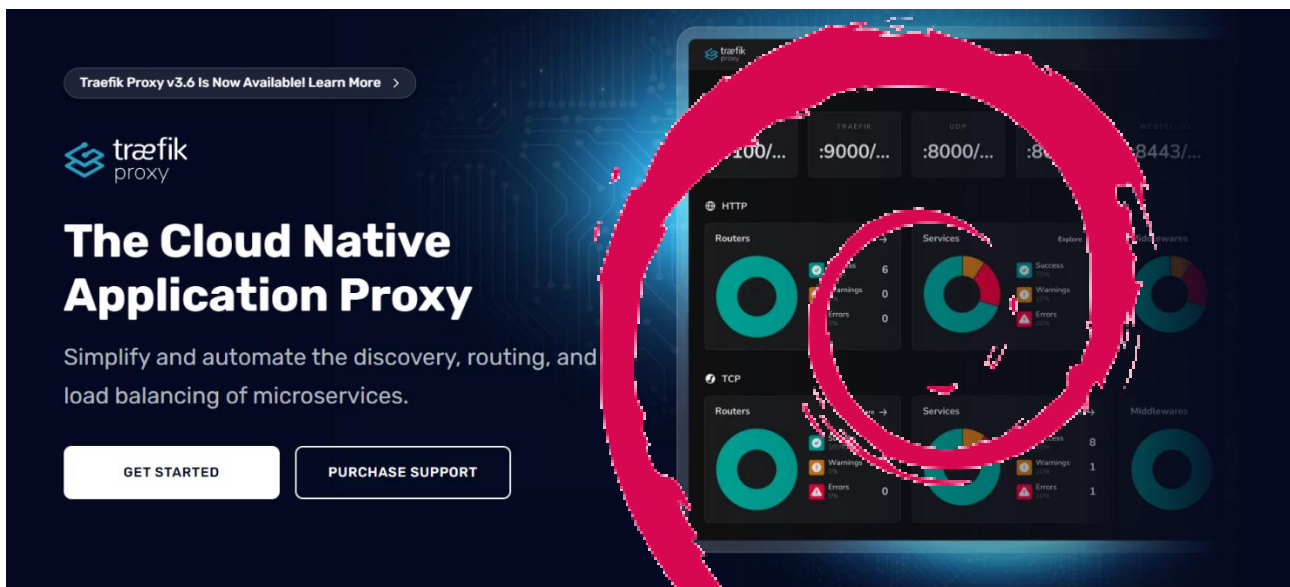


GUIDE DE MISE EN ŒUVRE TRAEFIK 3.6.5

Le reverse proxy moderne open source

Environnement de travail : Debian 13.2



debian

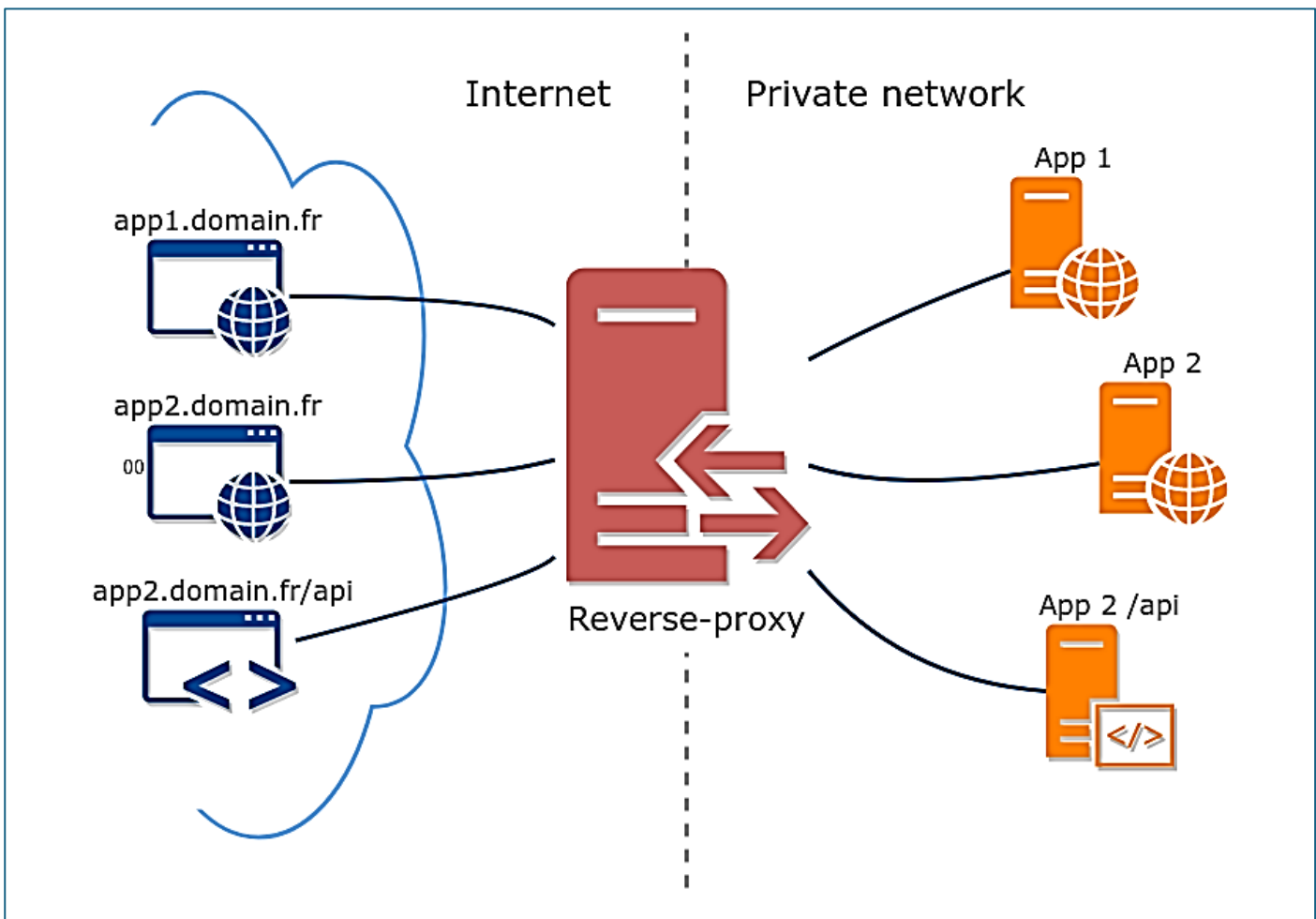
BTS SIO - Option SISR
Administration des systèmes et réseaux

1. Qu'est-ce que Traefik ?

Traefik est un outil écrit en Golang par Emile VAUGE et c'est français ! Traefik est un **reverse-proxy**.

Un reverse-proxy est une brique de votre infrastructure, qui permet d'être un intermédiaire de communication, entre un réseau public et un réseau privé, le nôtre par exemple. C'est sur ce réseau privé que l'on trouvera toutes les applications de notre système informatique qui ne sont pas accessibles depuis l'extérieur pour des raisons de sécurité principalement.

Le reverse-proxy doit donc connaître toutes les « routes » disponibles pour **transférer chaque requête vers le bon service**. Comme il est le point central du trafic, il peut également faire du « load balancing », ou encore appliquer des règles de sécurité comme le HTTPS (via Let's Encrypt par exemple).

