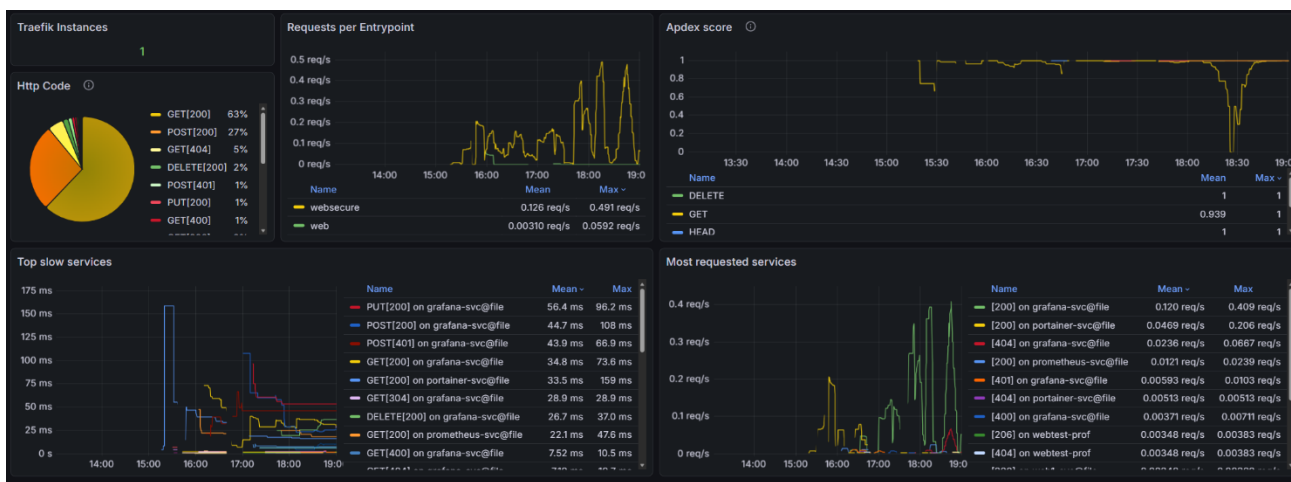


GUIDE DE MISE EN ŒUVRE

TRAEFIK *metrics*

Utiliser des métriques (metrics) dans Traefik 3.6.5

Environnement de travail : Debian 13.2



Prometheus



BTS SIO - Option SISR
Administration des systèmes et réseaux

1. Mise en place des métriques (metrics) dans Traefik 3.6.5

L'utilisation de ce guide suppose que vous avez réalisé, avec succès, les 3 guides précédents pour bien comprendre le fonctionnement général de Traefik.

Traefik expose nativement des métriques Prometheus (pull/scrape), simples à activer. Grafana propose un **dashboard officiel Traefik (standalone)** basé sur ces métriques.

Notion de “metrics” : ce que l’on mesure et pourquoi

Metrics/Logs/Tracing, quelles différences ?

- **Metrics** : mesures **agrégées** (compteurs, latences, taux d’erreur, saturation). Elles sont très utiles pour voir des tendances et déclencher des alertes.
- **Logs** : événements **ligne à ligne** (qui a fait quoi, quand).
- **Tracing** : parcours détaillé d’une requête.

Traefik produit typiquement des métriques portant sur :

- **Un nombre de requêtes** (par entryPoint/router/service),
- **Des codes HTTP** (2xx/3xx/4xx/5xx),
- **Des durées** (histogrammes de latence),
- **Un état interne.**

Traefik supporte plusieurs backends de metrics, dont **Prometheus**, et aussi **OpenTelemetry** (OTLP).

Points clés :

- **Prometheus scrape** l’endpoint métriques de Traefik (mode pull).
- **Grafana** se connecte à Prometheus et affiche des dashboards.

Les métriques se configurent **en statique** (dans le fichier « **traefik.yml** »). Il faut y ajouter un « **entrypoint** » dédié aux métriques (port interne dédié = **9100**).

- Depuis votre machine Traefik, éditez le fichier **/etc/traefik/traefik.yml** et ajoutez ceci :

```
websecure:
  address: ":443"
  http3:
    advertisedPort: 443    # ← Important : annonce le port QUIC aux clients (http/3)
  http:
    tls:
      certResolver: le      # ← Assure que Let's Encrypt est bien configuré
  metrics:
    address: "192.168.100.109:9100"
```

Faites attention à l’indentation (alignement avec « websecure » !

Ici, on a créé un « entryPoint » dédié “metrics” lié à l’IP de la machine Traefik.

- Ajoutez, également, ceci à la fin du fichier :

```
metrics:
prometheus:
  entryPoint: metrics
  addEntryPointsLabels: true
  addServicesLabels: true
  addRoutersLabels: false
  buckets: [0.1, 0.3, 1.2, 5.0]
```

Le début du fichier doit ressembler à ceci (attention à l'indentation ; le dernier « metrics » doit être aligné avec « entryPoints ») :

```
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 est bien configuré

metrics:
  address: "192.168.100.109:9100"

metrics:
  prometheus:
    entryPoint: metrics
    addEntryPointsLabels: true
    addServicesLabels: true
    addRoutersLabels: false
    buckets: [0.1, 0.3, 1.2, 5.0]
```

