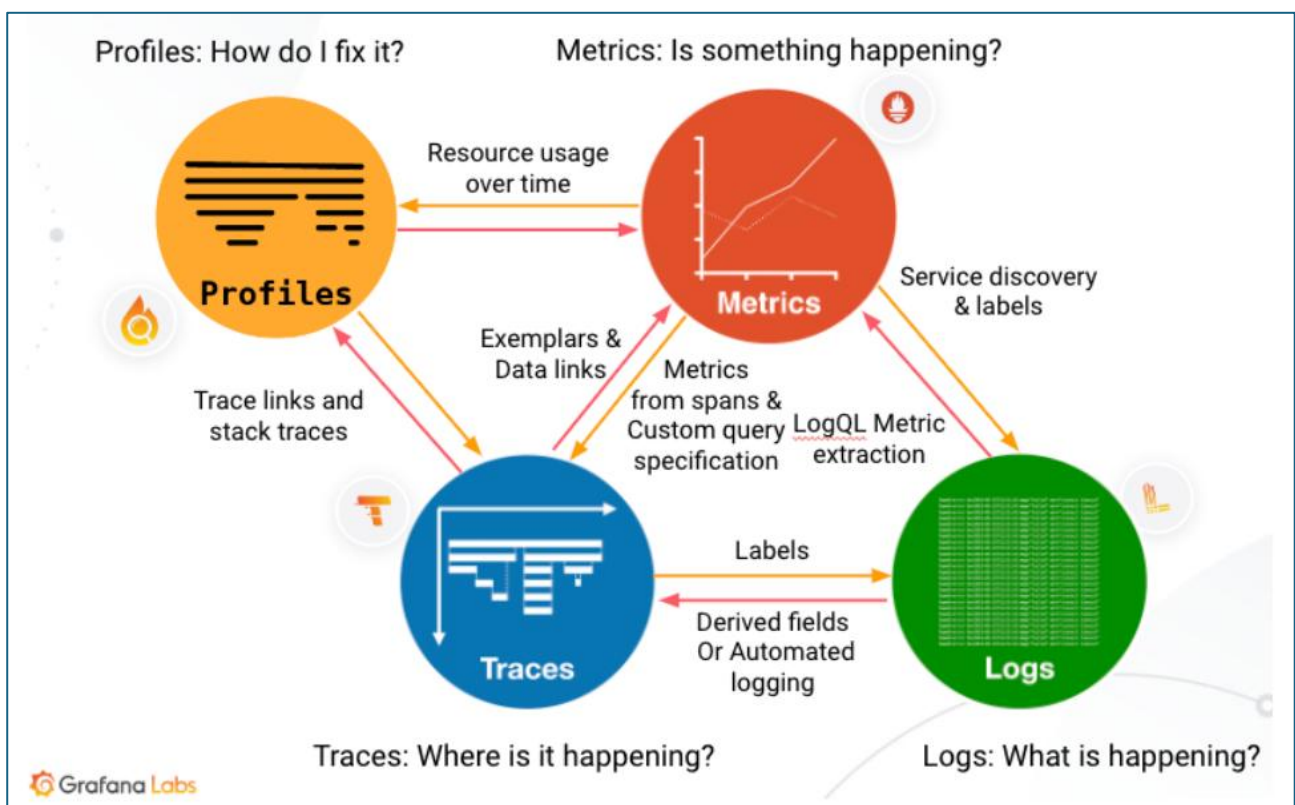


GUIDE DE MISE EN ŒUVRE TRAEFIK *accesslog*

Utiliser l'*accesslog* dans Traefik 3.6.5

Environnement de travail : Debian 13.2



Grafana
Loki
Alloy



BTS SIO - Option SISR
Administration des systèmes et réseaux

1. Qu'est-ce que l'AccessLog dans Traefik ?

L'utilisation de ce guide suppose que vous avez réalisé, avec succès, les 5 guides précédents pour bien comprendre le fonctionnement général de Traefik.

