BERRADA Ilyes 2025

Mise en place d'un reverse proxy sécurisé avec pfSense, HAProxy et Nginx

---

À quoi ça sert :

- Sécuriser l'accès aux services internes en interposant un reverse proxy entre les utilisateurs et les serveurs web.
- Centraliser les connexions HTTP/HTTPS et les répartir vers plusieurs services selon des règles (URL, ports, sous-domaines, etc.).
- Protéger l'accès aux services par une authentification centralisée au niveau du proxy.
- Appliquer une politique de sécurité réseau unifiée via un point d'entrée unique (SSL, filtrage, contrôle des accès).
- Améliorer la modularité de l'infrastructure en rendant les services internes invisibles directement depuis l'extérieur.

pfSense :

Routeur/firewall qui héberge le service HAProxy. Il jouera le rôle de reverse proxy, recevant les connexions externes et les redirigeant de manière conditionnelle vers les services internes. Il permet également la gestion des certificats et l'application de l'authentification HTTP.

HAProxy (sur pfSense) :

Composant principal du reverse proxy, chargé de gérer les entrées (frontends), les destinations internes (backends) et les règles d'accès. Il permet de sécuriser et contrôler le trafic HTTP(S) en intégrant une authentification simple (HTTP Basic Auth) ou avancée.

Serveurs Web (Ubuntu – Apache2 ou Nginx) :

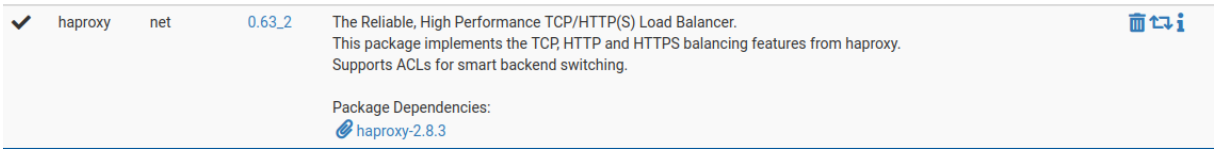
Machines cibles hébergeant les différents services internes (intranet, page de test, outils internes...). Elles ne sont pas exposées directement au client, seules les connexions validées via le proxy y accèdent.

Client (poste de test utilisateur) :

Ordinateur connecté au réseau interne, utilisé pour tester l'accès aux différents services web via le reverse proxy, avec ou sans authentification, et valider le bon fonctionnement du routage conditionnel et des règles de sécurité.

Nous avons dans un premier temps configuré un serveur web Apache sur notre machine Ubuntu. Comme le montre la capture d'écran, une page PHP simple a été mise en place afin de vérifier le bon fonctionnement du service. L'accès à cette page via l'adresse IP locale (ici 192.168.1.50) confirme que le serveur Apache est opérationnel et prêt à être utilisé comme backend pour notre futur reverse proxy.

Nous pouvons maintenant installer HAProxy sur pfSense afin de gérer la redirection sécurisée du trafic HTTP/HTTPS vers notre serveur web en backend.

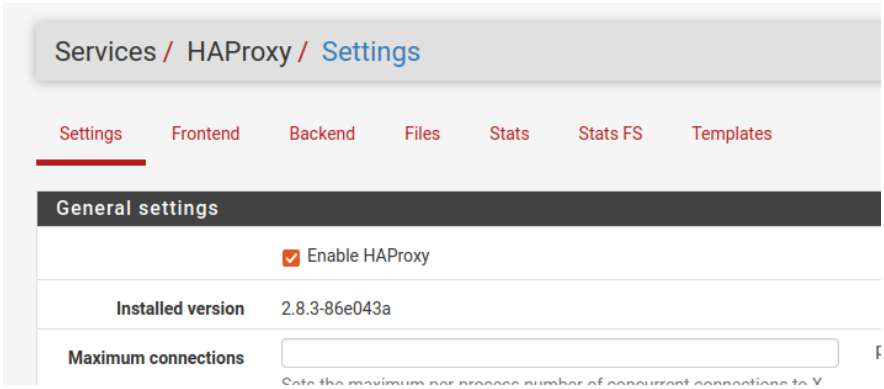


Dans un premier temps, nous activons HAProxy depuis l'onglet **Settings**. Il est nécessaire de cocher la case **Enable HAProxy** pour que le service soit fonctionnel.