- Quittez et sauvegardez le fichier
- Relancez Traefik (car il s'agit du fichier « statique » principal de Traefik) avec la commande :

sudo systemctl restart traefik

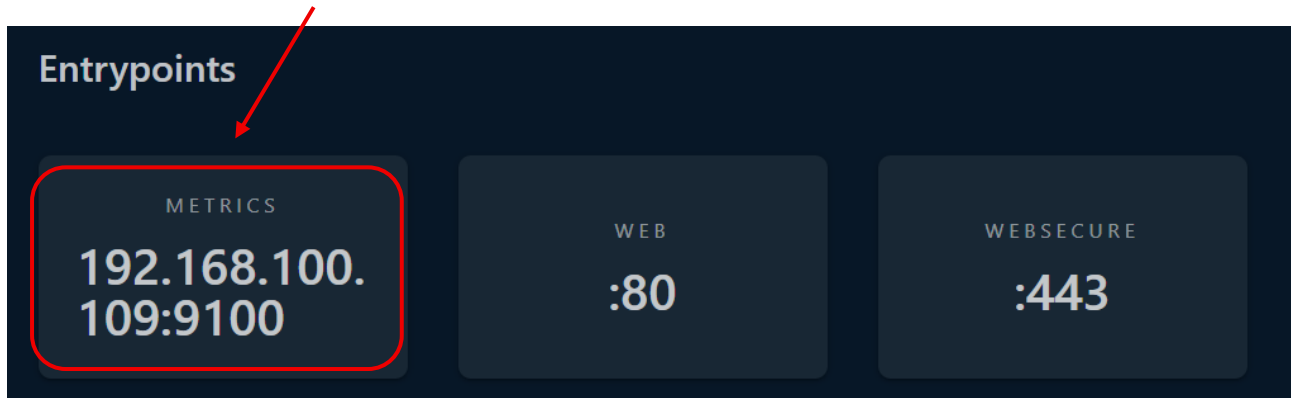
Vérifiez le fonctionnement des métriques avec la commande suivante :

curl -s http://192.168.100.109:9100/metrics | head

Résultat attendu (lignes au format Prometheus avec des métriques) :

```
traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:/etc/traefik$ curl -s http://192.168.100.109:9100/metrics | head
# HELP go_gc_duration_seconds A summary of the wall-time pause (stop-the-world) duration in garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 1.3252e-05
go_gc_duration_seconds{quantile="0.25"} 1.3252e-05
go_gc_duration_seconds{quantile="0.5"} 1.5015e-05
go_gc_duration_seconds{quantile="0.75"} 3.9854e-05
go_gc_duration_seconds{quantile="1"} 3.9854e-05
go_gc_duration_seconds_sum 6.8121e-05
go_gc_duration_seconds_count 3
# HELP go_gc_gogc_percent Heap size target percentage configured by the user, otherwise 100. This value is set by the GOGC environment variable, and the runtime/debug.SetGCPercent function. Sourced from /gc/gogc:percent.
```

Si vous vous connectez au tableau de bord de Traefik, vous verrez l'indication « **METRICS** » qui est apparue dans les « **Entrypoints** » avec l'adresse IP de votre machine Traefik :



2. Déploiement de Prometheus et Grafana

Afin d'exploiter les métriques, nous allons déployer, sur la machine Docker, Prometheus et Grafana. Pour cela, effectuez les manipulations suivantes (**depuis la machine Docker**) :

1^{ère} étape : création de l'arborescence de monitoring

sudo mkdir -p /opt/monitoring/prometheus

2^{ème} étape : création du fichier « prometheus.yml »

sudo nano /opt/monitoring/prometheus/prometheus.yml

Contenu à copier (en respectant l'indentation) :

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: "traefik"
    static_configs:
      - targets: ["192.168.100.109:9100"]
```

3^{ème} étape : création du « docker-compose.yml »

sudo nano /opt/monitoring/docker-compose.yml

Contenu à copier (en respectant l'indentation) ; modifiez les valeurs en **rouge** :

services:

prometheus:

image: prom/prometheus:latest

container_name: prometheus

restart: unless-stopped

command:

- "--config.file=/etc/prometheus/prometheus.yml"

Important pour reverse-proxy en sous-chemin :

- "--web.external-url=**https://metrics.teamX.sio-ndlp.fr/prometheus**"

volumes:

- ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml:ro

- prometheus_data:/prometheus

ports:

- "192.168.100.110:9090:9090"

grafana:

image: grafana/grafana:latest

container_name: grafana

restart: unless-stopped

environment:

- GF_SERVER_ROOT_URL=https://**metrics.teamX.sio-ndlp.fr**/grafana/

- GF_SERVER_SERVE_FROM_SUB_PATH=true

- GF_SERVER_DOMAIN=**metrics.teamX.sio-ndlp.fr**

- GF_SECURITY_ADMIN_USER=**administrateur**

- GF_SECURITY_ADMIN_PASSWORD=**AdminGRAFANA@SIO2**

- GF_USERS_ALLOW_SIGN_UP=false

volumes:

- grafana_data:/var/lib/grafana

ports:

- "192.168.100.110:3000:3000"

volumes:

prometheus_data:

grafana_data:

Lancez le docker compose avec la commande « **docker-compose up -d** » et patientez :

```
docker@DEBIAN-DOCKER-PROF:/opt/monitoring$ sudo docker compose up -d
[+] up 15/22
:: Image grafana/grafana:latest [#####..] 60.8MB / 210.3MB Pulling
:: Image prom/prometheus:latest [###..#####] 55.58MB / 135.1MB Pulling
```

Vérification de la création des 2 conteneurs dans Portainer /

☐	grafana	running	   	monitoring	grafana/grafana:latest	2025-12-24 15:42:21	172.21.0.2	 3000:3000
☐	prometheus	running	   	monitoring	prom/prometheus:latest	2025-12-24 15:42:21	172.21.0.3	 9090:9090

Vérifications du bon fonctionnement (à faire depuis la machine Traefik) :

```
curl -I http://192.168.100.110:9090
```

```
curl -I http://192.168.100.110:3000
```

Résultat attendu :