L'**access log** enregistre **chaque requête HTTP traitée par Traefik** : qui appelle, quoi, quand, vers quel routeur/service, quel code HTTP, quelle latence, etc.

Comparaison utile :

- **AccessLog** : “registre des requêtes” (audit + diagnostic + statistiques fines)
- **Metrics** : “tendances agrégées”
- **Tracing** : “détails d'une requête (spans)”

Dans la pratique, l'accesslog est idéal pour :

- prouver qu'une règle de routage/middleware est appliquée,
- identifier un 404/401/502 (et la cause probable),
- faire de la corrélation basique avec les IP clients et les codes.

Accesslog utilisera le format **JSON** car il est facile à « parser », filtrer, importer dans des outils tiers tels que Grafana Alloy/Loki.

2. Mise en œuvre de l'accesslog dans Traefik ?

1^{ère} étape : préparation de l'environnement

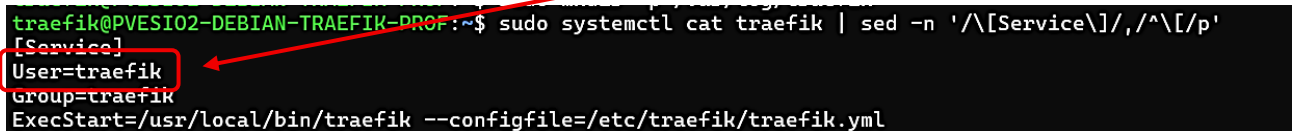
- Créez le dossier de logs **sur la machine Traefik** avec la commande suivante :

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

- Vérifiez l'utilisateur du service Traefik avec la commande suivante :

```
sudo systemctl cat traefik | sed -n '/\[Service\]/,/^\[/p'
```

Cherchez la ligne « **User=.** » et si vous voyez « **User=traefik** », c'est parfait.



```
traefik@PVESI02-DEBIAN-TRAEFIK-PROP:~$ sudo systemctl cat traefik | sed -n '/\[Service\]/,/^\[/p'
[Service]
User=traefik
Group=traefik
ExecStart=/usr/local/bin/traefik --configfile=/etc/traefik/traefik.yml
```

- Appliquez les droits à l'utilisateur « traefik » avec les commandes suivantes :

```
sudo chown -R traefik:traefik /var/log/traefik
```

```
sudo chmod 750 /var/log/traefik
```

2^{ème} étape : activation de l'accesslog dans Traefik

L'accessLog se configure, en ajoutant un bloc « accesslog », dans le fichier de configuration statique de Traefik « **/etc/traefik/traefik.yml** ». Cela signifie qu'il faudra redémarrer Traefik.

- Depuis la machine Traefik, éditez le fichier « **/etc/traefik/traefik.yml** » avec la commande suivante :

sudo nano /etc/traefik/traefik.yml

Contenu à ajouter au fichier (en respectant l'indentation) :

```
accessLog:
  filePath: "/var/log/traefik/access.log"
  format: json
  bufferingSize: 100
```

```
# Filtrage : on l'activera plus tard
# filters:
#   minDuration: "5ms"
```

```
fields:
  defaultMode: keep
  headers:
    defaultMode: drop
  names:
    User-Agent: keep
    Referer: keep
    X-Forwarded-For: keep
    X-Real-Ip: keep
```

```
tracing:
  otlp:
    http:
      endpoint: "http://192.168.100.110:4318/v1/traces"
  sampleRate: 1.0
  serviceName: "traefik-prof"
  resourceAttributes:
    service.namespace: "sio-ndlp"
    deployment.environment.name: "lab"
```

```
accessLog:
  filePath: "/var/log/traefik/access.log"
  format: json
  bufferingSize: 100
```

```
# Filtrage : on l'activera plus tard
# filters:
#   minDuration: "5ms"
```

```
fields:
  defaultMode: keep
  headers:
    defaultMode: drop
  names:
    User-Agent: keep
    Referer: keep
    X-Forwarded-For: keep
    X-Real-Ip: keep
```

Bloc « accesslog: » à ajouter dans le fichier de configuration statique de Traefik pour activer les accesslog.

