

01/07/2024

# Reverse Proxy Traefik

Installation et configuration de la  
solutions installé directement sur  
un Debian 12.06



Antonin POMIES

## Table des matières

<b>Contexte :</b> .....	2
<b>Prérequis :</b> .....	2
<b>Règles de NAT/PAT</b> .....	2
<b>Installation de Traefik (Méthode Binaire)</b> .....	3
<b>Sécurisation de Traefik (facultatif)</b> .....	6
<b>Prérequis :</b> .....	6

## Contexte :

Dans mon Homelab, j'ai l'habitude de faire passer mes services par un reverse proxy construit grâce à Apache, qui est une solution que je connais bien. J'ai tout de même un peu de mal avec la gestion des certificats, qui est un peu laborieuse, notamment lors de l'utilisation de certificats Let's Encrypt dans le cas du renouvellement (même si CertBot existe). C'est pourquoi j'ai souhaité apprendre une nouvelle façon de faire avec Traefik, qui me permettra d'avoir des configurations plus organisées et une gestion des certificats plus simple.

## Prérequis :

- Une machine sur Debian (12 de préférence pour être sûr que ça fonctionne).
- Quelques connaissances sur le Reverse Proxy en général.
- Une connexion à internet.
- Des services disponibles pour faire les tests (Guacamole / Home Assistant / Site Web).
- Un nom de domaine (Connaissance basique des enregistrements A / CNAME)

## Règles de NAT/PAT





Pour commencer il est impératif d'avoir fait les enregistrements A ou CNAME vers votre IP publique. Dans mon cas :

Un dynhost vers mon ip publique (A)

Un sous domaine pour guacamole (CNAME > dynhost)

Un sous domaine pour Home Assistant (CNAME > dynhost)

Sur mon routeur (Fortigate), j'ai créé deux VIP qui redirige le port 80 et 443 vers mon serveur Traefik. Dans tout les cas le port 80 est redirigé vers le 443

 HTTP to RPX	 WAN1 (wan1)	192.168.1.254 (TCP: 80)	192.168.6.251 (TCP: 80)	988	1
 HTTPS to RPX	 WAN1 (wan1)	192.168.1.254 (TCP: 443)	192.168.6.251 (TCP: 443)	5 834	1

# Installation de Traefik (Méthode Binaire)

On commence évidemment par la mise à jour des dépôts de Debian.

```
sudo apt update && sudo apt upgrade && sudo apt dist-upgrade
```

On télécharge le binaire « Traefik » sur le GitHub officiel.

```
wget https://github.com/traefik/traefik/releases/download/v3.1.0-rc2/traefik_v3.1.0-rc2_linux_amd64.tar.gz
```

On décompresse ensuite l'archive tar.

```
tar -xzf traefik_vX.X.X_linux_amd64.tar.gz
```

Puis on déplace le binaire dans le /bin.

```
sudo mv traefik /usr/local/bin/
```

On modifie les permissions pour le rendre exécutable.

```
chmod 755 /usr/local/bin/traefik
```

Création du fichier principal qui sera appelé lors du démarrage de Traefik, mais avant nous allons créer le dossier où se trouve la configuration de Traefik.

```
mkdir /etc/traefik
```

Puis on crée le fichier Traefik.

```
nano /etc/traefik/traefik.toml
```

Et on ajoute le contenu suivant.

```
[accesslog]
[api]
  insecure=true
  dashboard=true
  debug=true
[log]
  level="INFO"
[entryPoints]
[entryPoints.web]
  address=":80"
```

On va ensuite faire le premier lancement de Traefik.

```
sudo traefik --configFile=/etc/traefik/traefik.toml
```

Evidemment nous ajouterons de la sécurité plus loin dans cette procédure.

Nous allons maintenant créer un processus systemd pour pouvoir utiliser la commande systemctl. Pour ce faire on crée un fichier de service.

```
sudo nano /etc/systemd/system/traefik.service
```

On le remplit de cette manière.

```
[Unit]
Description=Traefik
Documentation=https://docs.traefik.io
After=network-online.target

[Service]
Type=simple
User=root
ExecStart=/usr/local/bin/traefik --configFile=/etc/traefik/traefik.toml
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Je ne vais pas détailler le contenu de ce fichier. Sachez juste qu'il faut modifier le fichier toml avec votre chemin.

Recharger les fichiers de configurations de systemd.

```
sudo systemctl daemon-reload
```

Une fois cela fait on peut démarrer le service et l'activer au démarrage.

```
sudo systemctl start traefik
sudo systemctl enable traefik
```

On peut dès maintenant accéder à l'interface web de Traefik à l'adresse suivante :

<http://127.0.0.1:8080>

Pour compléter le fichier traefik.toml, nous allons rajouter des lignes pour commencer la configuration de notre environnement.