Ensuite, nous définissons une limite de connexions simultanées raisonnable dans le champ **Maximum connections**. Ici, nous choisissons la valeur 10000, ce qui permet de supporter un grand nombre de requêtes tout en respectant les capacités du système.

Les autres paramètres peuvent être laissés par défaut. Aucun paramètre supplémentaire n'est requis dans le champ **Global Advanced pass thru** à ce stade du projet.

Cette configuration permet de préparer HAProxy à gérer les requêtes entrantes avant de mettre en place les règles de redirection vers le backend.

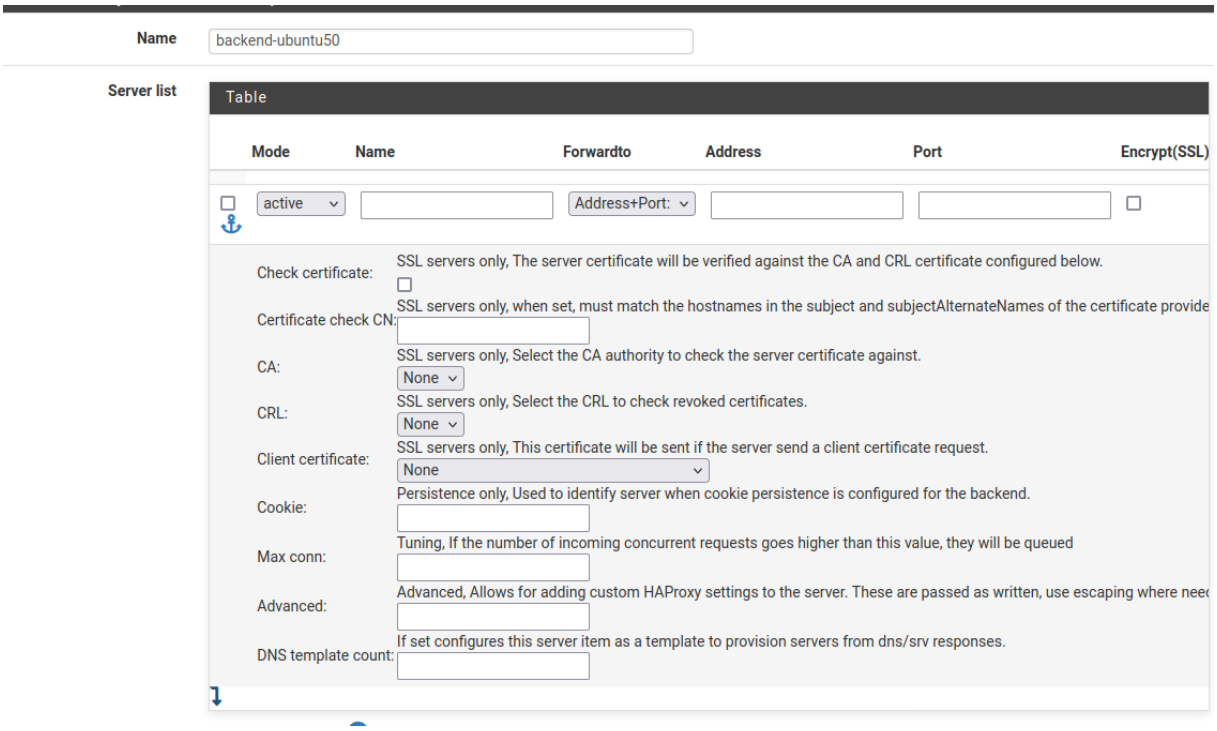


Nous configurons ici un **pool de serveurs backend** nommé backend-ubuntu50, qui redirigera les requêtes vers le serveur Ubuntu hébergeant notre service web.