Remarques :

- format: json : indispensable pour ingestion future vers les outils tiers Loki et Promtail.
- bufferingSize : limite les IO disque (écrit par blocs).
- headers.defaultMode: drop : on évite d'écrire des infos sensibles.
- on garde seulement quelques headers utiles.
- Redémarrez Traefik avec les commandes suivantes :

sudo systemctl restart traefik

sudo systemctl status traefik --no-pager

```
traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:~$ sudo systemctl status traefik --no-pager
● traefik.service - Traefik Reverse Proxy
   Loaded: loaded (/etc/systemd/system/traefik.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-12-25 11:27:33 CET; 7s ago
     Invocation: fe8b1105ba4746e7bda256f67b7896bb
       Main PID: 8055 (traefik)
```

- Vérifiez que le fichier apparaît avec la commande suivante :

sudo ls -l /var/log/traefik/

Résultat attendu :

```
traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:~$ sudo ls -l /var/log/traefik/
total 4
-rw-r--r-- 1 traefik traefik 2320 25 déc. 11:27 access.log
```

- Connectez-vous au tableau de bord de Traefik pour vérifier que l'accesslog est bien activé :

**3^{ème} étape : générer du trafic**

Depuis un poste Windows, on génère du trafic en saisissant les commandes suivantes en Powershell (adaptez les valeurs en rouge) :

```
1..10 | % { curl.exe -k -s -o NUL https://web1.teamX.sio-ndlp.fr/ }
1..10 | % { curl.exe -k -s -o NUL https://web2.teamX.sio-ndlp.fr/ }
1..10 | % { curl.exe -k -s -o NUL https://webtest.teamX.sio-ndlp.fr/ }
```

Vérifiez, depuis la machine Traefik, les accès en saisissant la commande suivante :

