Installation et configuration

Reverse Proxy Traefik

Installation et configuration sur une base Debian (Debian 13 / Ubuntu 24.04 etc...)

Auteur : Antonin POMIES
Dernière mise à jour : Janvier 2025

Contexte

Dans mon Homelab, j'utilisais habituellement Apache comme reverse proxy, une solution que je maîtrise bien. Cependant, la gestion des certificats s'avérait laborieuse, notamment pour le renouvellement des certificats Let's Encrypt, même avec CertBot.

C'est pourquoi j'ai décidé d'explorer Traefik, une solution moderne qui offre :

- Des configurations plus organisées et modulaires
- Une gestion automatique des certificats SSL/TLS
- Une découverte automatique des services (Docker)
- Un tableau de bord intégré pour le monitoring

Prérequis

Avant de commencer l'installation, assurez-vous d'avoir :

- Une machine sous **Debian 12** (ou equivalent)
- Des connaissances de base sur les **reverse proxy**
- Une connexion à **Internet**
- Des **services disponibles** pour les tests (Teleport, Home Assistant, site web, etc.)
- Un **nom de domaine** avec accès à la gestion DNS (enregistrements A / CNAME)
- Un accès administrateur (**sudo**) sur le serveur

Configuration DNS et NAT/PAT

Configuration DNS

Pour que Traefik puisse router correctement le trafic, vous devez configurer vos enregistrements DNS :

1. **Enregistrement A** : Pointez votre domaine principal vers votre IP publique
   - Exemple : `exemple.com` → `203.0.113.42`
   - Ou utilisez un service **DynDNS** si votre IP est dynamique
   - Pour les tests vous pouvez également faire du DNS menteur avec le fichier /etc/hosts ou un DNS local.
2. **Enregistrements CNAME** : Créez des sous-domaines pour vos services
   - `guacamole.exemple.com` → `exemple.com`
   - `homeassistant.exemple.com` → `exemple.com`

ℹ Vous pouvez également faire un enregistrement "Wildcard" de type "*.example.com"

Configuration du routeur (NAT/PAT)

Sur votre routeur/firewall, configurez les règles de redirection de ports :

Protocole	Port externe	Port interne	IP destination	Description
TCP	80	80	192.168.X.X	HTTP (redirigé vers HTTPS)
TCP	443	443	192.168.X.X	HTTPS

Note importante : Le port 80 doit être accessible pour permettre la validation des certificats Let's Encrypt via le challenge HTTP-01, mais tout le trafic sera automatiquement redirigé vers le port 443 (HTTPS).

Installation de Traefik (méthode binaire)

Étape 1 : Mise à jour du système

Commencez par mettre à jour votre système Debian :

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

Étape 2 : Téléchargement de Traefik

Téléchargez la dernière version stable de Traefik depuis le dépôt GitHub officiel. La version actuelle recommandée est la v3.6.7 :

```
cd /tmp
wget https://github.com/traefik/traefik/releases/download/v3.6.7/traefik_v3.6.7_linux_amd64.tar.gz
```

Note : Vérifiez toujours la page des releases officielle pour obtenir la dernière version stable.

Étape 3 : Extraction et installation

Décompressez l'archive téléchargée :

```
tar -xzf traefik_v3.6.7_linux_amd64.tar.gz
```

Déplacez le binaire dans le répertoire système :

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

Rendez le binaire exécutable :

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

Vérifiez l'installation :

```
traefik version
```

Étape 4 : Création de la structure des répertoires

Créez les répertoires nécessaires pour la configuration de Traefik :

```
sudo mkdir -p /etc/traefik/dynamic
sudo mkdir -p /var/log/traefik
```

Étape 5 : Configuration principale de Traefik

Créez le fichier de configuration principal :

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

Ajoutez la configuration suivante :

```
# Configuration globale
[global]
  checkNewVersion = true
  sendAnonymousUsage = false

# Logs
[log]
  level = "INFO"
  filePath = "/var/log/traefik/traefik.log"

# Logs d'accès
[accessLog]
  filePath = "/var/log/traefik/access.log"

# Points d'entrée (ports d'écoute)
[entryPoints]
  [entryPoints.web]
    address = ":80"
    # Redirection automatique HTTP vers HTTPS
    [entryPoints.web.http]
      [entryPoints.web.http.redirections]
        [entryPoints.web.http.redirections.entryPoint]
          to = "websecure"
          scheme = "https"
```

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

  [entryPoints.api]
    address = ":8080"

