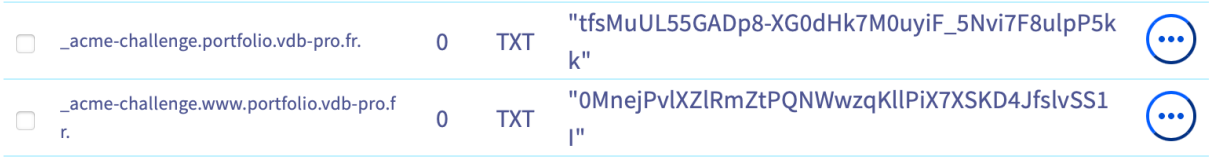


Validation via un enregistrement DNS

1. **Création de l'enregistrement TXT :**
   - Une fois que vous cliquez sur **Issue**, pfSense vous fournira une valeur spécifique pour un enregistrement DNS de type **TXT**. Par exemple :
     - **Nom** : _acme-challenge
     - **Valeur** : Une chaîne unique fournie par ACME.



2. **Ajouter l'enregistrement DNS :**
   - Connectez-vous à votre gestionnaire de domaine (par exemple OVH, Cloudflare, etc.).
   - Créez un nouvel enregistrement de type **TXT** avec les valeurs fournies.



3. **Finaliser la validation :**
   - Après avoir ajouté l'enregistrement DNS, revenez à pfSense et cliquez sur le bouton **Renew**.
4. **Vérifier la réussite :**
   - Si tout est configuré correctement, vous devriez voir une réponse positive avec des mentions comme **Cert success** et **BEGIN CERTIFICATE** dans les logs.

Les fichiers du certificat généré seront situés dans /tmp/acme/[nom_du_certificat]/.

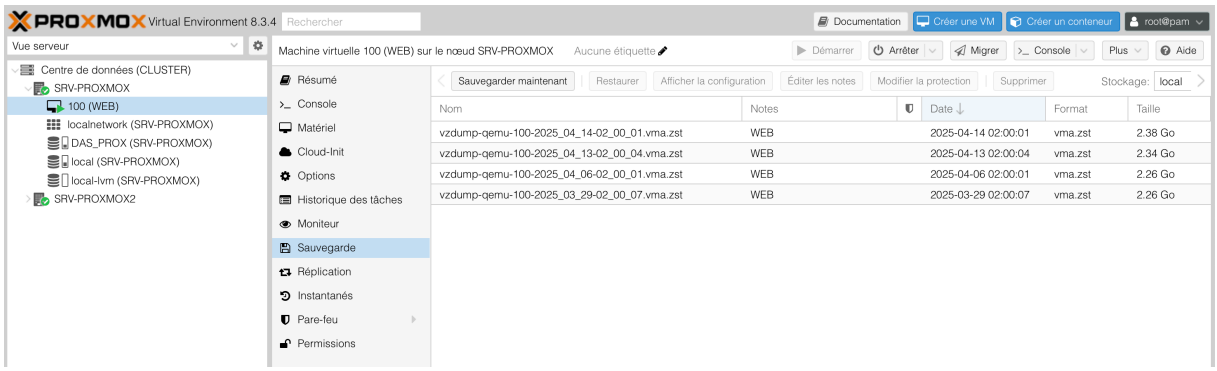
Résultat final

- Le certificat SSL sera automatiquement renouvelé grâce à Let's Encrypt.
- HAProxy sécurisera les connexions entre les visiteurs et votre serveur en utilisant ce certificat.
- Tout le trafic vers votre IP sera redirigé vers votre domaine avec HTTPS.

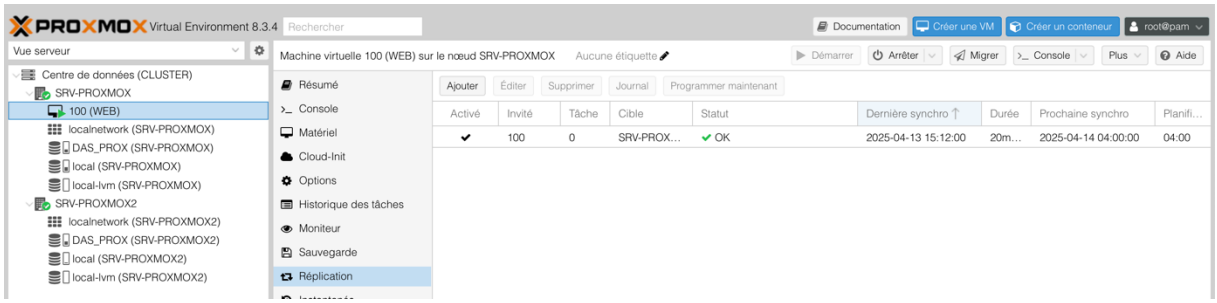
Le site est sécurisé par un certificat SSL valide et renouvelable automatiquement !