```
docker@DEBIAN-DOCKER-PROF:/opt/monitoring$ curl -I http://192.168.100.110:3000
HTTP/1.1 301 Moved Permanently
Cache-Control: no-store
Content-Type: text/html; charset=utf-8
Location: https://metrics.prof.sio-ndlp.fr/grafana/
X-Content-Type-Options: nosniff
X-Frame-Options: deny
X-Xss-Protection: 1; mode=block
Date: Wed, 24 Dec 2025 14:42:52 GMT

docker@DEBIAN-DOCKER-PROF:/opt/monitoring$ curl -I http://192.168.100.110:9090
HTTP/1.1 405 Method Not Allowed
Allow: GET, OPTIONS
Content-Type: text/plain; charset=utf-8
X-Content-Type-Options: nosniff
Date: Wed, 24 Dec 2025 14:44:07 GMT
Content-Length: 19
```

4^{ème} étape : exposition de Prometheus/Grafana via Traefik

Depuis la machine Traefik, on génère un mot de passe fort avec la commande suivante :

```
sudo htpasswd -nbB metricsadmin AdminMETRICS@SIO2.
```

Copiez le résultat de cette commande dans un petit fichier texte car nous en aurons besoin :

```
metricsadmin:$2y$05$.....
```

Créez le fichier de configuration dynamique « **metrics-ui.yml** » avec la commande suivante :

```
sudo nano /etc/traefik/dynamic/metrics-ui.yml
```

Contenu à copier (en respectant l'indentation ; valeurs en **rouge** à modifier) :

```
http:
  middlewares:
    metrics-auth:
      basicAuth:
        realm: "Metrics"
        removeHeader: true
      users:
        - "metricsadmin:$2y$10$txM3....."

metrics-rl:
  rateLimit:
    average: 20
    burst: 40

grafana-add-slash:
  redirectRegex:
```

```
regex: "^https?://metrics\\.teamX\\.sio-ndlp\\.fr/grafana$"
replacement: "https://metrics.teamX.sio-ndlp.fr/grafana/"
permanent: true
```

routers:

```
metrics-root:
  entryPoints: ["websecure"]
  rule: "Host(`metrics.teamX.sio-ndlp.fr`) && Path(`/`)"
  middlewares:
    - metrics-rl
    - metrics-auth
  service: noop
  tls:
    certResolver: le
```

grafana-slash:

```
entryPoints: ["websecure"]
rule: "Host(`metrics.teamX.sio-ndlp.fr`) && Path(`/grafana`)"
middlewares:
  - metrics-rl
  - metrics-auth
  - grafana-add-slash
service: noop
tls:
  certResolver: le
```

grafana-ui:

```
entryPoints: ["websecure"]
rule: "Host(`metrics.teamX.sio-ndlp.fr`) && PathPrefix(`/grafana/`)"
middlewares:
  - metrics-rl
  - metrics-auth
  - Compression
  - SecureHeaders
service: grafana-svc
tls:
  certResolver: le
```

prometheus-ui:

```
entryPoints: ["websecure"]
rule: "Host(`metrics.teamX.sio-ndlp.fr`) && PathPrefix(`/prometheus/`)"
middlewares:
  - metrics-rl
  - metrics-auth
  - Compression
  - SecureHeaders
service: prometheus-svc
tls:
  certResolver: le
```

services:

```
grafana-svc:
  loadBalancer:
    servers:
```

- url: "http://192.168.100.110:3000"

prometheus-svc:

loadBalancer:

servers:

- url: "http://192.168.100.110:9090"

Service requis par Traefik

noop:

loadBalancer:

servers:

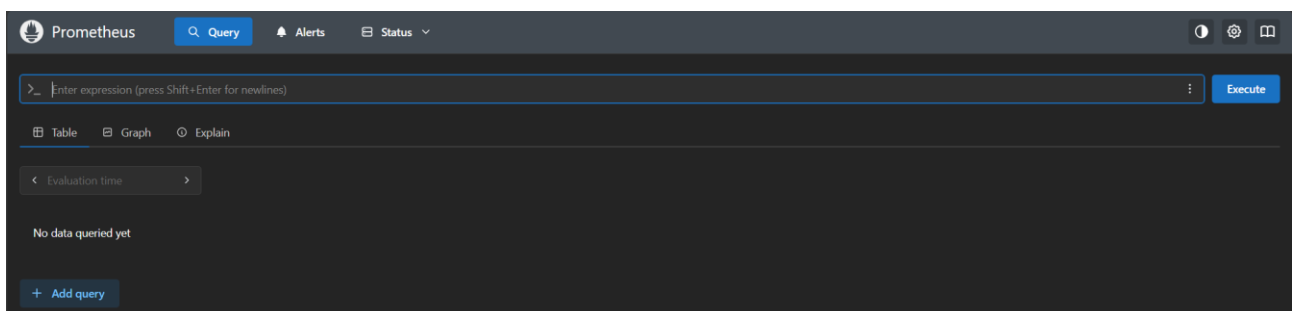
- url: "http://127.0.0.1"

Test de connexion à Prometheus depuis un navigateur :

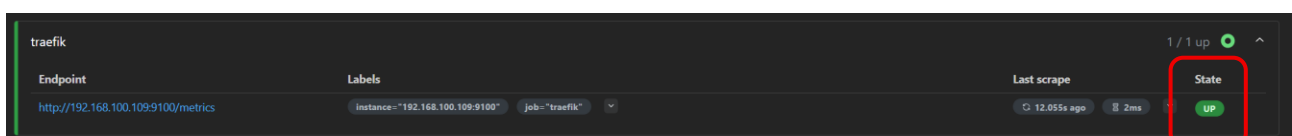
Saisissez l'URL <https://metrics.teamX.sio-ndlp/prometheus>

Une fenêtre d'authentification doit s'afficher ; saisissez « **metricsadmin** » et le mot de passe défini précédemment, puis cliquez le bouton bleu « **Se connecter** » :

La page d'accueil de Prometheus s'affiche :



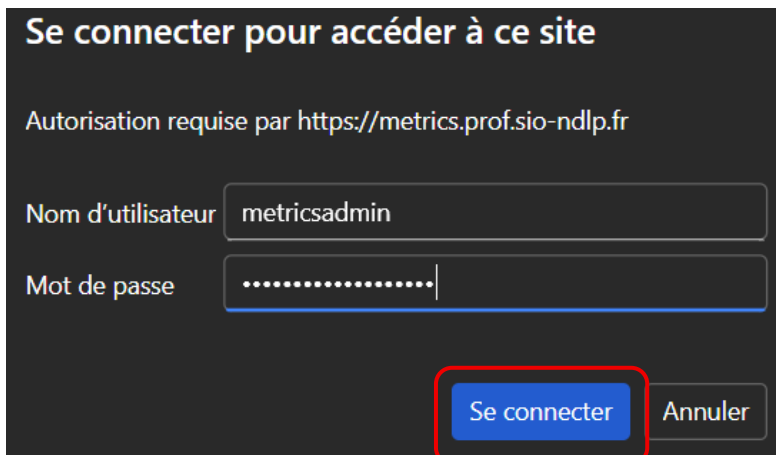
Dans la barre de menus, cliquez sur « **Status** » et « **Target health** » ; Traefik doit être affiché avec le statut « **UP** » en vert :



Test de connexion à Grafana depuis un navigateur :

Saisissez l'URL <https://metrics.teamX.sio-ndlp/grafana>