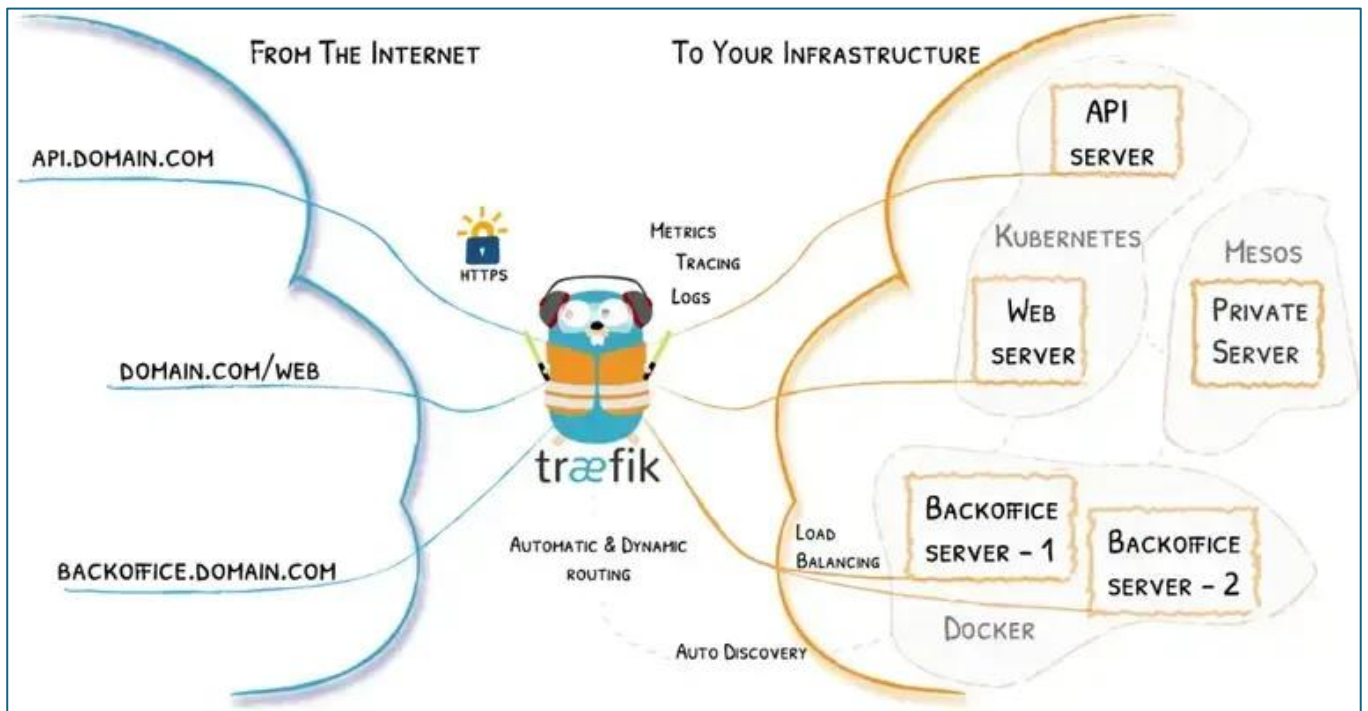


La véritable **force de Traefik** c'est que **sa configuration est dynamique** : il va se connecter au socket Docker et récupérer toutes les informations sur les containers qui tournent en temps réel de façon à pouvoir router les requêtes sur ces derniers de manière automatique ! Traefik supporte plusieurs algorithmes de répartition des charges et gère :

- le HTTPS/TLS et la génération des certificats via Let's encrypt (plus besoin de Certbot)
- le HTTP et le HTTP/2 mais aussi Websocket, GRPC, TCP et UDP
- une interface web pour surveiller l'état des services
- une exposition de métriques vers Prometheus et la rétention des logs d'accès

À ce jour (janvier 2026), la **dernière version en date stable** de Traefik est la **3.6**

Traefik est conçu pour les applications conteneurisées et les architectures microservices. En tant que tel, il joue un rôle important dans le routage des requêtes des clients vers les services appropriés au sein d'une infrastructure.