# API et Dashboard
[api]
  dashboard = true
  # insecure = true # À commenter en production (voir section sécurisation)

# Providers - Configuration dynamique via fichiers
[providers]
  [providers.file]
    directory = "/etc/traefik/dynamic"
    watch = true

# Gestion automatique des certificats Let's Encrypt
[certificatesResolvers.letsencrypt]
  [certificatesResolvers.letsencrypt.acme]
    email = "votre-email@exemple.com"  # ⚠ Remplacez par votre email
    storage = "/etc/traefik/acme.json"
    [certificatesResolvers.letsencrypt.acme.httpChallenge]
      entryPoint = "web"
```

Points importants :

- Remplacez `votre-email@exemple.com` par votre adresse email réelle
- L'option `insecure = true` dans la section API permet un accès sans authentification au dashboard. Commentez-la en production.

Étape 6 : Création du fichier de stockage des certificats

Créez le fichier qui stockera les certificats Let's Encrypt :

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

Important : Les permissions `600` sont obligatoires pour des raisons de sécurité. Ci ce n'est pas fait alors il y aura des erreurs de certificats.

Étape 7 : Configuration dynamique des routes

Créez le fichier de configuration dynamique pour vos services :

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

Ajoutez une configuration exemple (à adapter selon vos services) :

```
# Configuration HTTP
[http]
  # Middlewares
  [http.middlewares]
    # Middleware de redirection HTTP vers HTTPS
    [http.middlewares.redirect-to-https.redirectScheme]
      scheme = "https"
      permanent = true

  # Routeurs
  [http.routers]
    # Router pour Guacamole (HTTPS)
    [http.routers.to-guacamole]
      rule = "Host(`guacamole.votre-domaine.com`)"
      entryPoints = ["websecure"]
      service = "guacamole-service"
      [http.routers.to-guacamole.tls]
        certResolver = "letsencrypt"

    # Router pour Home Assistant (HTTPS)
    [http.routers.to-homeassistant]
      rule = "Host(`homeassistant.votre-domaine.com`)"
      entryPoints = ["websecure"]
      service = "homeassistant-service"
      [http.routers.to-homeassistant.tls]
        certResolver = "letsencrypt"
```

```
  # Services (backends)
  [http.services]
    # Service Guacamole
    [http.services.guacamole-service.loadBalancer]
      [[http.services.guacamole-service.loadBalancer.servers]]
        url = "http://192.168.X.X:8080"  # ⚠ Remplacez par l'IP de votre service

    # Service Home Assistant
    [http.services.homeassistant-service.loadBalancer]
      [[http.services.homeassistant-service.loadBalancer.servers]]
        url = "http://192.168.X.X:8123"  # ⚠ Remplacez par l'IP de votre service
```

À personnaliser :

- Remplacez `votre-domaine.com` par votre domaine réel
- Adaptez les adresses IP et ports selon vos services
- Ajoutez autant de routeurs et services que nécessaire

Étape 8 : Création du service systemd

Pour que Traefik démarre automatiquement au boot, créez un service systemd :

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

Ajoutez le contenu suivant :

```
[Unit]
Description=Traefik Reverse Proxy
Documentation=https://doc.traefik.io/traefik/
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
User=root
ExecStart=/usr/local/bin/traefik --configFile=/etc/traefik/traefik.toml
Restart=on-failure
RestartSec=5s

# Sécurité
```

```
NoNewPrivileges=true
PrivateTmp=true

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

Étape 9 : Activation et démarrage de Traefik

Rechargez la configuration systemd :

```
sudo systemctl daemon-reload
```

Démarrez le service Traefik :

```
sudo systemctl start traefik
```

Activez le démarrage automatique au boot :

```
sudo systemctl enable traefik
```

Vérifiez le statut du service :

```
sudo systemctl status traefik
```

Si tout fonctionne correctement, vous devriez voir le service actif et sans erreurs.

Étape 10 : Vérification

Accédez au dashboard Traefik via votre navigateur :

- **URL locale** : `http://127.0.0.1:8080` (depuis le serveur)
- **URL locale** : `http://adresse-ip-serveur:8080` (depuis votre réseau local)

Vous devriez voir :

- L'interface du dashboard Traefik
- Vos routeurs configurés
- Les services backend
- L'état des certificats

Testez l'accès à vos services :

- `https://guacamole.votre-domaine.com`
- `https://homeassistant.votre-domaine.com`

Les certificats Let's Encrypt devraient être générés automatiquement lors de la première connexion.

Sécurisation de Traefik

Par défaut, le dashboard Traefik est accessible sans authentification. Il est fortement recommandé de le sécuriser, surtout si vous l'exposez sur Internet.