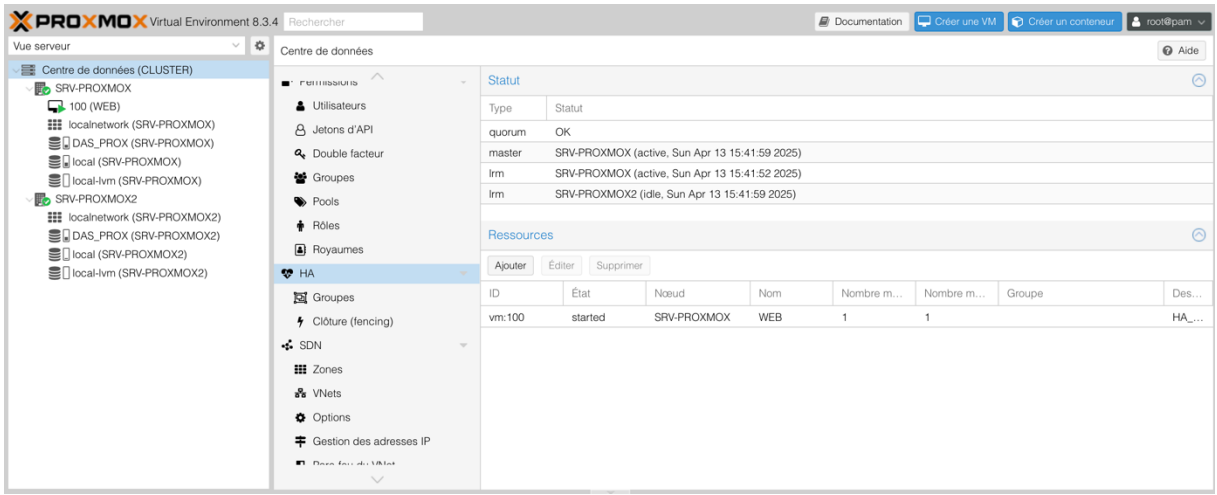
Gérer le patrimoine informatique

J'ai mis en place un système de sauvegardes quotidiennes pour garantir la sécurité et la pérennité des données. Ces sauvegardes sont automatisées via l'outil intégré de Proxmox, avec plusieurs points de rétention pour assurer un retour en arrière rapide en cas d'erreur ou d'incident.



En complément, j'ai également configuré un cluster Proxmox avec plusieurs nœuds, permettant la haute disponibilité des machines virtuelles via de la réplication géré par Proxmox. Cette architecture assure une continuité de service même en cas de panne d'un des serveurs, ce qui est essentiel pour garantir la fiabilité et la stabilité de l'infrastructure.

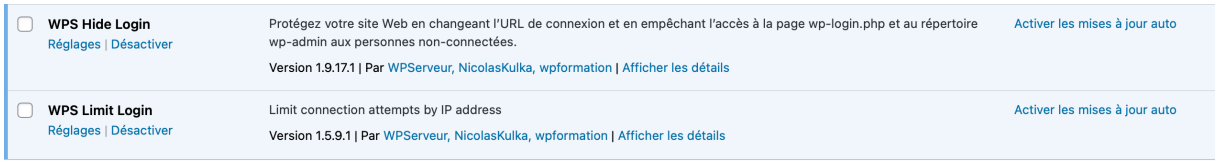




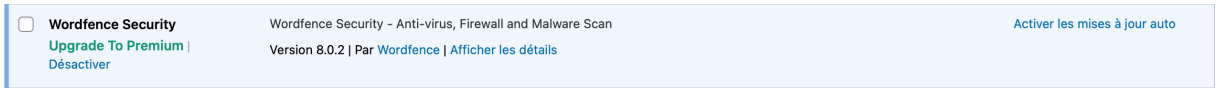
Sécurisation du Wordpress via des extensions

Pour protéger mon site WordPress, j'ai mis en place plusieurs extensions de sécurité :

1. WPS Limit Login :
    - Cette extension limite le nombre de tentatives de connexion pour prévenir les attaques par force brute.
    - Elle permet de bloquer temporairement une adresse IP après plusieurs tentatives échouées.
2. WPS Hide Login :
    - J'ai utilisé cette extension pour changer l'URL de la page de connexion WordPress, rendant celle-ci moins prévisible et plus difficile à cibler par des attaquants.



3. Wordfence :
    - Une extension complète pour la sécurité WordPress.
    - Fonctionnalités utilisées :
        - Scan de sécurité : Identifie les fichiers vulnérables, les plugins obsolètes ou les configurations potentiellement dangereuses.
        - Pare-feu intégré : Protège le site contre les intrusions et les attaques.
        - Blocage des IP malveillantes : Utilisation d'une base de données mondiale des IP suspectes pour bloquer automatiquement les connexions malveillantes.
        - MFA : Obtenir un code avant connexion qui se reinitialise toutes les 30 secondes



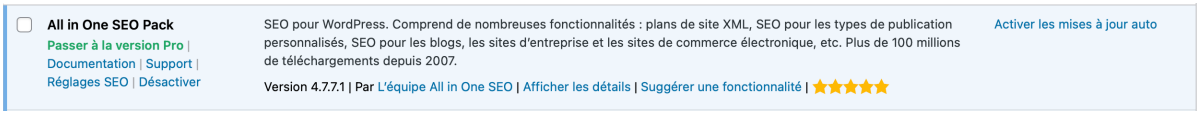
Grâce à ces outils, mon site est mieux protégé contre diverses menaces courantes, tout en restant accessible et performant. La combinaison de ces extensions permet une sécurité à plusieurs niveaux, renforçant ainsi la protection de mon environnement web.

Mise en place d’une extension de SEO pour le référencement

J'avais pour objectif de rendre mon site accessible en ligne et facilement trouvable via une recherche dans les navigateurs, un processus appelé **référencement**. Ce dernier permet d’améliorer la visibilité du site sur les moteurs de recherche comme Google, en optimisant son contenu et sa structure pour qu’il apparaisse dans les premiers résultats lors de recherches pertinentes.

Pour optimiser le référencement de mon site WordPress et améliorer sa visibilité sur les moteurs de recherche, j'ai installé et configuré l'extension **All In One SEO**.