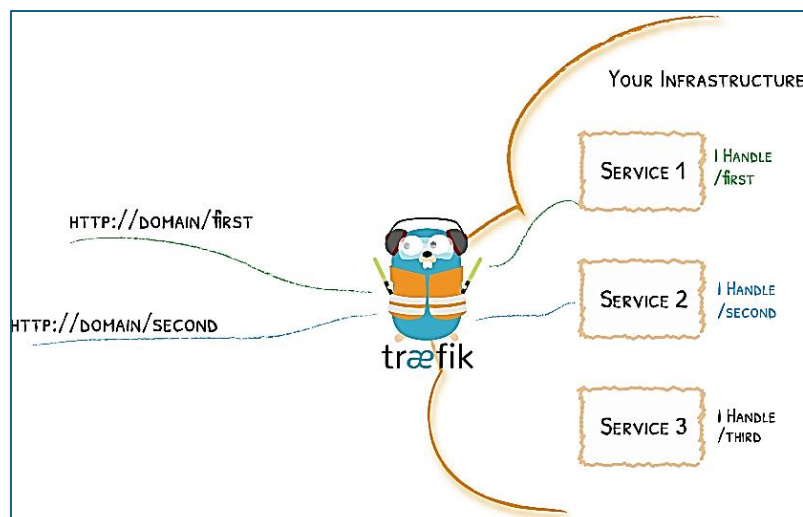


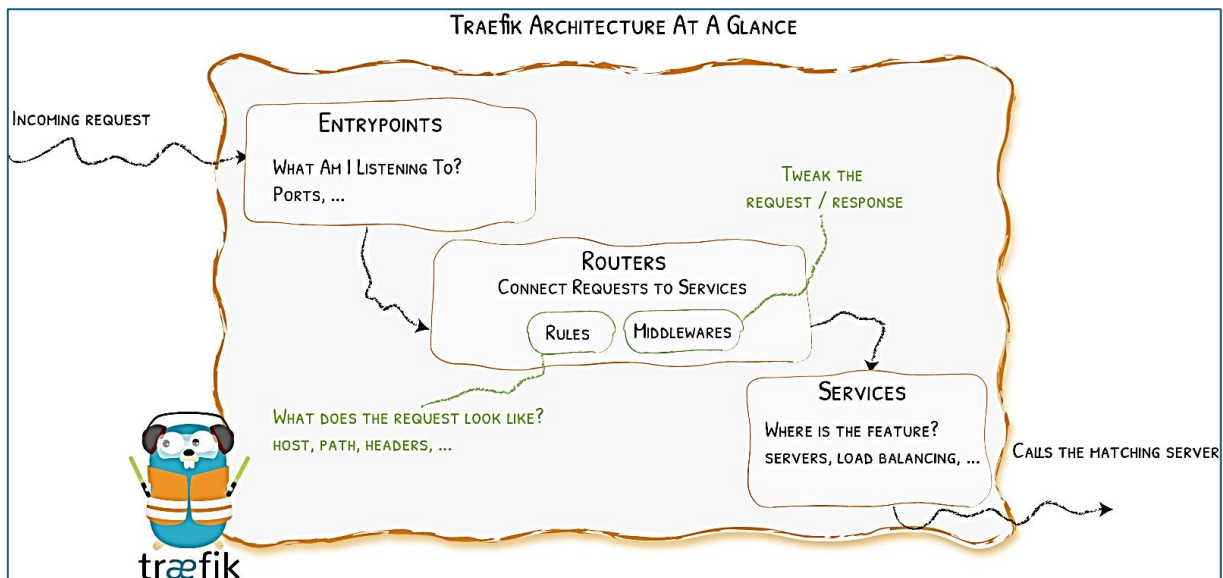
Lorsqu'un utilisateur envoie une requête HTTP à Traefik, celle-ci contient toutes les informations nécessaires pour indiquer à Traefik sa destination. En général, on utilise le nom de domaine et le chemin de la requête pour déterminer sa destination.

Traefik analyse la requête et cherche dans sa liste de « routes » si l'une d'elles peut correspondre. Si c'est le cas **il va transférer la requête vers son « service » associé**. Dans le cas contraire il va retourner à l'utilisateur une « erreur 404 ».

Pour fonctionner, Traefik utilise 2 types de configurations qui sont importantes à connaître :

- la configuration dite **STATIQUE** (qui nécessitera un redémarrage de l'outil) ;
- la configuration dite **DYNAMIQUE** (qui permettra d'appliquer des modifications « à la volée »).





Les **ENTRYPPOINTS** sont les **points d'entrées**. Il s'agit des adresses et ports exposés par Traefik qui permettront d'exposer nos applications. La plupart du temps, ce sont les **ports 443 et 80** qui sont exposés par défaut.

Les **ROUTERS** gèrent les **règles de redirection** du reverse. Un routeur pointe un service. On affecte au routeur des URL, des entrypoints.

Les **SERVICES** sont les **applications**, par exemple Nextcloud. Nous pouvons recenser 3 types de services : HTTP, TCP et UDP.

Les **PROVIDERS** sont les **sources de génération de la configuration**. C'est ce qui fait la force de Traefik. Il existe plusieurs types de « **providers** » parmi lesquels :

- Moteur de conteneur :
 - **Docker** : Traefik obtient l'accès au socket de docker et pourra gérer les services (conteneurs).
 - **Kubernetes**
 - **Rancher**
 - **Marathon**
- Base de clé/valeur :
 - **consul**
 - **consulcatalog**
 - **etcd**
 - **redis**
 - **zookeeper**
- **file** (ce provider permet d'indiquer un chemin vers un fichier de configuration par exemple)

Dans ce guide, nous nous contenterons d'étudier les providers « Docker » et « file ».