sudo tail -n 20 /var/log/traefik/access.log

Résultat attendu (lignes JSON) :

```
traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:~$ sudo tail -n 20 /var/log/traefik/access.log
{"ClientAddr":"90.58.71.216:56688","ClientHost":"90.58.71.216","ClientPort":"56688","ClientUsername":"","DownstreamContentSize":233,"DownstreamStatus":200,"Duration":1416786,"OriginContentSize":233,"OriginDuration":1227504,"OriginStatus":200,"Overhead":189282,"RequestAddr":"web2.prof.sio-ndlp.fr","RequestContentSize":0,"RequestCount":13,"RequestHost":"web2.prof.sio-ndlp.fr","RequestMethod":"GET","RequestPath":"/","RequestPort":"","RequestProtocol":"HTTP/1.1","RequestScheme":"https","RetryAttempts":0,"RouterName":"web2-https@file","ServiceAddr":"192.168.100.110:8082","ServiceName":"web2-svc@file","ServiceURL":"http://192.168.100.110:8082","SpanId":"d6893239d592f10f","StartLocal":"2025-12-25T11:30:40.43358963Z","StartUTC":"2025-12-25T10:30:40.43358963Z","TLSCipher":"TLS_AES_128_GCM_SHA256","TLSVersion":"1.3","TraceId":"6e6b9d6ac84085477daeeaf6acd51c","entryPointName":"websecure","level":"info","msg":"","request_User-Agent":"curl/8.16.0","request_X-Real-IP":"90.58.71.216","time":"2025-12-25T11:30:40+01:00"}
{"ClientAddr":"90.58.71.216:56691","ClientHost":"90.58.71.216","ClientPort":"56691","ClientUsername":"","DownstreamContentSize":233,"DownstreamStatus":200,"Duration":922341,"OriginContentSize":233,"OriginDuration":655566,"OriginStatus":200,"Overhead":266775,"RequestAddr":"web2.prof.sio-ndlp.fr","RequestContentSize":0,"RequestCount":14,"RequestHost":"web2.prof.sio-ndlp.fr","RequestMethod":"GET","RequestPath":"/","RequestPort":"","RequestProtocol":"HTTP/1.1","RequestScheme":"https","RetryAttempts":0,"RouterName":"web2-https@file","ServiceAddr":"192.168.100.110:8082","ServiceName":"web2-svc@file","ServiceURL":"http://192.168.100.110:8082","SpanId":"d6893239d592f10f","StartLocal":"2025-12-25T11:30:40.526553085+01:00","StartUTC":"2025-12-25T10:30:40.526553085Z","TLSCipher":"TLS_AES_128_GCM_SHA256","TLSVersion":"1.3","TraceId":"6e6b9d6ac84085477daeeaf6acd51c","entryPointName":"websecure","level":"info","msg":"","request_User-Agent":"curl/8.16.0","request_X-Real-IP":"90.58.71.216","time":"2025-12-25T11:30:40+01:00"}
{"ClientAddr":"90.58.71.216:56694","ClientHost":"90.58.71.216","ClientPort":"56694","ClientUsername":"","DownstreamContentSize":233,"DownstreamStatus":200,"Duration":1056676,"OriginContentSize":233,"OriginDuration":842395,"OriginStatus":200,"Overhead":214281,"RequestAddr":"web2.prof.sio-ndlp.fr","RequestContentSize":0,"RequestCount":15,"RequestHost":"web2.prof.sio-ndlp.fr","RequestMethod":"GET","RequestPath":"/","RequestPort":"","RequestProtocol":"HTTP/1.1","RequestScheme":"https","RetryAttempts":0,"RouterName":"web2-https@file","ServiceAddr":"192.168.100.110:8082","ServiceName":"web2-svc@file","ServiceURL":"http://192.168.100.110:8082","SpanId":"a063d0cd905e0e55","StartLocal":"2025-12-25T11:30:40.616695594+01:00","StartUTC":"2025-12-25T10:30:40.616695594Z","TLSCipher":"TLS_AES_128_GCM_SHA256","TLSVersion":"1.3","TraceId":"b19c6a7fed7fc6bf1ee5b387c2ec261f","entryPointName":"websecure","level":"info","msg":"","request_User-Agent":"curl/8.16.0","request_X-Real-IP":"90.58.71.216","time":"2025-12-25T11:30:40+01:00"}

```

L'activation de « l'accesslog »
génère des logs d'activité.

4^{ème} étape : exploitation des accesslogs avec l'outil « jq »

- Sur la machine Traefik, installez l'outil « jq » avec les commandes suivantes :

```
sudo apt update
sudo apt install -y jq
```

- Saisissez la commande suivante (commande sur une ligne) :

```
sudo jq '{time: .time, status: .status, method: .requestMethod, host: .requestHost, path: .requestPath, router: .routerName, service: .serviceName, duration: .duration}' /var/log/traefik/access.log | tail -n 10
```

Résultat attendu :

```
traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:~$ sudo jq '{time: .time, status: .status, method: .requestMethod, host: .requestHost, path: .requestPath, router: .routerName, service: .serviceName, duration: .duration}' /var/log/traefik/access.log | tail -n 10
{
  "time": "2025-12-25T11:30:48+01:00",
  "status": null,
  "method": null,
  "host": null,
  "path": null,
  "router": null,
  "service": null,
  "duration": null
}
```

- Saisissez la commande suivante pour filtrer les erreurs > à 400 (commande sur une ligne) :

```
sudo jq 'select(.status >= 400) | {time: .time, status: .status, host: .requestHost, path: .requestPath, router: .routerName, service: .serviceName}' /var/log/traefik/access.log | tail -n 30
```

- Saisissez la commande suivante pour filtrer les 20 chemins les plus demandés :

```
sudo jq -r '.requestPath' /var/log/traefik/access.log | sort | uniq -c | sort -nr | head -n 20
```

Il est conseillé de « réduire le bruit » afin de filtrer les logs par durée ou erreurs par exemple. Pour cela il faut modifier le fichier de configuration statique « /etc/traefik/traefik.yml ».