1. Fonctionnalités utilisées :
    - **Optimisation des balises méta** : Ajout de titres, descriptions, et mots-clés adaptés pour chaque page et article, afin de répondre aux critères des moteurs de recherche.
    - **Analyse SEO en temps réel** : Assistance pour identifier les éléments à améliorer, comme la densité des mots-clés, la lisibilité du contenu ou la structure des URLs.
    - **Sitemaps automatiques** : Génération de sitemaps XML pour faciliter l'exploration du site par les robots de Google.
    - **Optimisation des réseaux sociaux** : Ajout des balises Open Graph et Twitter Cards pour un meilleur partage sur les plateformes sociales.
2. Impact attendu :
    - Amélioration du positionnement du site dans les résultats de recherche.
    - Augmentation du trafic organique grâce à un contenu optimisé.
    - Une expérience utilisateur améliorée avec des URLs propres et une structure claire.

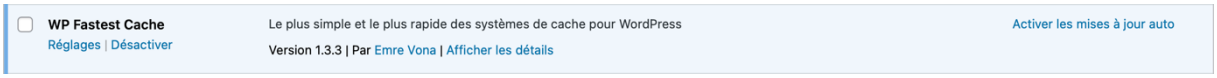


L’extension **All In One SEO** est un outil essentiel pour rendre mon site plus compétitif en ligne tout en restant simple à gérer.

Mise en place d’une extension de cache

Pour améliorer les performances de mon site web, j'ai installé l'extension **WP Fastest Cache**, spécialisée dans la gestion et l'optimisation du cache.

1. Optimisation des ressources :
    - Cette extension permet de **créer un cache pour les pages web**, ce qui réduit les temps de chargement en évitant de générer les pages dynamiquement à chaque visite.
    - Elle optimise également les éléments tels que les images, vidéos, blocs, et fichiers CSS/JS, garantissant un affichage fluide et rapide.
2. Préchargement des pages :
    - L'une des fonctionnalités clés de WP Fastest Cache est le **préchargement des pages**.
    - Cela signifie que la page est mise en cache et prête à être affichée instantanément, avant même que l'utilisateur ne clique dessus. Cette technique améliore significativement l’expérience utilisateur.
3. Réduction des requêtes et optimisation du code :
    - L’extension réduit le nombre de requêtes HTML envoyées au serveur et propose une compression des fichiers HTML, CSS et JS.
    - Cette optimisation du code contribue également à améliorer le **référencement SEO**, car les moteurs de recherche valorisent les sites rapides et bien structurés.

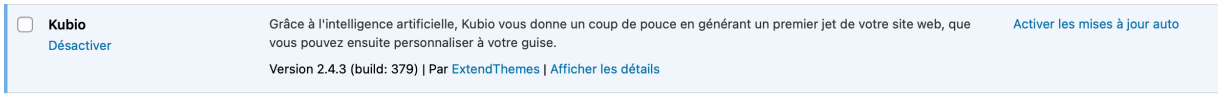


En utilisant **WP Fastest Cache**, mon site gagne non seulement en rapidité et efficacité, mais il devient également plus compétitif en termes de référencement sur des plateformes comme Google. Cela répond aux attentes des visiteurs tout en maximisant les performances techniques de mon infrastructure.

Mise en place d’une extension de personnalisation

Pour améliorer l'expérience de création et de gestion de mon site, j'ai installé une extension appelée **Kubio**, qui facilite grandement la conception de pages web.

1. Création de pages personnalisées :
    - Kubio fonctionne sur un système de **blocs personnalisés**, ce qui me permet de construire des pages de manière visuelle et intuitive.
    - Chaque élément du site, comme les sections, les colonnes, ou les boutons, peut être configuré et agencé selon mes besoins sans avoir à manipuler directement du code.
2. Gains en productivité :
    - Cette extension simplifie et accélère le processus de création de contenu, ce qui est particulièrement utile pour un projet évolutif comme le mien.
    - Grâce à sa bibliothèque de blocs prédéfinis et d'options de personnalisation, je peux rapidement donner un aspect professionnel à mon site.
3. Flexibilité et design sur mesure :
    - Kubio offre une grande flexibilité pour adapter le design à mon identité visuelle.
    - Cela me permet d’ajouter une touche personnelle à chaque page, tout en assurant une cohérence esthétique sur l’ensemble du site.

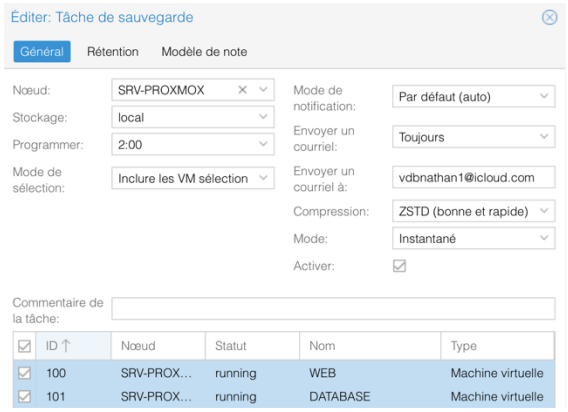


En utilisant cette extension, je bénéficie d’une solution performante et accessible pour concevoir un site moderne et fonctionnel, sans dépendre exclusivement de compétences en développement web. Cela répond parfaitement à mon besoin de flexibilité et de simplicité pour un projet en constante évolution.