Une fenêtre d'authentification s'affiche ; saisissez « **metricsadmin** » et le mot de passe créé précédemment et cliquez le bouton bleu « **Se connecter** » :

A dark-themed authentication form titled "Se connecter pour accéder à ce site". Below the title, it says "Autorisation requise par https://metrics.prof.sio-ndlp.fr". There are two input fields: "Nom d'utilisateur" with the value "metricsadmin" and "Mot de passe" with masked dots. At the bottom, there are two buttons: "Se connecter" (highlighted with a red box) and "Annuler".

Se connecter pour accéder à ce site

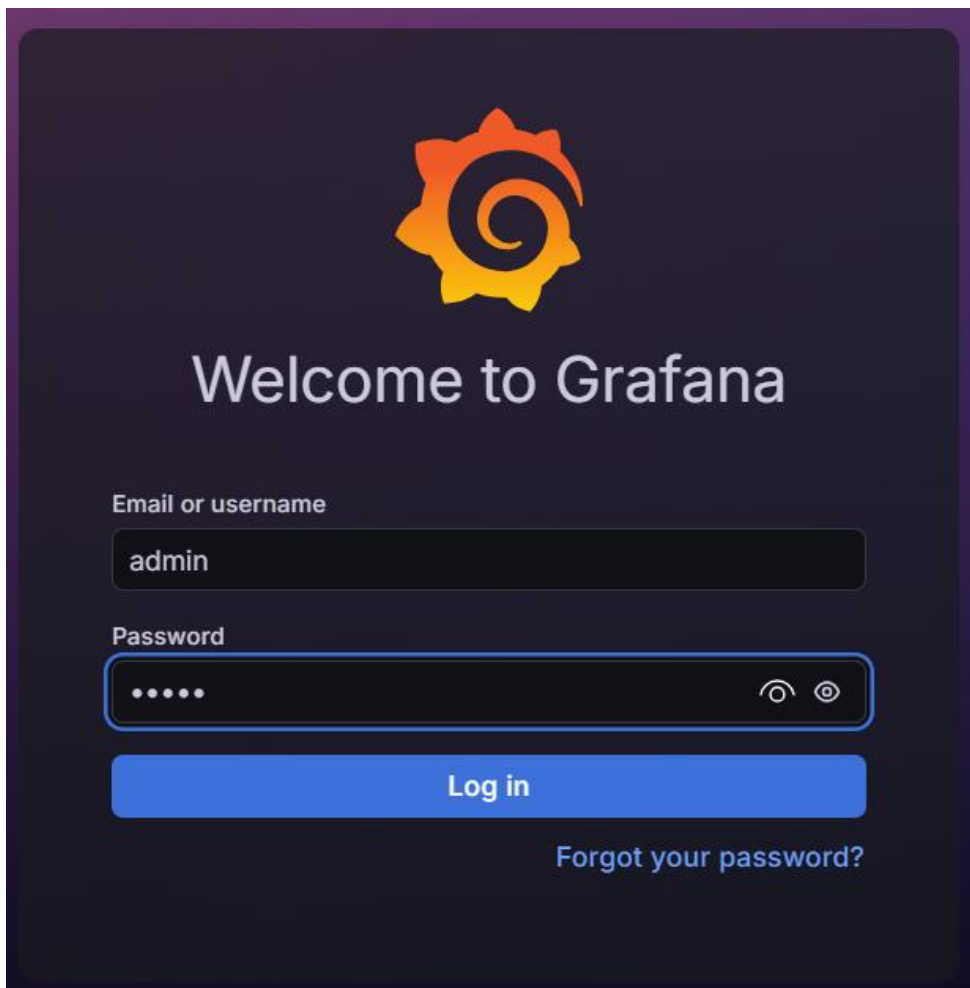
Autorisation requise par https://metrics.prof.sio-ndlp.fr

Nom d'utilisateur metricsadmin

Mot de passe

Se connecter Annuler

La fenêtre Grafana s'affiche alors ; saisissez « **admin** » et le mot de passe qui a été défini lorsque vous avez créé votre docker compose et cliquez le bouton bleu « **Login** » :

The Grafana login screen features the Grafana logo (a yellow and orange gear-like spiral) at the top. Below it, the text "Welcome to Grafana" is displayed. There are two input fields: "Email or username" with the value "admin" and "Password" with masked dots. To the right of the password field are icons for showing and hiding the password. At the bottom, there is a large blue "Log in" button and a link "Forgot your password?".

Welcome to Grafana

Email or username admin

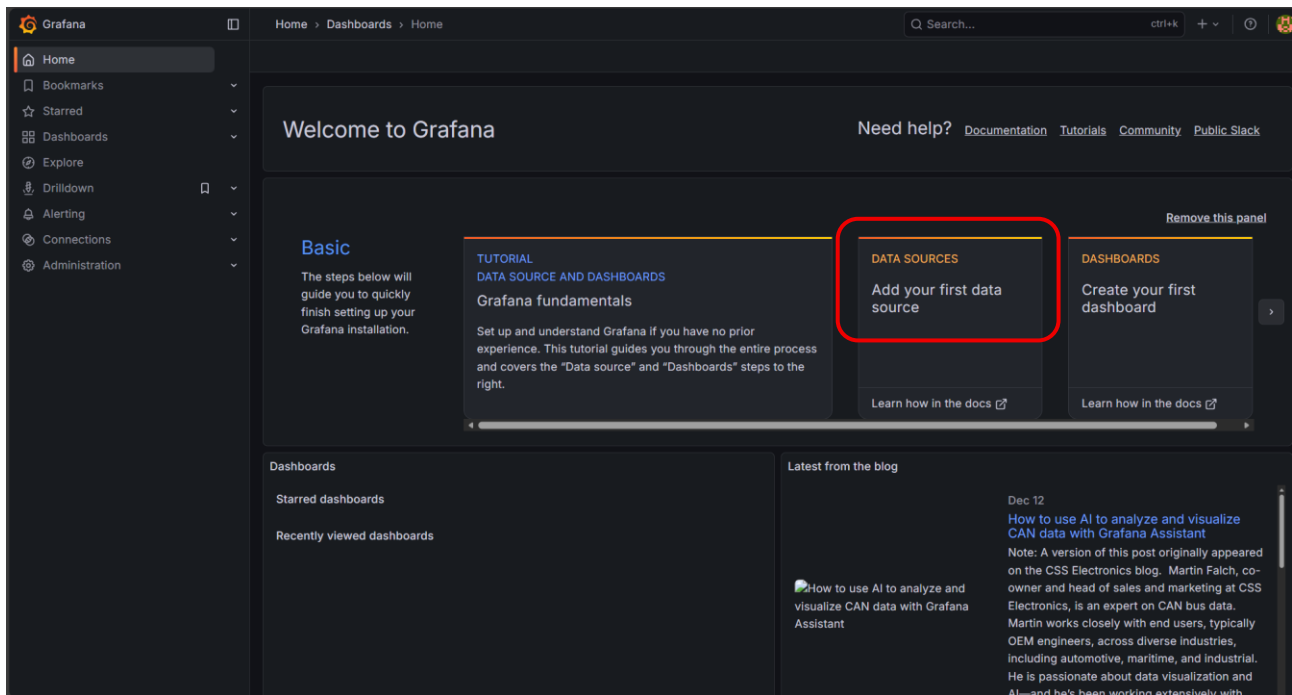
Password

Log in

[Forgot your password?](#)

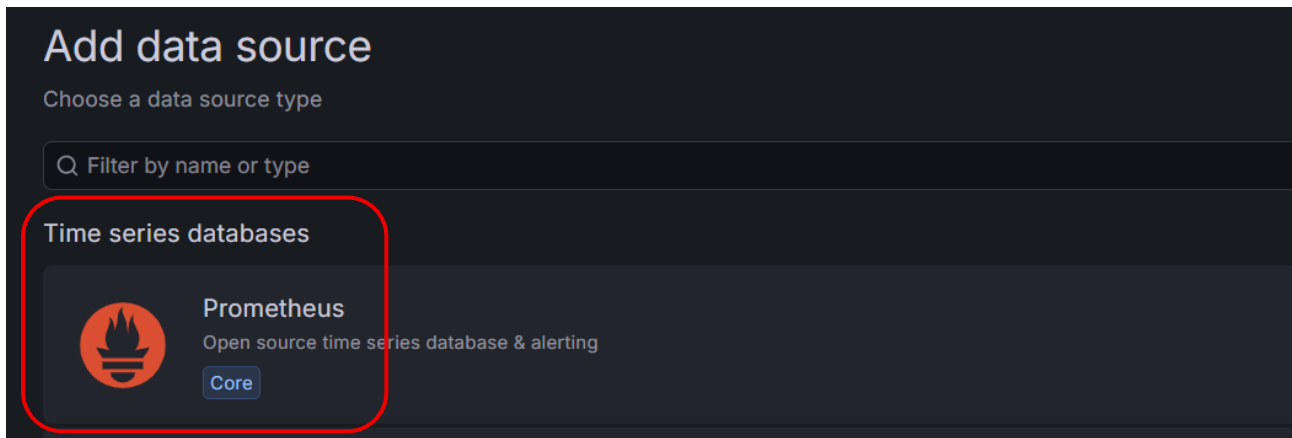
Le mot de passe à saisir, ici, correspond à celui qui a été défini dans le fichier Docker Compose.

La page d'accueil de Grafana doit s'afficher :

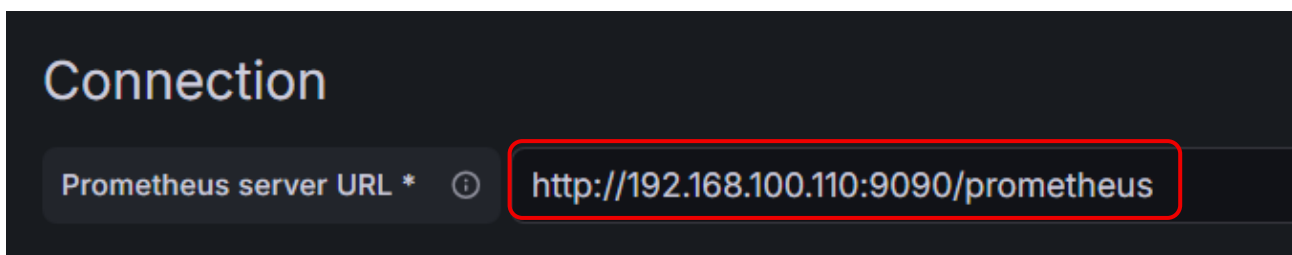


Il faut, dans un premier temps, importer la source de données (Prometheus) ; pour cela :

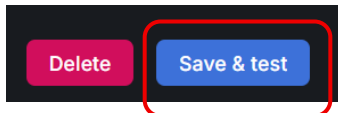
- Cliquez sur « **Add your first data source** »
- Cliquez sur « **Prometheus** » :



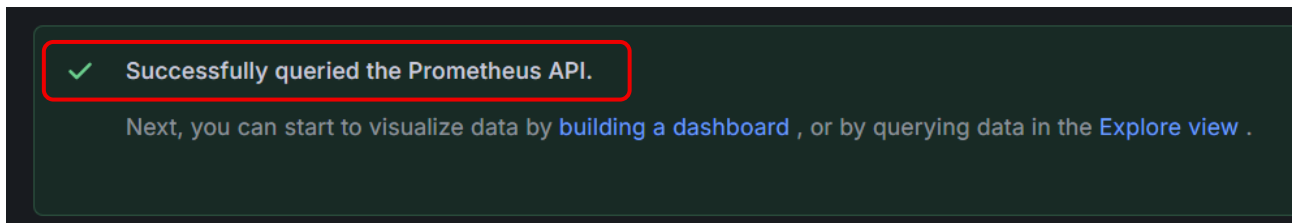
- Saisissez l'URL de la machine Prometheus :



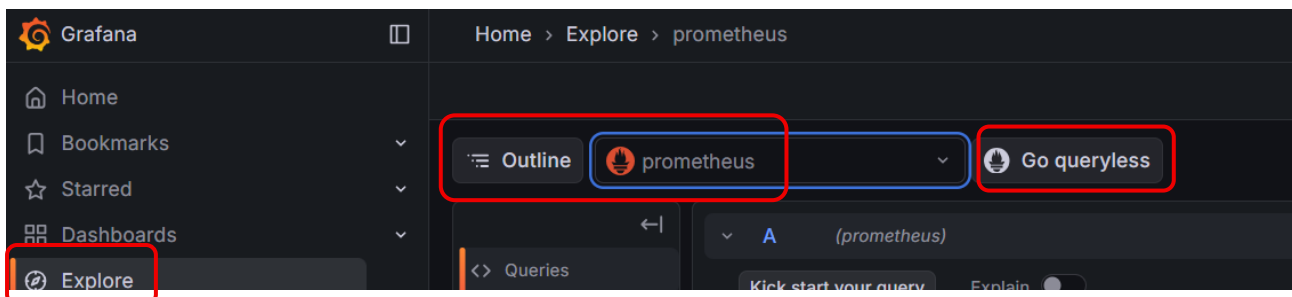
- Cliquez le bouton bleu « **Save & test** » :



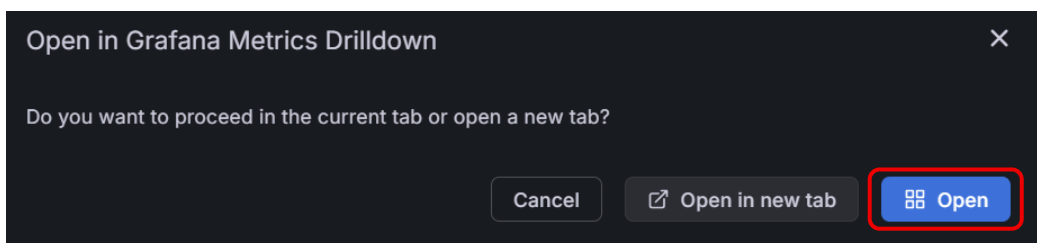