sudo nano /etc/traefik/traefik.yml

```
accessLog:
  # ...
  filters:
    minDuration: "10ms"
```

```
accessLog:
  # ...
  filters:
    statusCodes:
      - "400-499"
      - "500-599"
```

Modification à apporter :

```
accessLog:
  filePath: "/var/log/traefik/access.log"
  format: json
  bufferingSize: 100
```

```
filters:
  minDuration: "10ms"
  statusCodes:
    - "400-499"
    - "500-599"
```

On peut filtrer les logs par codes d'erreur par exemple afin de « réduire le bruit ».

```
fields:|
  defaultMode: keep
  headers:
    defaultMode: drop
    names:
      User-Agent: keep
      Referer: keep
      X-Forwarded-For: keep
      X-Real-IP: keep
```

Après modification du fichier statique de Traefik, n'oubliez pas de redémarrer Traefik avec la commande « **sudo systemctl restart traefik** ».

Vérifiez le status de Traefik avec la commande « **sudo systemctl status traefik --no-pager** »

```
traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:~$ sudo systemctl status traefik --no-pager
● traefik.service - Traefik Reverse Proxy
   Loaded: loaded (/etc/systemd/system/traefik.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-12-25 11:37:58 CET; 15s ago
   Main PID: 733#(systemd)
   CGroup: /systemd/system/traefik.service
```

5^{ème} étape : rotation des logs

Il est important d'imposer un délai de rotations des logs pour éviter le grossissement illimité du fichier. Pour cela, on utilise l'outil « logrotate » en l'installant ainsi sur la machine Traefik :

sudo apt install -y logrotate

Sur Linux, les fichiers de logs (ici /var/log/traefik/access.log) peuvent grossir très vite. **logrotate** est l'outil standard qui :

- “coupe” le fichier régulièrement (chaque jour, chaque semaine, etc.),
- archive l'ancien fichier (access.log.1, access.log.2.gz, ...),
- compresse,
- conserve un historique limité,
- évite de saturer le disque de la machine Traefik.

Le fichier « **traefik-access** » que l'on va créer est une **règle** qui dit à logrotate comment gérer **ce log-là**.

On crée, ensuite, un fichier « **traefik-access** » dans « **/etc/logrotate.d/** » :

sudo nano /etc/logrotate.d/traefik-access

Contenu à copier :

```
/var/log/traefik/access.log {
daily
rotate 14
compress
delaycompress
missingok
notifempty
copytruncate
create 640 traefik traefik
}
```

- Quittez et sauvegardez le fichier

Explication des règles :

daily

Rotation **tous les jours**.

rotate 14

On garde **14 rotations** :

- access.log.1
- access.log.2.gz
- ...
- jusqu'à 14, puis les plus anciennes sont supprimées.

compress

Les rotations anciennes sont **compressées** (.gz) pour gagner de la place.

delaycompress

La plus récente rotation (access.log.1) n'est pas compressée immédiatement.

missingok

Si le fichier n'existe pas, logrotate **ne plante pas**.

notifempty

Si le fichier est vide, logrotate **ne fait rien** (pas de rotation inutile).

copytruncate (le point clé)

C'est ce qui rend cette option "simple et fiable" avec Traefik :

- logrotate **copie** le contenu actuel de access.log vers un fichier de rotation (access.log.1)
- puis il **vide** le fichier original access.log (truncate à 0 octet)
- Traefik continue d'écrire dans **le même fichier** (même inode), donc pas besoin de lui demander de rouvrir le log.