Champs à compléter :

- **Mode** : active
- **Name** : ubuntu50 (nom du serveur dans le pool)
- **Forwardto** : Address+Port
- **Address** : 192.168.1.50 (adresse IP de ton serveur Ubuntu avec Apache)
- **Port** : 80 (port standard HTTP)
- **Encrypt(SSL)** : décoché (puisque le backend ne sert pas en HTTPS)

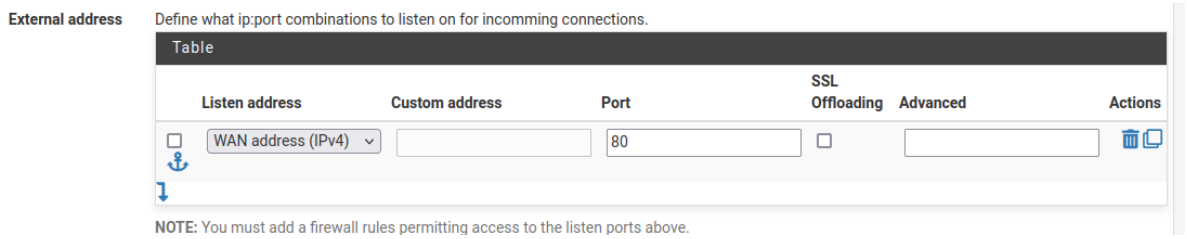
Cette étape permet à HAProxy de savoir vers quelle machine il doit rediriger les requêtes entrantes une fois qu'elles sont acceptées par le frontend.



nous définissons l'interface et le port sur lesquels HAProxy doit écouter les connexions entrantes.

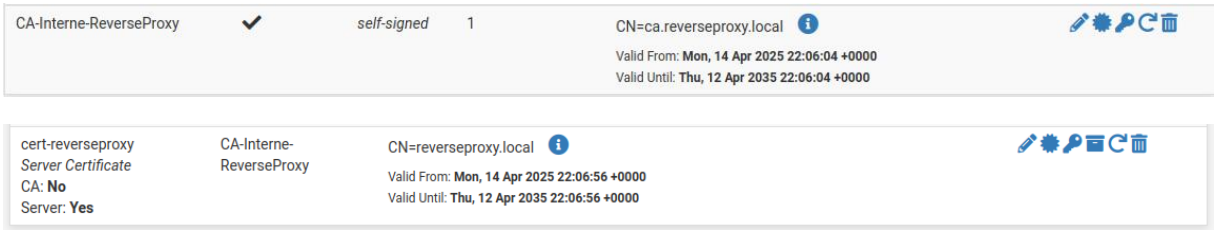
Configuration :

- **Listen address** : WAN address (IPv4)
  Cela signifie que HAProxy écoutera sur l'adresse IP publique (ou celle assignée à l'interface WAN de pfSense).
- **Port** : 80
  Il s'agit ici du port HTTP standard, utilisé pour recevoir les requêtes en clair.

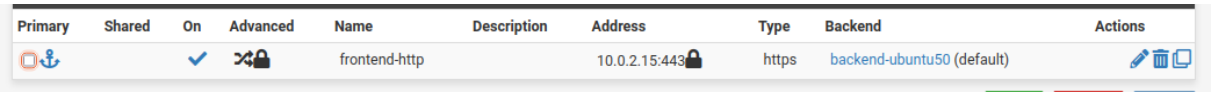


À présent, une machine cliente configurée sur le même réseau **NAT** que pfSense a été utilisée pour tester l'accès au service web. En entrant l'adresse **WAN de pfSense (10.0.2.15)** dans le navigateur, nous avons bien été redirigés vers le serveur Apache situé sur la machine Ubuntu, comme en atteste l'affichage de la page "Backend Ubuntu Service Web". Cela confirme que HAProxy intercepte correctement les requêtes entrantes et les redirige vers le bon serveur backend.

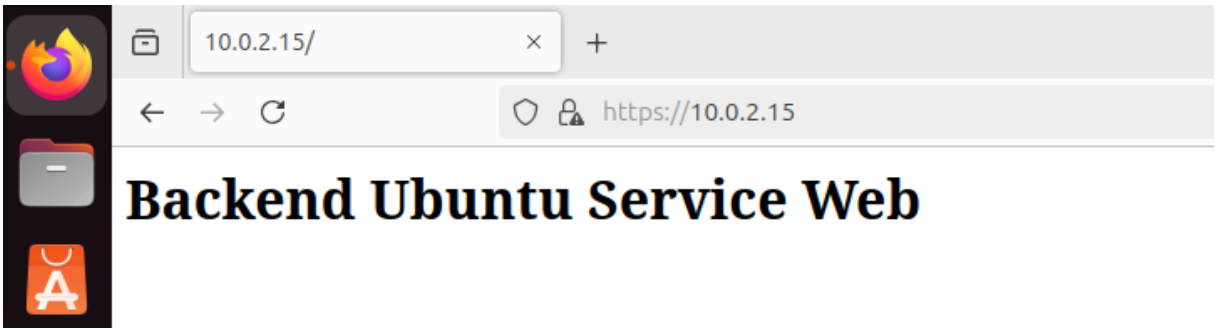
Nous avons créé une autorité de certification interne auto-signée pour générer un certificat SSL. Cela permet de sécuriser les connexions au reverse proxy via HTTPS.