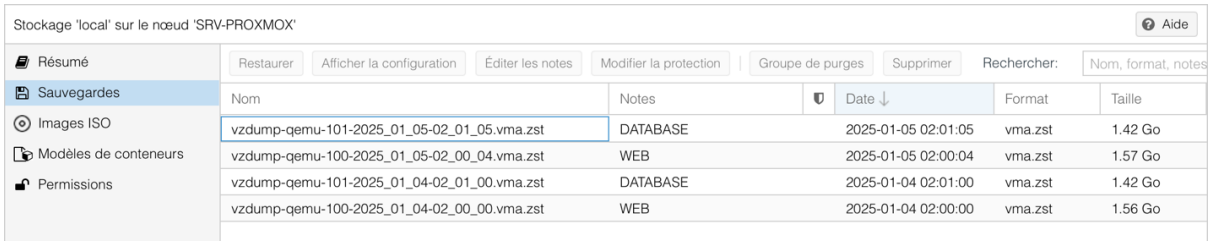
Mise en place des sauvegardes pour les machines virtuelles

La mise en place de sauvegardes est une étape cruciale pour garantir la **résilience** et la **sécurité** de l'infrastructure. Étant donné que mon site sera accessible au public, il est exposé à des menaces potentielles comme des attaques malveillantes ou des erreurs humaines lors des modifications. Pour éviter des pertes de données ou des interruptions de service, j'ai mis en place un **système de sauvegarde** performant.

1. Sauvegardes via Proxmox :
    - J'ai configuré des sauvegardes automatiques des machines virtuelles (VM) directement sur **Proxmox**.
    - Les sauvegardes s'effectuent tous les jours à **2h du matin**, sur un **disque SSD dédié** intégré dans Proxmox, garantissant une vitesse et une fiabilité optimales.



2. Politique de rétention :
    - Actuellement, deux points de rétention sont configurés pour conserver les sauvegardes les plus récentes.
    - Je prévois d'ajouter des points de rétention supplémentaires :
        - **1 point hebdomadaire** pour conserver une sauvegarde de la semaine.
        - **1 point mensuel** pour conserver une sauvegarde à long terme.
3. Évolutions prévues :
    - À l'avenir, je souhaite **héberger un NAS** (Network Attached Storage) chez moi pour y sauvegarder mes VM. Cela offrirait une solution de sauvegarde secondaire en cas de défaillance du disque SSD principal.
    - Cette configuration permettrait également d'avoir des sauvegardes accessibles et sécurisées sur un périphérique séparé de l'hyperviseur.



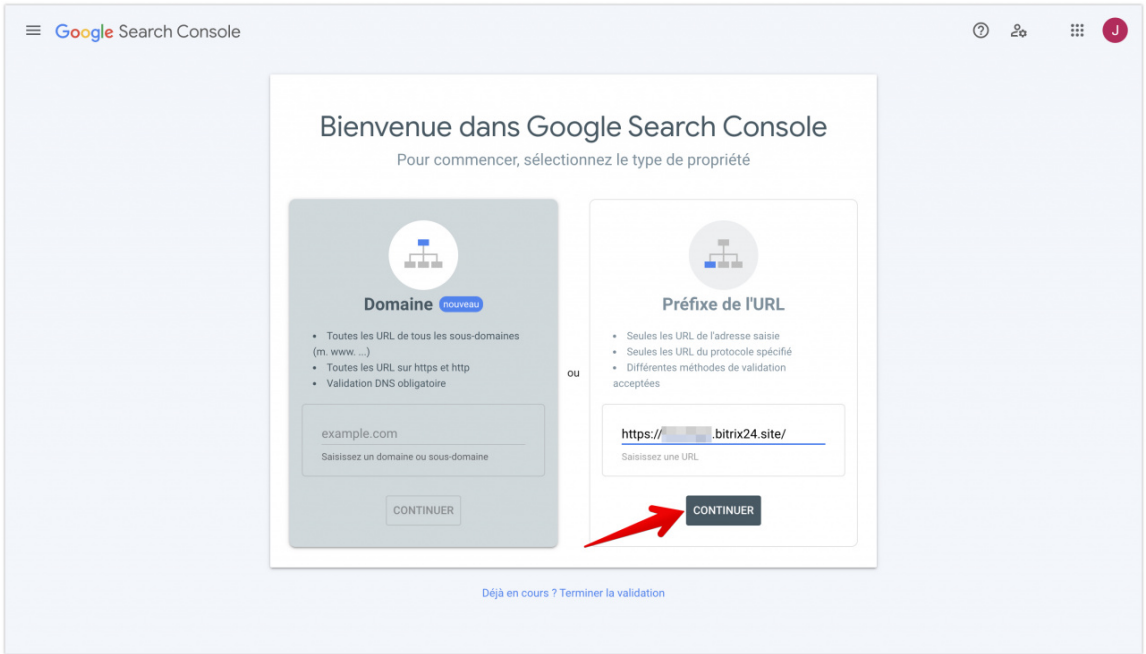
Avec ce système en place, mon infrastructure est mieux protégée contre les incidents et les erreurs, garantissant ainsi la continuité de mon site web.