Les **MIDDLEWARES** sont des **étapes intermédiaires entre le router et le service**. Le « **middleware** » peut être utilisé pour faire de la **compression**, de la **sécurisation**, de l'**authentification** à une page comme celui qu'on utilisera ici pour l'accès au tableau de bord de Traefik.

2. Installation de Traefik 3.6.5 en bare métal sur Debian 13.2

Le guide ci-dessous va vous permettre d'installer le reverse proxy Traefik en bare métal sur votre machine Debian 13 externalisée (**nous ne l'installerons pas avec Docker Compose pour bien comprendre le fonctionnement de ce reverse proxy moderne**).

- Connectez-vous, en SSH, à votre machine « **traefik.teamX.sio-ndlp.fr** » en vous servant des identifiants fournis. **Faites précéder les commandes fournies de « sudo » !**

1^{ère} étape : préparation de l'environnement Traefik (sur la machine Debian Traefik)

Commandes à saisir (une ligne par commande) :

```
sudo apt update && sudo apt upgrade -y
sudo apt install curl -y
sudo mkdir -p /etc/traefik /var/lib/traefik
sudo chown traefik:traefik /etc/traefik /var/lib/traefik
```

Téléchargement des sources officielles (**attention la commande « wget » tient sur une seule ligne**) :

```
sudo wget
https://github.com/traefik/traefik/releases/download/v3.6.5/traefik_v3.6.5_linux_amd64.tar.gz

sudo tar -xvzf traefik_v3.6.5_linux_amd64.tar.gz -C /usr/local/bin traefik

sudo chmod +x /usr/local/bin/traefik
```

2^{ème} étape : préparation du fichier de configuration STATIQUE

```
sudo nano /etc/traefik/traefik.yml
```

Contenu à copier en respectant l'indentation ; les valeurs en **rouge** sont à modifier :

```
log:
  level: INFO

api:
  dashboard: true
  insecure: false

entryPoints:
  web:
    address: ":80"
  http:
    redirections:
      entryPoint:
        to: websecure
        scheme: https
    permanent: true

websecure:
  address: ":443"
  http3:
    advertisedPort: 443 # ← Important : annonce le port QUIC aux clients (http/3)
```

```
http:
  tls:
    certResolver: le    # ← Assure que Let's Encrypt (le) est bien configuré
```

```
# Charge la config dynamique
providers:
  file:
    directory: "/etc/traefik/dynamic"
    watch: true
```

```
certificatesResolvers:
  le:
    acme:
      email: "votremail@votre domaine.mail"
      storage: "/var/lib/traefik/acme.json"
      httpChallenge:
        entryPoint: web
```

3^{ème} étape : création du fichier « acme.json » pour les certificats Let's Encrypt

Commandes à saisir (une par ligne) ; les valeurs en rouge sont à modifier :

```
sudo touch /var/lib/traefik/acme.json
sudo chown traefik:traefik /var/lib/traefik/acme.json
sudo chmod 600 /var/lib/traefik/acme.json
```

4^{ème} étape : création du mot de passe chiffré pour le tableau de bord Traefik ; les valeurs en rouge sont à modifier

```
sudo apt install apache2-utils -y
sudo htpasswd -nbB admin mon_mot_de_passe
```

Remplacez « mon_mot_de_passe » par le mot de passe souhaité pour l'authentification au tableau de bord de Traefik.

Copiez le mot de passe chiffré obtenu car il servira plus tard.

5^{ème} étape : création du tableau de bord Traefik

```
sudo mkdir /etc/traefik/dynamic
sudo nano /etc/traefik/dynamic/dashboard.yml
```

Contenu à coller (en respectant l'indentation) ; les valeurs en rouge sont à modifier

```
http:
  routers:
    dashboard:
      rule: "Host(`traefik.teamX.sio-ndlp.fr`) && (PathPrefix(`/api`) || PathPrefix(`/dashboard`))"
      entryPoints: ["websecure"]
      service: api@internal
      tls:
        certResolver: le
      middlewares:
        - auth
        - SecureHeaders
        - Compression
```

```

dashboard-root:
  rule: "Host(`traefik.teamX.sio-ndlp.fr`) && Path(`/`)"
  entryPoints: ["websecure"]
  service: api@internal
  tls:
    certResolver: le
  middlewares:
    - dash-redirect
    - SecureHeaders

```

```

middlewares:
  auth:
    basicAuth:
      realm: "Traefik"
      users:
        - "admin:coller_ici_le_mot_de_passe_chiffré"