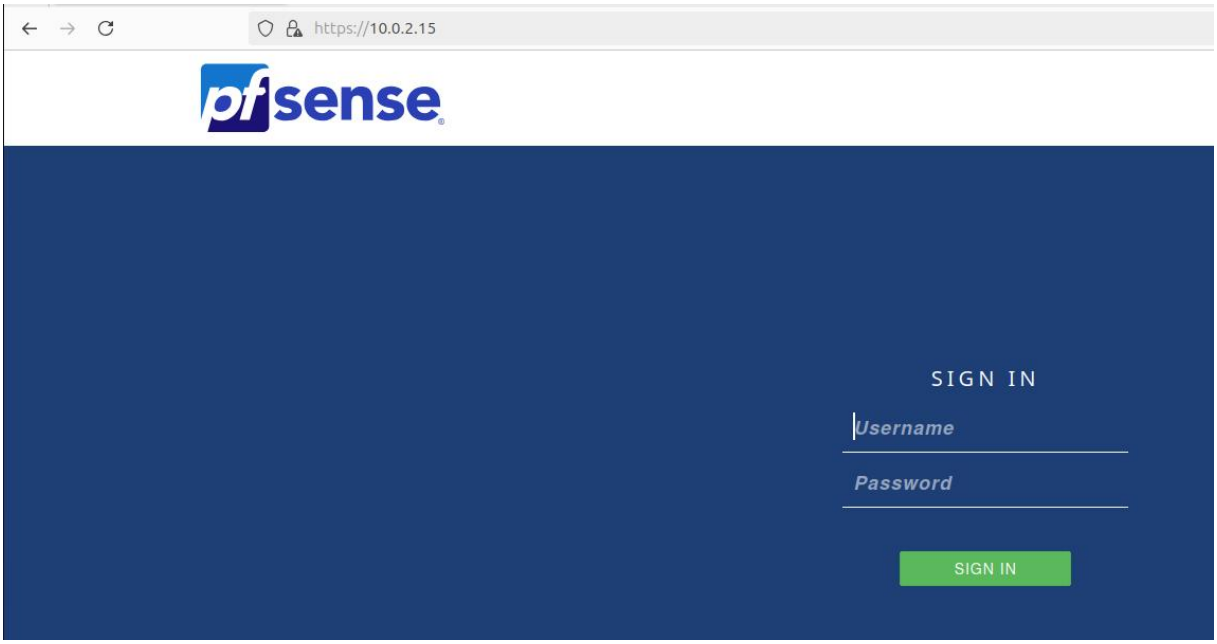
À présent, nous avons associé notre frontend HTTPS au backend configuré précédemment. Cette liaison permet de rediriger les connexions entrantes sécurisées vers notre serveur web Ubuntu.



Nous testons maintenant l'accès au service web via l'adresse WAN de pfSense. La redirection HTTPS vers le backend Ubuntu est fonctionnelle, comme l'indique l'affichage de la page web configurée sur Apache.



Afin de valider que le rôle de HAProxy est bien opérationnel, nous allons temporairement désactiver le service. En accédant à l'adresse IP du WAN depuis un navigateur, nous devrions alors être redirigés vers l'interface web de pfSense. Ce test permet de démontrer que lorsque HAProxy est actif, c'est bien lui qui intercepte les connexions et redirige le trafic vers le serveur web backend.



Conclusion

Grâce à cette configuration, nous avons mis en place un reverse proxy fonctionnel à l'aide de **HAProxy** sur **pfSense**, permettant de rediriger de manière sécurisée les connexions entrantes vers un service web hébergé sur une machine Ubuntu. Cette redirection a été testée avec succès en environnement NAT, démontrant que le flux réseau transite bien par HAProxy. La désactivation de HAProxy a permis de confirmer son rôle clé dans cette redirection. Ce système offre une couche supplémentaire de contrôle et de sécurité, essentielle dans une architecture réseau professionnelle.