Référencement de mon site dans Google et Bing

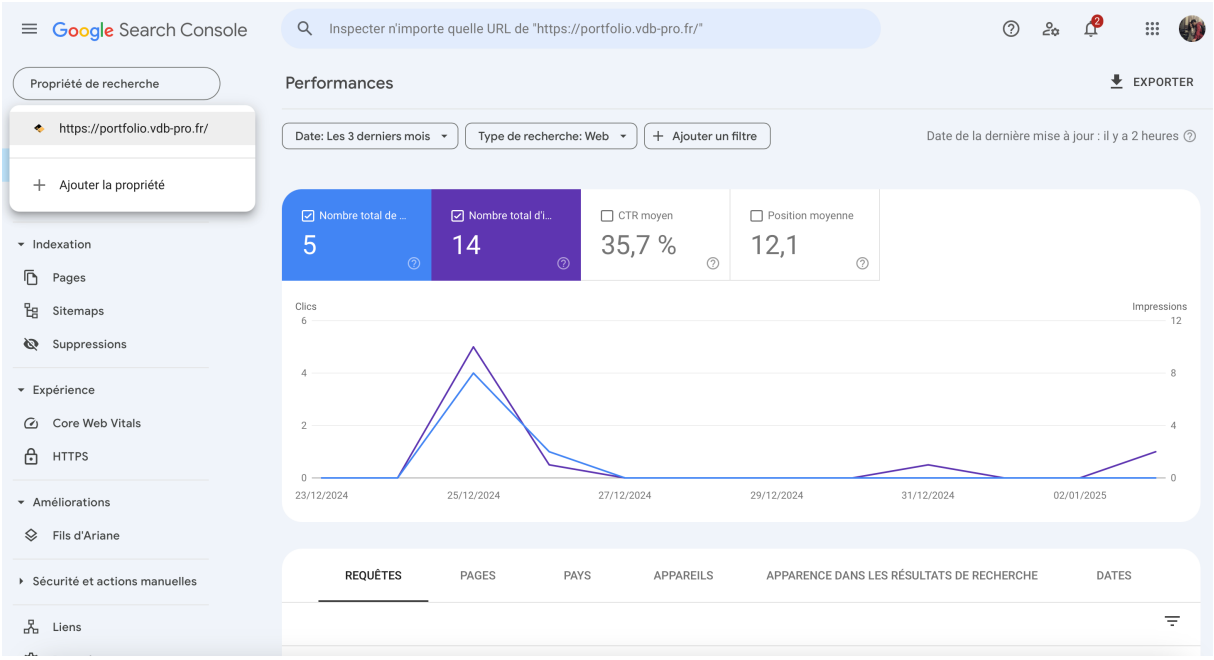
Pour rendre mon site accessible via une recherche en tapant "portfolio vdb pro", je l'ai inscrit sur **Google Search Console**et **Bing Webmaster Tools**. Voici les étapes que j'ai suivies :

1. Accéder à Google Search Console :
   - Je suis allé sur la page de connexion de Google Search Console et ai cliqué sur **Commencer maintenant**.

2. Ajout de l'URL du site :
   - J'ai choisi l'option **Préfixe de l'URL**, saisi l'URL de mon site web, puis cliqué sur **Continuer**.



3. Validation de la propriété :
    - Google propose plusieurs méthodes pour prouver que l'on est propriétaire du site :
        - **Balise HTML** : Ajout d'une balise meta dans le code HTML de la page principale.
        - **Fournisseur de nom de domaine** : Liaison directe avec mon fournisseur DNS.
        - **Google Analytics** : Utilisation d'un compte Analytics lié au site.
        - **Google Tag Manager** : Vérification via Tag Manager.
    - J'ai opté pour la méthode **Fournisseur de nom de domaine**, car elle permet de valider directement la propriété via les paramètres DNS de mon nom de domaine.
4. Validation et confirmation :
    - Une fois la méthode choisie, j'ai suivi les instructions pour configurer les entrées DNS fournies par Google dans l'interface de gestion de mon nom de domaine chez OVH.
    - Après l'ajout, la propriété de mon site a été validée, et il est désormais indexé par Google.



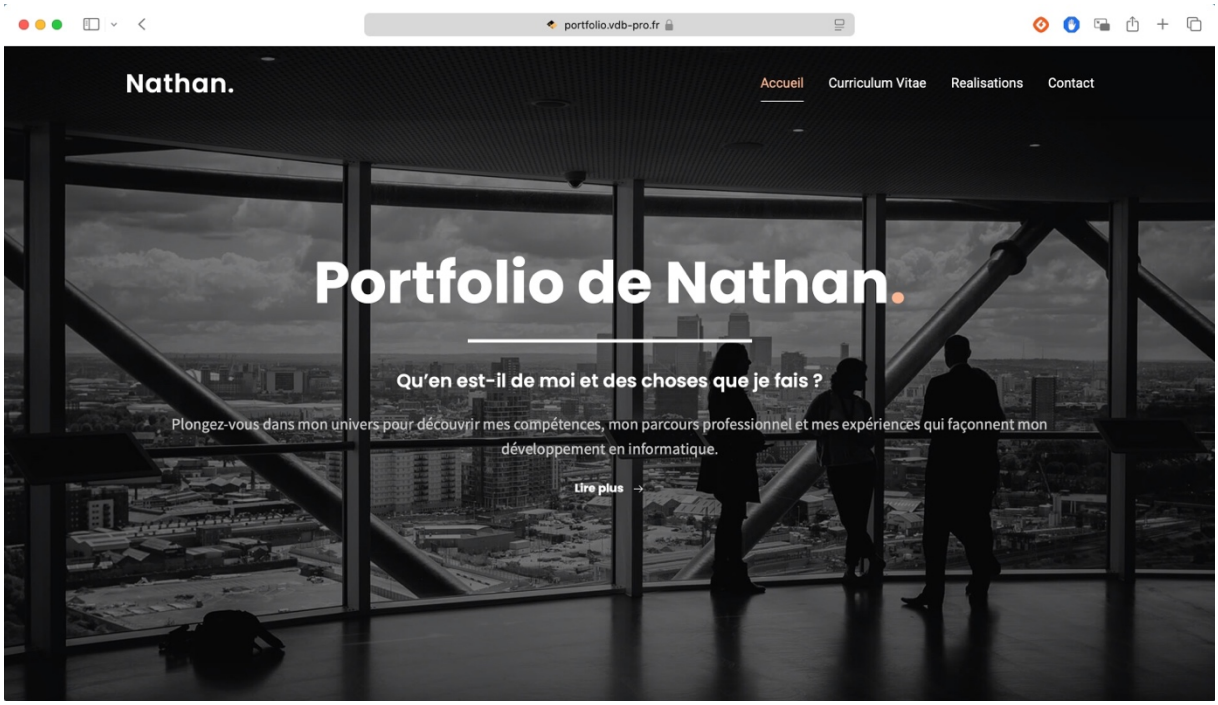
5. Ajout sur Bing Webmaster Tools :
    - J'ai suivi une procédure similaire pour Bing en accédant à Bing Webmaster Tools.
    - J'ai ajouté mon site, configuré les DNS pour prouver ma propriété et validé l'inscription.