```

```

dash-redirect:
  redirectRegex:
    regex: "^/$"
    remplacement: "/dashboard/"
  permanent: true

```

6^{ème} étape : création du fichier de configuration dynamique TLS

sudo nano /etc/traefik/dynamic/tls.yml

Contenu à coller (en respectant l'indentation) :

```

# Configuration TLS
tls:
  options:
    default:
      minVersion: "VersionTLS12"
      sniStrict: true
      cipherSuites:
        - "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256"
        - "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384"
        - "TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305"
        - "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256"
        - "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384"
        - "TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305"
      curvePreferences:
        - CurveP256
        - CurveP384
        - CurveP521

```

7^{ème} étape : configuration des « headers »**sudo nano /etc/traefik/dynamic/headers.yml**

Contenu à copier (en respectant l'indentation) :

```

http:
  middlewares:
    Compression:
      compress:
        excludedContentTypes:
          - text/event-stream

SecureHeaders:
  headers:
    # HTTPS/HSTS
    forceSTSHeader: true
    stsSeconds: 63072000
    stsIncludeSubdomains: true
    stsPreload: true

    # X-Frame-Options (garde SAMEORIGIN)
    customFrameOptionsValue: "SAMEORIGIN"

    # Divers sûreté
    contentTypeNosniff: true
    referrerPolicy: "same-origin"

    # Permissions-Policy
    permissionsPolicy: >-
      accelerometer=(), camera=(), geolocation=(),
      gyroscope=(), magnetometer=(), microphone=(),
      payment=(), usb=()

    # CSP moyennement durcie avec compatibilité Dashboard Traefik et Cockpit
    # - autorise inline JS/CSS + eval (Cockpit en a besoin)
    # - autorise WebSockets (wss:) pour /api & Cockpit
    # - garde les garde-fous (frame-ancestors/base-uri/form-action)
    contentSecurityPolicy: >-
      default-src 'self';
      script-src 'self' 'unsafe-inline' 'unsafe-eval';
      style-src 'self' 'unsafe-inline';
      img-src 'self' data: blob;;
      font-src 'self' data;;
      connect-src 'self' https: wss;;
      frame-ancestors 'self';
      base-uri 'self';
      form-action 'self';
      upgrade-insecure-requests

    # CORS (laisse désactivé si inutile)
    # accessControlAllowOriginList:
    # - "https://app.exemple.com"
    # accessControlAllowMethods:

```

```
# - "GET"
# - "POST"
# - "OPTIONS"
# accessControlAllowHeaders:
# - "Authorization"
# - "Content-Type"
# accessControlMaxAge: 600
# addVaryHeader: true
```