```
[entryPoints]
  [entryPoints.web]
    address = ":80"
  [entryPoints.websecure]
    address = ":443"

[providers]
  [providers.file]
    directory = "/etc/traefik/dynamic"
    watch = true

[api]
  dashboard = true
  insecure = true

[certificatesResolvers.letsencrypt]
  [certificatesResolvers.letsencrypt.acme]
    email = "votre_email"
    storage = "/etc/traefik/acme.json"
  [certificatesResolvers.letsencrypt.acme.httpChallenge]
    entryPoint = "web"
```

Comme vu dans le fichier, il est question d'un dossier provider qui est en fait le dossier où les configurations seront présentes on va donc créer le dossier.

```
mkdir /etc/traefik/dynamic
```

Puis on crée le fichier config.toml.

```
nano /etc/traefik/dynamic/config.toml
```

Dans ce fichier, il va y avoir ce qui peut être comparable au vhosts d'apache et nginx. Evidemment présenté différemment car en format .toml.

Dans mon cas la configuration sera la suivante :

```
[http]
[http.routers]
[http.routers.bastion-http-to-https]
  rule = "Host(`sous domaine`)"
  entryPoints = ["web"]
  middlewares = ["redirect-to-https"]
  service = "noop"

[http.routers.homeassistant-http-to-https]
  rule = "Host(`sous domaine`)"
  entryPoints = ["web"]
  middlewares = ["redirect-to-https"]
  service = "noop"

[http.routers.to-bastion]
  rule = "Host(`sous domaine`)"
  entryPoints = ["websecure"]
  service = "bastion-service"
[http.routers.to-bastion.tls]
  certResolver = "letsencrypt"

[http.routers.to-homeassistant]
  rule = "Host(`sous domaine`)"
  entryPoints = ["websecure"]
  service = "homeassistant-service"
[http.routers.to-homeassistant.tls]
  certResolver = "letsencrypt"

[http.middlewares]
[http.middlewares.redirect-to-https.redirectScheme]
  scheme = "https"
  permanent = true

[http.services]
[http.services.bastion-service.loadBalancer]
  [[http.services.bastion-service.loadBalancer.servers]]
    url = "http://192.168.6.253:8080"

[http.services.homeassistant-service.loadBalancer]
  [[http.services.homeassistant-service.loadBalancer.servers]]
    url = "http://192.168.6.252:8123"
```

Nous allons maintenant nous occuper de la partie certificat let's encrypt.

```
touch /etc/traefik/acme.json
```

Puis on donne les droits nécessaires et suffisant (600 en général).

```
chmod 600 /etc/traefik/acme.json
```

On redémarre le service pour finir.

```
sudo systemctl restart traefik
```

Pour cette configuration j'ai mis un bastion guacamole et un homeassistant derrière traefik et tout fonctionne bien.

# Sécurisation de Traefik (facultatif)

## Prérequis :

Installation de Apache Utils.

```
sudo apt-get install apache2-utils
```

On génère le hash d'un mot de passe pour l'authentification.

```
htpasswd -nB debian motdepasse
```

Ça génère un résultat similaire à ça.

```
debian:$apr1$M3yRZieu$HN/YDf7/hCMeQDu6czeMb/
```

Une fois cela fait, nous allons commencer les modifications dans les fichiers de configurations :

Pour commencer, on commente la ligne suivante dans le fichier traefik.toml.

```
#insecure = true
```

On va ensuite ajouter un entry point api qui sera en écoute sur le port 8080.

```
[entryPoints.api]
  address = ":8080"
```

Pour la configuration de config.toml, nous allons ajouter deux bloc un middleware pour déclarer l'authentification et un router pour définir l'accès à l'interface et ajouter l'authentification.

Le middleware avec le hash généré auparavant.

```
[http.middlewares.auth.basicAuth]
  users = [
    "debian:$apr1$M3yRZieu$HN/YDf7/hCMeQDu6czeMb/" # Mot de Passe de connexion
  ]
```

Et le router avec les paramétrages nécessaires.

```
[http.routers.api]
  service = "api@internal" #pour le service a protégé dans notre cas l'api
  rule = "PathPrefix(`/api`)||PathPrefix(`/dashboard`)" #les deux préfixe à protéger
  entryPoints = ["api"] #l'entrypoint défini plus tôt
  middlewares = ["auth"] #le middlewares qui configure l'authentification
```

On peut donc ensuite redémarrer le service Traefik.

```
sudo systemctl restart traefik
```

Normalement lors de votre reconnexion au Dashboard, une fenêtre de connexion devrait apparaître. L'authentification n'est pas super sophistiquée mais le Traefik ne sera pas exposé sur internet.

# Fin