Ces démarches permettent à mon site d'être indexé et facilement trouvé via les moteurs de recherche. Cette visibilité est essentielle pour améliorer mon référencement et faciliter l'accès des utilisateurs et examinateurs.

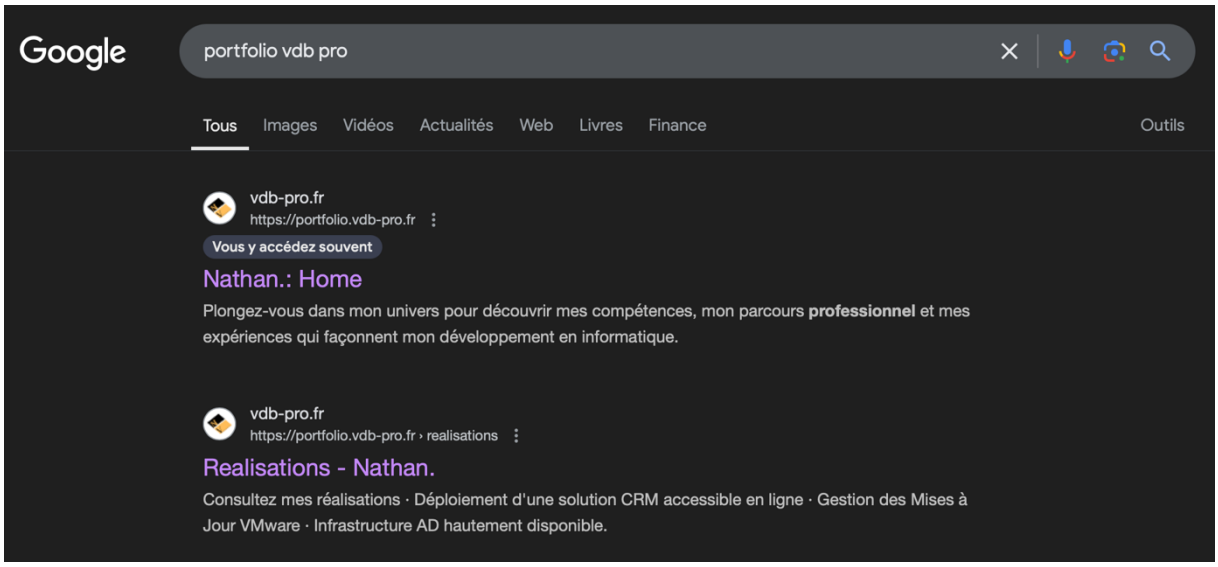
https://developers.google.com/search/docs/fundamentals/do-i-need-seo?hl=fr

Partie 2 – Validation

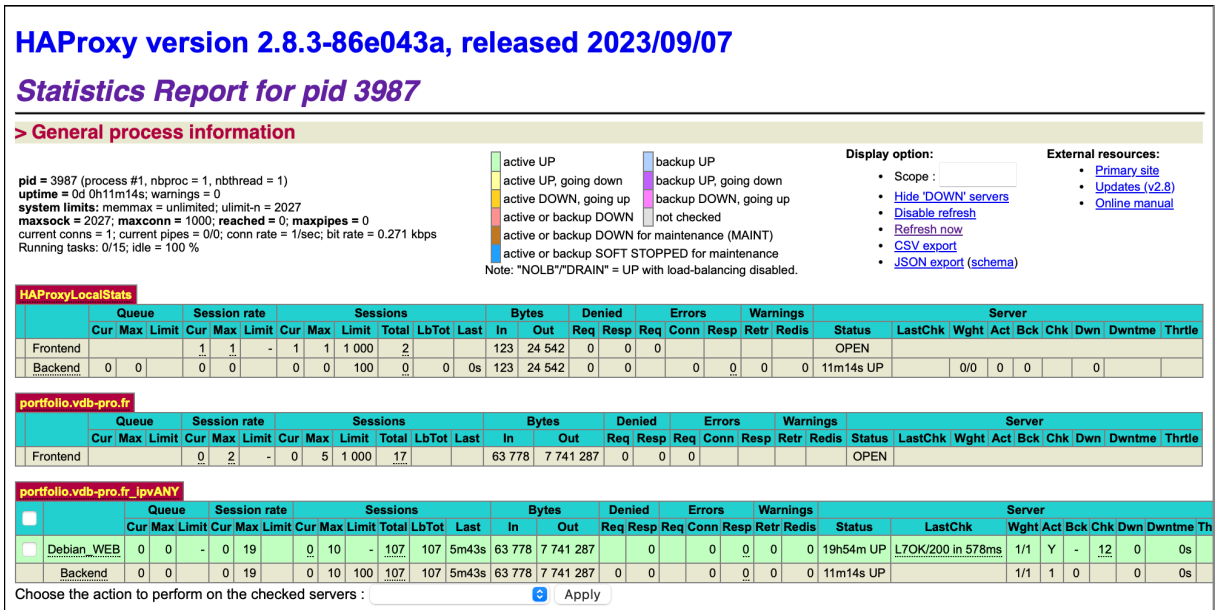
Avec toute la procédure décrite ci-dessus, j'ai réussi à mettre en place une infrastructure réseau **stable** et **sécurisée**, ainsi qu'un site web fonctionnel, accessible publiquement via https://portfolio.vdb-pro.fr.