Si vos paramètres sont corrects, la connexion à Prometheus est validée :



Nous allons, maintenant, importer un « dashboard » Grafana. Pour cela, cliquez « **Explore** » dans le menu de gauche, sélectionnez « **Prometheus** » dans « **Outline** » et cliquez sur « **Go queryless** » :



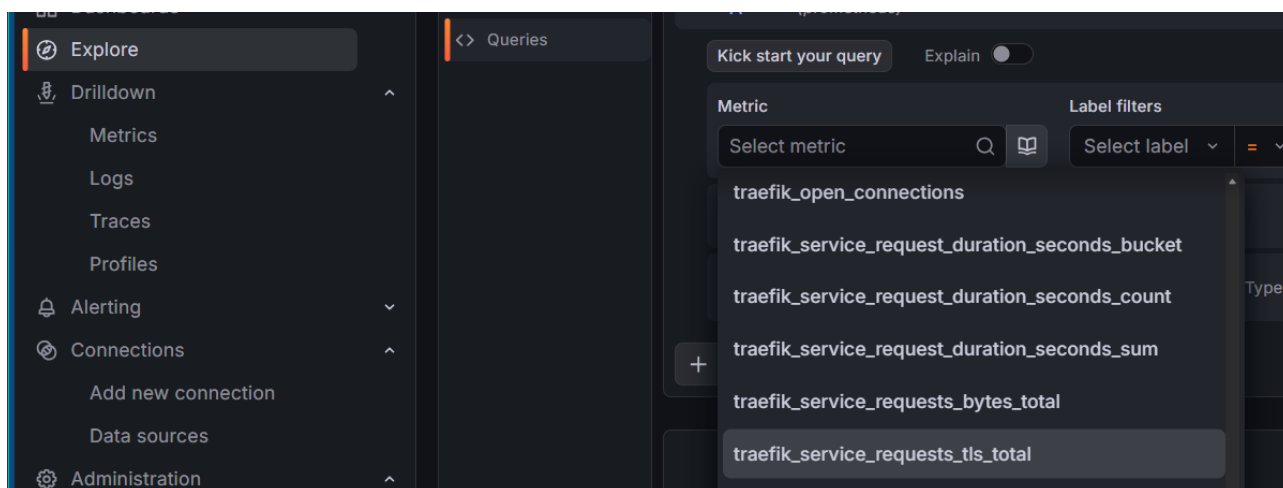
- Cliquez le bouton bleu « **Open** » :



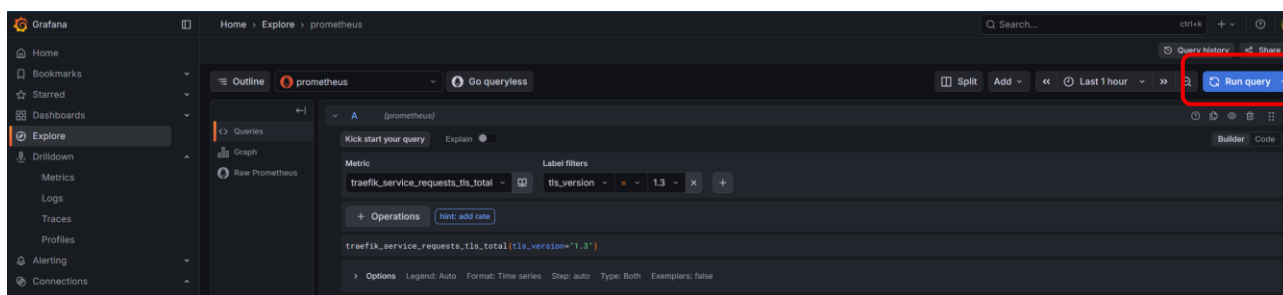
Un tableau de bord de base avec des métriques en provenance de Prometheus s'affiche :



Il est possible, également, de lancer des requêtes directes Traefik en saisissant « **traefik** » dans la rubrique « **Metric** » :



Une fois la requête Traefik sélectionnée, cliquez le bouton bleu « **Run query** » :



Les métriques liées à Traefik s'affichent en temps réel :

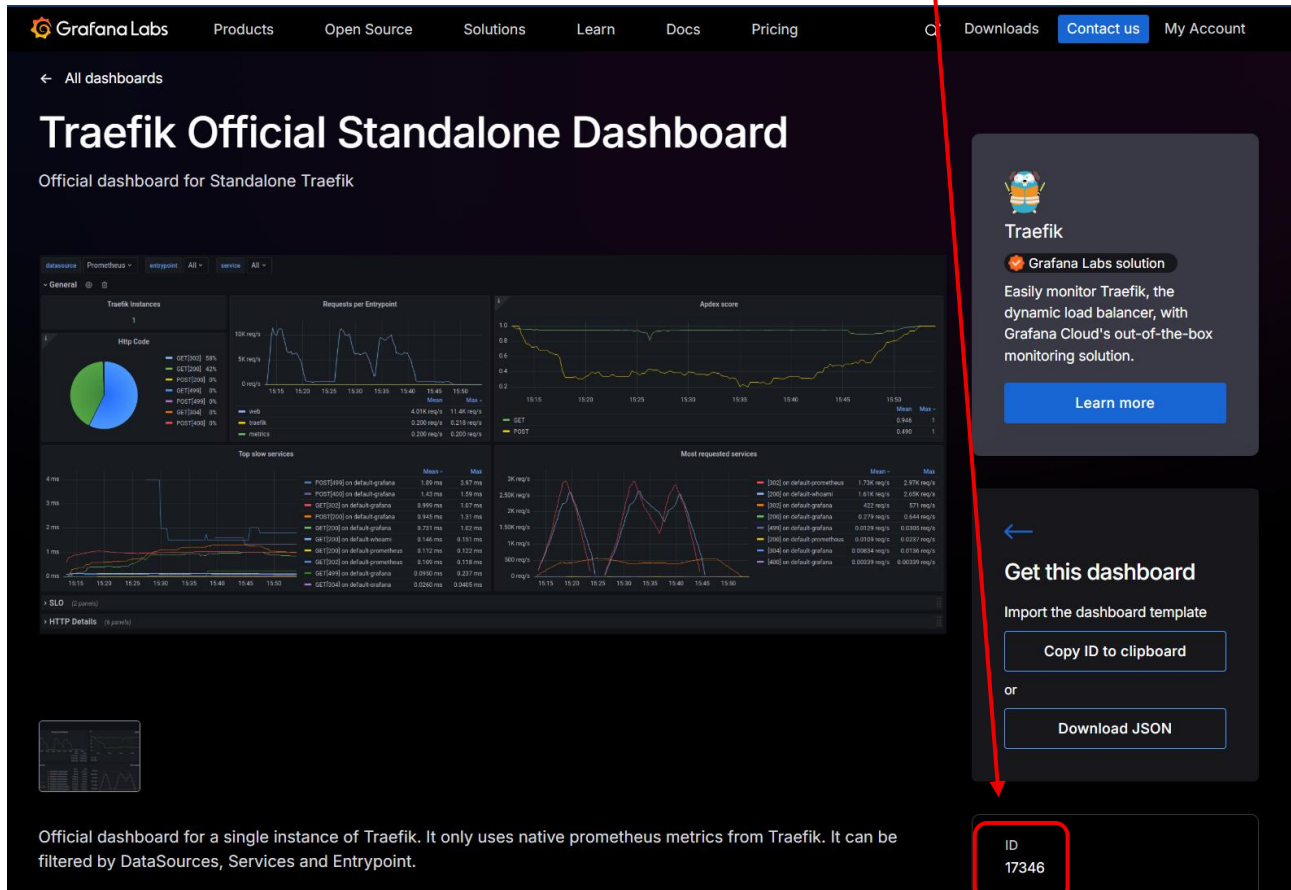


IMPORTATION DE TABLEAUX DE BORD GRAFANA-TRAEFIK

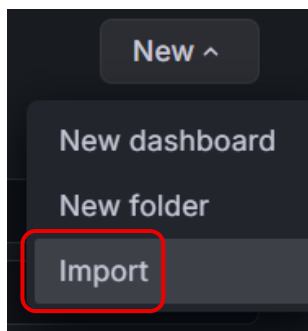
Grafana propose des tableaux de bord en lien avec Traefik. Pour cela, connectez-vous au site suivant :