Prérequis : Installation d'Apache Utils

Installez le paquet contenant l'utilitaire `htpasswd` :

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

Génération du mot de passe hashé

Générez un hash de mot de passe pour l'authentification :

```
htpasswd -nB utilisateur
```

Remplacez `utilisateur` par le nom d'utilisateur souhaité. Le système vous demandera de saisir le mot de passe deux fois.

Exemple de sortie :

```
utilisateur:$2y$05$sKx9gFdKLgDeOYc3qKMCOe6c7YqJvMKlVKXQzhPkrm3YWsYJgW1Gy
```

Important : Copiez cette ligne complète, vous en aurez besoin pour la configuration.

Configuration de l'authentification

Méthode 1 : Authentification basique (recommandée)

Modifiez le fichier de configuration dynamique :

```
sudo nano /etc/traefik/dynamic/dashboard.toml
```

Ajoutez la configuration suivante :

```
# Middleware d'authentification pour le dashboard
[http.middlewares]
  [http.middlewares.dashboard-auth.basicAuth]
    users = [
      "utilisateur:$2y$05$sKx9gFdKLgDeOYc3qKMCOe6c7YqJvMKlVKXQzhPkrm3YWsYJgW1Gy"
    ]

# Router pour le dashboard sécurisé
[http.routers]
  [http.routers.dashboard]
    rule = "Host(`traefik.votre-domaine.com`)"
    entryPoints = ["websecure"]
    service = "api@internal"
    middlewares = ["dashboard-auth"]
    [http.routers.dashboard.tls]
      certResolver = "letsencrypt"
```

À personnaliser :

- Remplacez la ligne `users` par le hash généré précédemment
- Remplacez `traefik.votre-domaine.com` par votre sous-domaine
- Créez l'enregistrement DNS correspondant

Méthode 2 : Restriction par IP (optionnelle)

Pour restreindre l'accès au dashboard à certaines adresses IP, ajoutez un middleware supplémentaire :

```
# Middleware de restriction IP
[http.middlewares]
  [http.middlewares.ip-whitelist.ipWhiteList]
    sourceRange = [
      "192.168.1.0/24",     # Réseau local
      "203.0.113.42/32"     # IP publique spécifique
    ]
```

```
# Router pour le dashboard avec IP whitelist
[http.routers]
  [http.routers.dashboard]
    rule = "Host(`traefik.votre-domaine.com`)"
    entryPoints = ["websecure"]
    service = "api@internal"
    middlewares = ["dashboard-auth", "ip-whitelist"]  # Combinaison des deux
    [http.routers.dashboard.tls]
      certResolver = "letsencrypt"
```

Désactivation du mode insecure

Une fois l'authentification configurée, modifiez le fichier principal :

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

Commentez ou supprimez la ligne `insecure = true` dans la section `[api]` :

```
[api]
  dashboard = true
  # insecure = true  # ⚠ À commenter ou supprimer
```

Redémarrage de Traefik

Appliquez les modifications :

```
sudo systemctl restart traefik
```

Vérifiez que tout fonctionne :

```
sudo systemctl status traefik
```

Vérification de la sécurisation

Accédez au dashboard via l'URL configurée :

- `https://traefik.votre-domaine.com`

Vous devriez maintenant être invité à saisir vos identifiants. L'accès via `http://adresse-ip:8080` ne devrait plus fonctionner.

Maintenance et surveillance

Consultation des logs

Traefik génère deux types de logs :

Logs applicatifs (erreurs, démarrages, etc.) :

```
sudo tail -f /var/log/traefik/traefik.log
```

Logs d'accès (requêtes HTTP) :

```
sudo tail -f /var/log/traefik/access.log
```

Vérification de l'état du service

```
sudo systemctl status traefik
```

Redémarrage après modification de la configuration

Configuration statique (`traefik.toml`) :

```
sudo systemctl restart traefik
```

Configuration dynamique (`dynamic/*.toml`) :

- Les modifications sont appliquées automatiquement grâce à l'option `watch = true`
- Pas besoin de redémarrage

Mise à jour de Traefik

Pour mettre à jour Traefik vers une nouvelle version :

1. Téléchargez la nouvelle version :

```
cd /tmp
wget https://github.com/traefik/traefik/releases/download/vX.Y.Z/traefik_vX.Y.Z_linux_amd64.tar.gz
tar -xzf traefik_vX.Y.Z_linux_amd64.tar.gz
```