Vous pouvez également accéder à mon site en recherchant "**portfolio vdb pro**" sur Google.fr, ce qui permettra de vérifier que le site est bien référencé et qu'il apparaît correctement dans les résultats de recherche.



Nous pouvons donc suivre le trafic du reverse proxy via **Stats**, comme illustré dans la photo ci-dessous.



HAProxy version 2.8.3-86e043a, released 2023/09/07

Statistics Report for pid 3987

> General process information

pid = 3987 (process #1, nbproc = 1, nbthread = 1)
uptime = 0d 0h11m14s; warnings = 0
system limits: memmax = unlimited; ulimit-n = 2027
maxsock = 2027; **maxconn =** 1000; **reached =** 0; **maxpipes =** 0
current conns = 1; current pipes = 0/0; conn rate = 1/sec; bit rate = 0.271 kbps
Running tasks: 0/15; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
- Scope :
- Hide 'DOWN' servers
- Disable refresh
- Refresh now
- CSV export
- JSON export (schema)

External resources:
- Primary site
- Updates (v2.8)
- Online manual

HAProxyLocalStats

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				1	1	-	1	1	1 000	2			123	24 542	0	0	0					OPEN								
Backend	0	0		0	0		0	0	100	0	0	0s	123	24 542	0	0		0	0	0	0	11m14s UP		0/0	0	0		0		

portfolio.vdb-pro.fr

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	2	-	0	5	1 000	17			63 778	7 741 287	0	0	0					OPEN								

portfolio.vdb-pro.fr_ipvANY

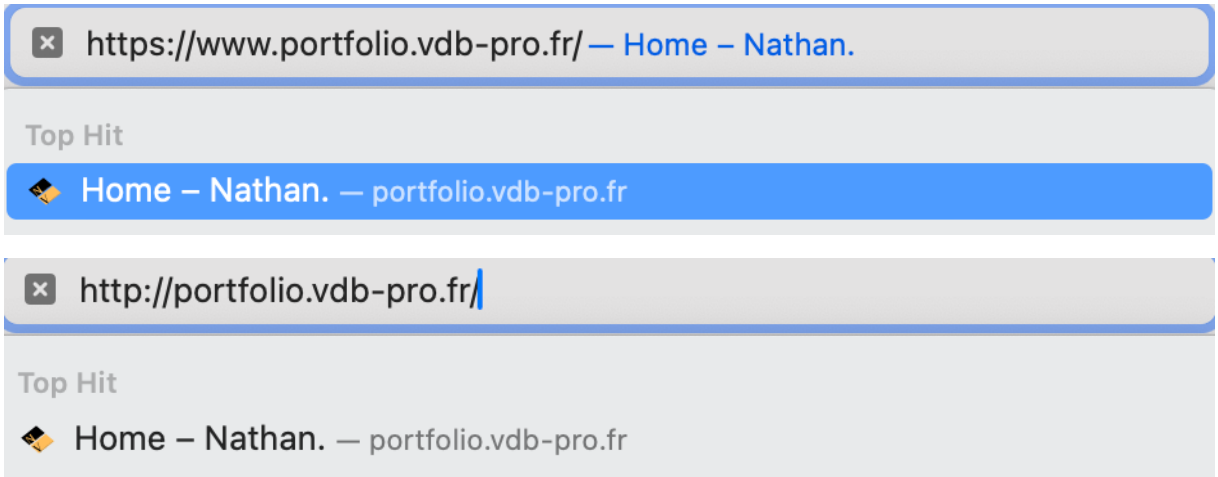
	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Th
Debian_WEB	0	0	-	0	19		0	10	-	107	107	5m43s	63 778	7 741 287		0		0	0	0	0	19h54m UP	L7OK/200 in 578ms	1/1	Y	-	12	0	0s	
Backend	0	0		0	19		0	10	100	107	107	5m43s	63 778	7 741 287	0	0		0	0	0	0	11m14s UP		1/1	1	0		0	0s	

Choose the action to perform on the checked servers : Apply

On peut également vérifier si les sauvegardes fonctionnent correctement en consultant les logs et les points de restauration configurés dans Proxmox, ce qui garantit que les machines virtuelles sont bien sauvegardées et peuvent être restaurées en cas de besoin.

Nom	Notes		Date ↓	Format	Taille
vzdump-qemu-101-2025_01_10-02_01_26.vma.zst	DATABASE		2025-01-10 02:01:26	vma.zst	1.42 Go
vzdump-qemu-100-2025_01_10-02_00_04.vma.zst	WEB		2025-01-10 02:00:04	vma.zst	2.18 Go
vzdump-qemu-101-2025_01_09-02_01_24.vma.zst	DATABASE		2025-01-09 02:01:24	vma.zst	1.42 Go
vzdump-qemu-100-2025_01_09-02_00_02.vma.zst	WEB		2025-01-09 02:00:02	vma.zst	2.19 Go

Enfin, nous pouvons vérifier si les redirections HTTP et WWW fonctionnent correctement en testant l'accès à mon site via différentes URL, telles que http://portfolio.vdb-pro.fr ou https://www.portfolio.vdb-pro.fr, et en s'assurant que celles-ci redirigent automatiquement vers la version sécurisée https://portfolio.vdb-pro.fr.



Partie 3 – Veille technologique

Dans le cadre de cette **veille technologique**, plusieurs axes d'amélioration ont été identifiés pour le projet, et des choix techniques différents auraient pu être envisagés. Je vais explorer les améliorations possibles et justifier certains choix, tant au niveau des logiciels que du matériel, en expliquant pourquoi ils n'ont pas été retenus pour cette infrastructure.