8^{ème} étape : création et activation du service Traefik

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

Contenu à copier (en respectant l'indentation) :

```
[Unit]
Description=Traefik Reverse Proxy
After=network-online.target
Wants=network-online.target

[Service]
User=traefik
Group=traefik
ExecStart=/usr/local/bin/traefik --configfile=/etc/traefik/traefik.yml
WorkingDirectory=/etc/traefik
Restart=on-failure
AmbientCapabilities=CAP_NET_BIND_SERVICE
CapabilityBoundingSet=CAP_NET_BIND_SERVICE
NoNewPrivileges=true
LimitNOFILE=1048576
ProtectSystem=full
ProtectHome=true
PrivateTmp=true
ReadWritePaths=/var/lib/traefik /etc/traefik

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

sudo systemctl daemon-reload

sudo systemctl enable --now traefik

sudo systemctl status traefik

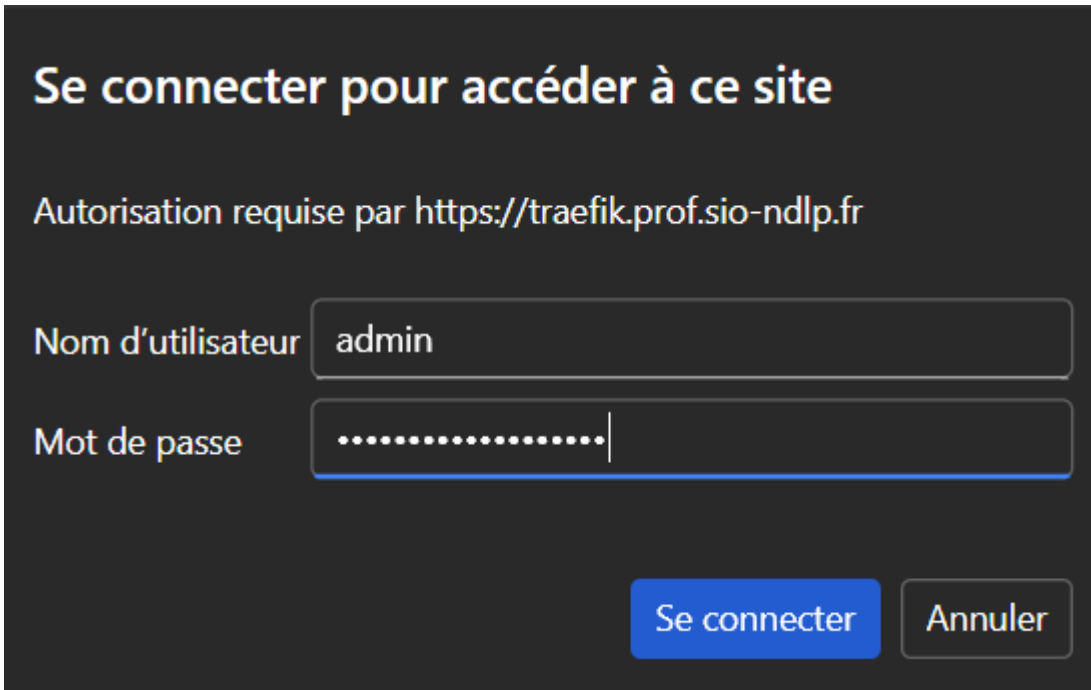
Si le statut est en mode « running », pressez la touche « Q » : Traefik est fonctionnel. Vous pouvez vous connecter au tableau de bord Traefik, depuis un navigateur, en saisissant l'URL suivante :

<https://traefik/teamX.sio-ndlp.fr>

Une fenêtre d'authentification doit s'afficher :

- Saisissez « admin »
- Saisissez le mot de passe tel que vous l'avez défini lors de l'installation

Fenêtre d'authentification au tableau de bord Traefik :



Se connecter pour accéder à ce site

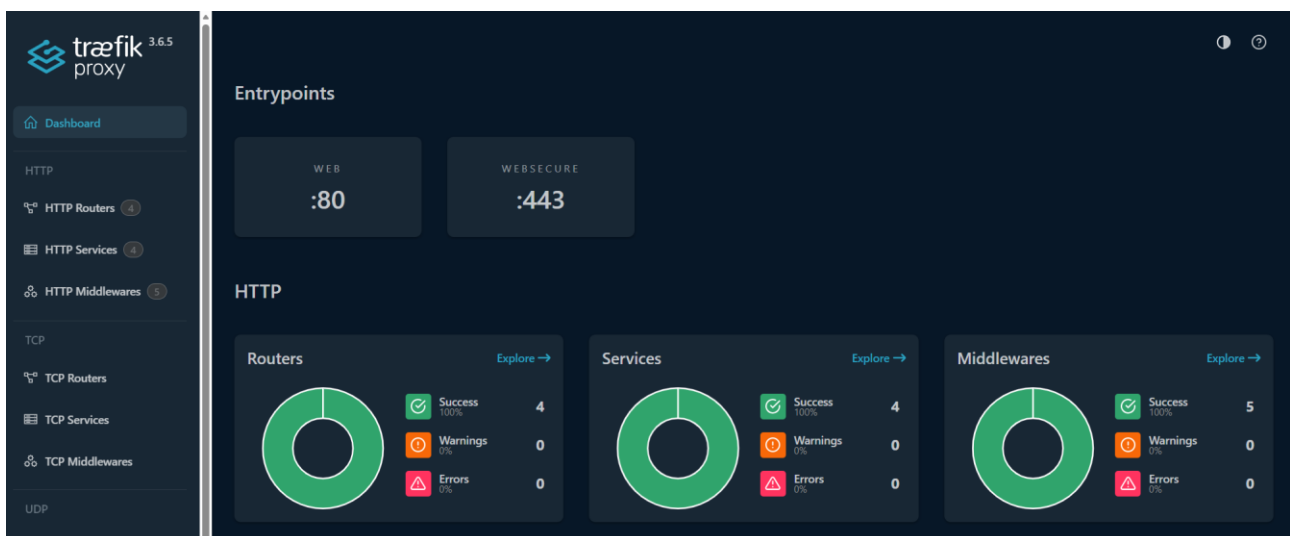
Autorisation requise par <https://traefik.prof.sio-ndlp.fr>

Nom d'utilisateur

Mot de passe

Se connecter Annuler

Exemple de tableau de bord Traefik (version 3.6.5) :