2. Arrêtez le service :

```
sudo systemctl stop traefik
```

3. Remplacez le binaire :

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

4. Redémarrez le service :

```
sudo systemctl start traefik
```

5. Vérifiez la version :

```
traefik version
```

Sauvegarde de la configuration

Il est recommandé de sauvegarder régulièrement :

- `/etc/traefik/` (configurations)
- `/etc/traefik/acme.json` (certificats)

```
sudo tar -czf traefik-backup-$(date +%Y%m%d).tar.gz /etc/traefik/
```

Résolution des problèmes courants

Le dashboard n'est pas accessible

Vérifications :

1. Le service Traefik est-il démarré ?

```
sudo systemctl status traefik
```

2. Le port 8080 est-il en écoute ?

```
sudo ss -tlnp | grep 8080
```

3. Le pare-feu autorise-t-il le port 8080 ?

```
sudo ufw status
```

Les certificats Let's Encrypt ne se génèrent pas

Causes possibles :

1. Le port 80 n'est pas accessible depuis Internet
2. Les enregistrements DNS ne pointent pas vers la bonne IP
3. Le fichier `acme.json` n'a pas les bonnes permissions

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

4. L'email dans `traefik.toml` n'est pas valide

Vérification :

```
sudo tail -f /var/log/traefik/traefik.log
```

Le service backend n'est pas accessible

Vérifications :

1. Le service backend est-il démarré ?
2. L'adresse IP et le port sont-ils corrects dans `config.toml` ?
3. Traefik peut-il atteindre le backend ?

```
curl http://192.168.X.X:port
```

4. Vérifiez les logs pour identifier l'erreur :

```
sudo journalctl -u traefik -f
```

Exemple de configuration avancée

Ajout d'un nouveau service

Pour ajouter un nouveau service (par exemple, un serveur web) :

1. Créez ou modifiez `/etc/traefik/dynamic/config.toml` :

```
# Router pour le nouveau service
[http.routers.to-webapp]
  rule = "Host(`webapp.votre-domaine.com`)"
  entryPoints = ["websecure"]
  service = "webapp-service"
  [http.routers.to-webapp.tls]
    certResolver = "letsencrypt"

# Service backend
[http.services.webapp-service.loadBalancer]
  [[http.services.webapp-service.loadBalancer.servers]]
    url = "http://192.168.X.X:80"
```

2. Créez l'enregistrement DNS : `webapp.votre-domaine.com`
3. Traefik détectera automatiquement la nouvelle configuration

Middlewares utiles

Compression Gzip

```
[http.middlewares.gzip-compress.compress]
```

Headers de sécurité

```
[http.middlewares.security-headers.headers]
  [http.middlewares.security-headers.headers.customResponseHeaders]
    X-Frame-Options = "SAMEORIGIN"
    X-Content-Type-Options = "nosniff"
    X-XSS-Protection = "1; mode=block"
    Referrer-Policy = "strict-origin-when-cross-origin"
```

Rate limiting

```
[http.middlewares.rate-limit.rateLimit]
  average = 100
  burst = 200
```

Appliquez ensuite ces middlewares à vos routeurs :

```
[http.routers.to-webapp]
  rule = "Host(`webapp.votre-domaine.com`)"
  entryPoints = ["websecure"]
  service = "webapp-service"
  middlewares = ["gzip-compress", "security-headers", "rate-limit"]
  [http.routers.to-webapp.tls]
    certResolver = "letsencrypt"
```

Ressources complémentaires

- **Documentation officielle** : https://doc.traefik.io/traefik/
- **GitHub officiel** : https://github.com/traefik/traefik
- **Forum communautaire** : https://community.traefik.io/
- **Exemples de configuration** : https://github.com/traefik/traefik/tree/master/examples

---

Conclusion

Traefik est désormais installé et configuré sur votre serveur Debian. Les avantages principaux de cette solution sont :

- **Automatisation** : Génération et renouvellement automatiques des certificats SSL
- **Simplicité** : Configuration en fichiers TOML facilement lisibles
- **Modularité** : Ajout de services sans redémarrage
- **Monitoring** : Dashboard intégré pour surveiller le trafic
- **Performance** : Proxy moderne et optimisé

Cette configuration de base peut être étendue avec de nombreuses fonctionnalités avancées selon vos besoins (load balancing, service mesh, intégration Docker/Kubernetes, etc.).

Document créé par Antonin POMIES - Mis à jour en janvier 2025

Revision #1
Created 30 January 2026 17:43:06 by Antonin POMIES
Updated 30 January 2026 17:53:31 by Antonin POMIES