Améliorations futures :

1. **Amélioration des performances des machines virtuelles :** Afin d'augmenter la capacité et la performance des machines virtuelles (VM), il aurait été pertinent d'utiliser des ordinateurs avec des configurations matérielles plus puissantes. Cela aurait permis de garantir une évolution fluide et une gestion des ressources plus efficace à mesure que le nombre de VM et les exigences de performances augmentent.
2. **Ajout d'un NAS pour externaliser les sauvegardes :** Dans un souci de sécurité et pour améliorer la résilience du système, l'ajout d'un NAS (Network Attached Storage) chez moi serait une prochaine étape. Cela permettrait de déporter les sauvegardes dans un autre emplacement physique, augmentant ainsi la sécurité des données et minimisant les risques de perte d'informations critiques.

Choix logiciels :

1. **Apache vs Nginx pour l'hébergement web :** J'ai initialement envisagé d'utiliser Nginx pour l'hébergement de mon site web, car il est très performant et léger. Cependant, connaissant mieux Apache, j'ai préféré l'utiliser, car il est plus simple à configurer et bénéficie d'un vaste écosystème de modules et d'extensions. Toutefois, Nginx aurait été une alternative très intéressante pour la gestion de trafic, notamment en tant que reverse proxy.
2. **HAProxy vs Squid pour le reverse proxy :** Bien que Squid soit une solution robuste pour la gestion de proxy, j'ai choisi d'implémenter HAProxy en raison de sa simplicité d'utilisation et de sa capacité à gérer facilement les tâches de reverse proxy et d'équilibrage de charge. Squid est une alternative intéressante, mais HAProxy répondait mieux à mes besoins pour ce projet précis.
3. **HTML vs WordPress pour la création de site web :** Une alternative à l'utilisation de WordPress aurait été de créer un site en HTML pur, en apprenant le langage et en utilisant un template Bootstrap pour le design. J'ai d'ailleurs commencé cette démarche, mais après réflexion, j'ai opté pour WordPress en raison de sa simplicité d'utilisation et de son immense catalogue d'extensions. WordPress facilite la mise en place de fonctionnalités de sécurité et d'optimisation, ce qui était crucial pour ce projet. Bien qu'une solution en HTML pur offre une plus grande liberté et soit totalement gratuite, la flexibilité et la richesse de WordPress ont été décisives dans ce choix.

Choix matériels :

1. **UTM Sophos vs pfSense :** Une autre option aurait été d'utiliser un UTM Sophos au lieu de pfSense. Les UTM Sophos sont de bonnes solutions de sécurité, mais elles nécessitent des licences payantes, ce qui n'était pas justifié pour ce projet. pfSense, étant une solution gratuite et très robuste, a été un choix plus adapté, d'autant plus que je souhaitais maîtriser l'aspect sécurité et réseau de manière autonome sans coût supplémentaire.
2. **ESXi vs Proxmox pour la gestion des VM :** Bien que je sois plus familier avec VMware ESXi, j'ai choisi Proxmox pour gérer les machines virtuelles. En effet, Proxmox est plus léger en termes de ressources et parfaitement adapté à des machines moins puissantes, comme celles utilisées pour ce projet. Proxmox est également très bien intégré avec des solutions open-source et offre une gestion simplifiée des containers LXC, ce qui est un atout pour la flexibilité et l'évolutivité de l'infrastructure.

Mise en place d'une solution de supervision :

Afin de garantir une surveillance continue et un retour détaillé sur la performance de l'infrastructure et des différents équipements du réseau, une solution de supervision aurait été un ajout pertinent à ce projet. Pour cela, des outils comme **Zabbix**, **Nagios**, ou **Prometheus** auraient pu être utilisés pour surveiller les performances des serveurs, des machines virtuelles, ainsi que du trafic réseau.

- **Zabbix** : Ce logiciel open-source est parfait pour surveiller l'ensemble des équipements d'une infrastructure. Il permet de collecter des données de performance, d'alerter en cas d'anomalies et de générer des rapports détaillés sur les ressources utilisées (CPU, RAM, disque, etc.). Il est très flexible et évolutif, et peut s'intégrer à une large variété d'équipements et de services.

La mise en place de ce type de solution de supervision aurait permis de **détecter rapidement toute anomalie** dans les machines virtuelles, les équipements réseau ou les services web, et de réagir de manière proactive avant qu'un incident ne survienne. De plus, cela permet de surveiller l'évolution de la **charge système** et **d'anticiper** les besoins futurs en termes de ressources, ce qui est essentiel pour assurer la pérennité du projet à long terme.

Ces différents choix et pistes d'amélioration montrent l'importance de toujours évaluer les solutions disponibles en fonction des **besoins** et des **contraintes** spécifiques d'un projet. Je continue à explorer de nouvelles technologies et solutions pour **optimiser** cette infrastructure, en tenant compte des **évolutions** possibles à long terme.

Organiser son développement professionnel

Pour développer mes compétences et m'améliorer continuellement, j'ai mis en place une démarche personnelle d'apprentissage. Je me suis formé de manière autonome en consultant des tutoriels en ligne, en suivant des documentations officielles (comme celles de Proxmox, WordPress ou pfSense), et en testant différentes configurations en environnement réel ou virtualisé. J'ai également appris à résoudre des problèmes concrets en faisant face aux erreurs et en cherchant des solutions par moi-même. Cette méthode m'a permis d'acquérir une meilleure compréhension des systèmes, de renforcer mes connaissances techniques et de gagner en autonomie dans mes projets.