Votre reverse proxy Traefik est maintenant opérationnel ! Etant donné qu'un enregistrement « A » a été enregistré chez un hébergeur et que ce dernier pointe vers notre machine Debian, Traefik gère lui-même l'obtention d'un certificat Let's Encrypt et sécurise, ainsi, la connexion (https) :

Émis pour

Nom commun (CN) traefik.prof.sio-ndlp.fr

Émis par

Nom commun (CN) R12

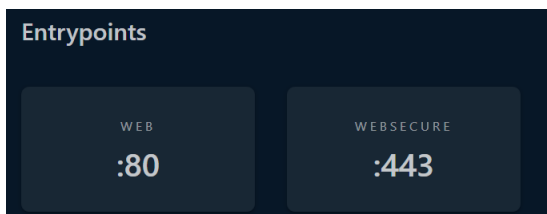
Organisation (O) Let's Encrypt

3. Le tableau de bord de Traefik 3.6.5

Le tableau de bord de Traefik 3.6.5 présente différents modules parmi lesquels :

GESTION DES ENTRYPOINTS

Les **ENTRYPOINTS** sont les **points d'entrées**. Il s'agit des adresses et ports exposés par Traefik qui permettront d'exposer nos applications. La plupart du temps, ce sont les **ports 443 et 80** qui sont exposés par défaut. Si vous saisissez une adresse en « http » elle sera redirigée vers « https » :

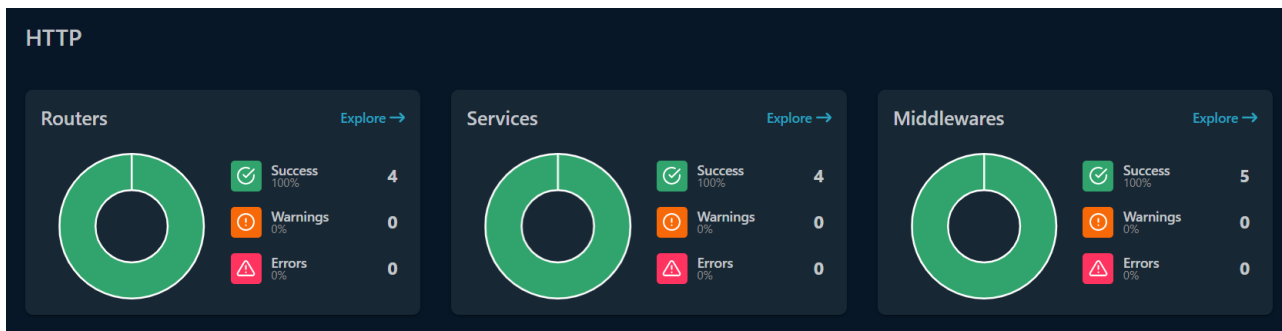


LE MODULE HTTP

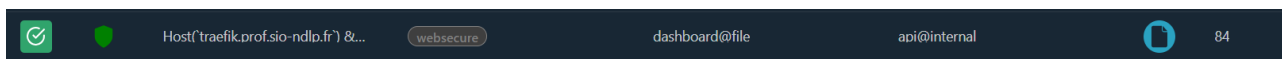
Ce module présente les différents routeurs et services présents ainsi que les « Middlewares » qui ont été déclarés dans le dossier de configuration dynamique de Traefik.

Les **ROUTERS** gèrent les **règles de redirection** du reverse. **Un routeur pointe un service**. On affecte au routeur des URL, des entrypoints.

Les **SERVICES** sont les **applications**, par exemple Portainer, GLPI, etc. Nous pouvons recenser 3 types de services : HTTP, TCP et UDP.



Exemple de routeur (pour le tableau de bord de Traefik) :



Explications complémentaires (« headers », « protocole QUIC », « TLS ») :

Vous trouverez les explications ici :

[TUTORIEL 2025 – INSTALLER TRAEFIK 3.5 EN BARE METAL SUR DEBIAN 13](#)