[traefiklabs Dashboards | Grafana Labs](https://traefiklabs.com/dashboards)

Choisissez le dashboard « **Traefik Official Standalone Dashboard** » (**ID 17346**) :



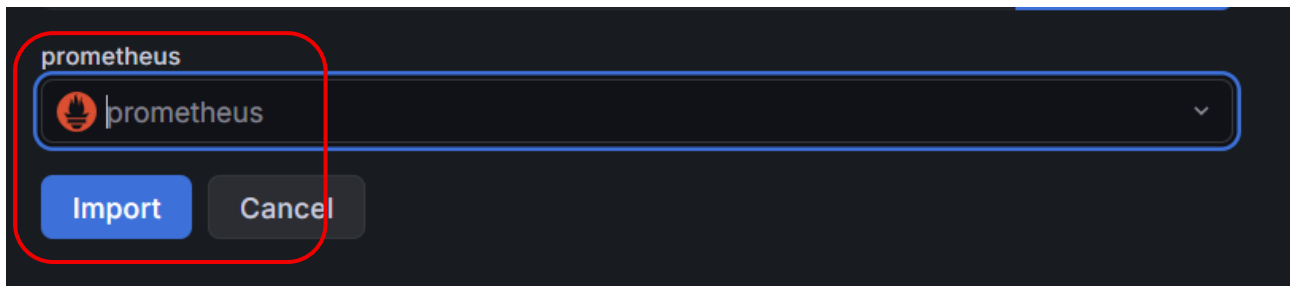
- Dans l'interface Grafana, cliquez sur « **Dashboard** » et sur « **New** » - « **Import** » :



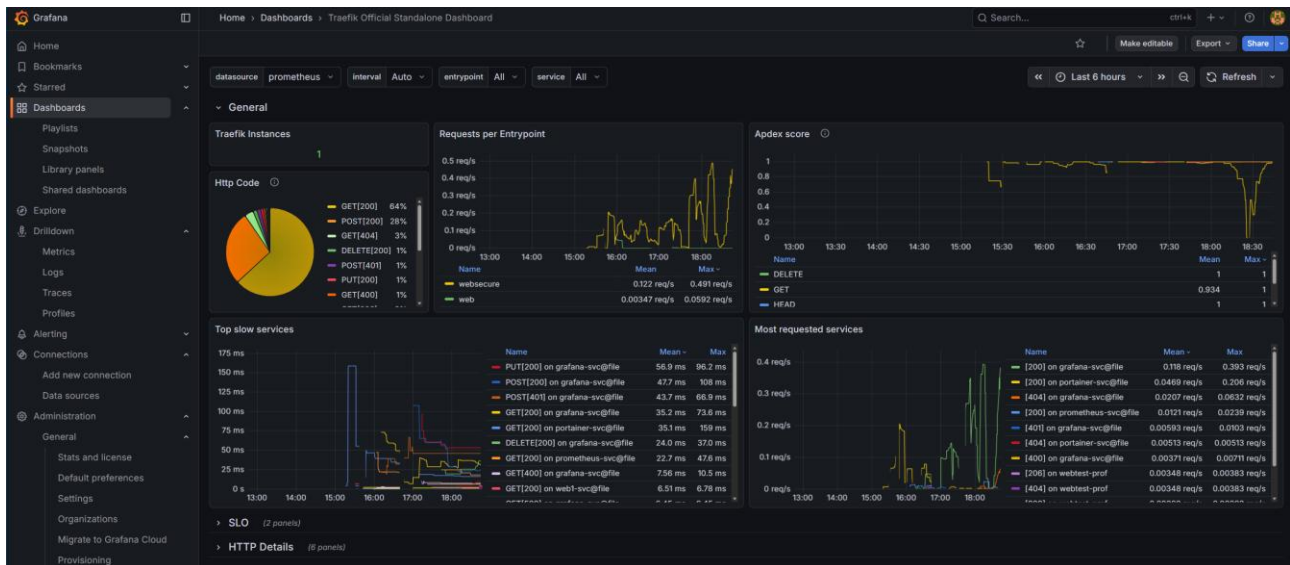
- Saisissez l'ID du dashboard officiel (donné sur le site Grafana ; ici **17346**) et cliquez « **Load** » :

The screenshot shows the Grafana 'Find and import dashboards for common applications at grafana.com/dashboards' page. The page has a dark background with a search bar and a 'Load' button. The search bar contains the text '17346' and the 'Load' button is blue. Both the search bar and the 'Load' button are highlighted with red boxes.

- Sélectionnez la source (Prometheus) et cliquez le bouton bleu « **Import** » :



Le tableau de bord s'affiche :



Pour aller plus loin...

Il est possible de créer des tableaux de bord personnalisés à partir de requêtes Prometheus. A vous de jouer !