Certains logiciels, quand logrotate "renomme" le fichier, continuent d'écrire dans l'ancien fichier renommé si on ne leur envoie pas un signal de "reopen". « copytruncate » évite ce problème.

create 640 traefik traefik

Après rotation, logrotate s'assure que **le fichier access.log existe** avec :

- permissions 640 (lecture/écriture propriétaire, lecture groupe, rien pour les autres)
- propriétaire traefik
- groupe traefik

C'est important pour :

- éviter un log en root:root illisible,
- permettre à Traefik de continuer à écrire,
- permettre éventuellement à Promtail (si on l'ajoute au groupe traefik) de lire.
- Forcez une rotation pour tester avec la commande suivante :

sudo logrotate -f /etc/logrotate.d/traefik-access

- Vérifiez le résultat avec la commande suivante :

sudo ls -lah /var/log/traefik/

Résultat attendu :

On doit voir :

- access.log (fichier actuel, vidé ou presque)
- access.log.1 (rotation)
- éventuellement .gz après d'autres rotations

```
traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:~$ sudo ls -lah /var/log/traefik/
total 44K
drwxr-x--- 2 traefik traefik 4,0K 25 déc. 11:40 .
drwxr-xr-x 8 root    root    4,0K 25 déc. 11:24 ..
-rw-r--r-- 1 traefik traefik  0 25 déc. 11:40 access.log
-rw-r--r-- 1 traefik traefik 33K 25 déc. 11:40 access.log.1
```

3. Mise en œuvre de Loki sur la machine Docker

Loki est le **backend de logs** : il reçoit les logs, les stocke, et permet des requêtes **LogQL** depuis Grafana.

Loki : le “serveur de logs” (stockage + requêtes)

Rôle principal

- Recevoir des logs (poussés par Promtail ou autres agents).
- Les **indexer principalement par labels** (comme Prometheus).
- Stocker le contenu des logs (« chunks ») sur le disque.
- Répondre à des requêtes **LogQL** (filtrage, recherche, agrégation temporelle).

Ce que Loki n’est pas

- Ce n’est pas un SIEM complet comme « ELK ».
- Ce n’est pas fait pour indexer chaque mot de chaque log (c’est volontairement plus léger).

Ce que Loki apporte

- Une plateforme centralisée pour chercher : quels 401 sur metrics ?, quels 404 favicon ?, quels 5xx sur web1, web2, webtest, etc ?
- Une intégration avec Grafana (Explore, dashboards, alerting).

Loki indexe surtout par **labels**. Les principaux labels à mettre sont : host, router, service, status, method (labels stables, peu de valeurs).

1^{ère} étape : préparation de l’environnement Loki (**sur la machine Docker**)

- Connectez-vous **à votre machine Docker** et saisissez les commandes suivantes :

sudo mkdir -p /opt/monitoring/loki

- Créez le fichier « loki.yml » avec la commande suivante :

sudo nano /opt/monitoring/loki/loki.yml

Contenu à copier en respectant l’indentation :

```
auth_enabled: false
```

```
server:
```

```
  http_listen_port: 3100
```

```
common:
```

```
  path_prefix: /loki
```

```
  storage:
```

```
    filesystem:
```

```
      chunks_directory: /loki/chunks
```

```
      rules_directory: /loki/rules
```

```
  replication_factor: 1
```

```
ring:
```

```
kvstore:
  store: inmemory
```

```
schema_config:
  configs:
    - from: 2024-01-01
      store: tsdb
      object_store: filesystem
      schema: v13
      index:
        prefix: index_
        period: 24h
```

```
limits_config:
  retention_period: 14d
```

```
compactor:
  retention_enabled: true
  delete_request_store: filesystem
  working_directory: /loki/compactor
  compaction_interval: 10m
```

- Quittez et sauvegardez le fichier

2^{ème} étape : ajout de Loki au docker compose du dossier « /opt/monitoring » créé lors du précédent guide

- Editer le fichier « docker-compose.yml » avec la commande suivante :

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

Contenu à ajouter (en respectant l'indentation) :

```
loki:
  image: grafana/loki:3.0.0
  container_name: loki
  restart: unless-stopped
  command: ["-config.file=/etc/loki/loki.yml"]
  volumes:
    - ./loki/loki.yml:/etc/loki/loki.yml:ro
    - loki_data:/loki
  ports:
    - "192.168.100.110:3100:3100"
```

```
volumes:
  prometheus_data:
  grafana_data:
  loki_data:
```

Contenu à ajouter (bloc Loki) :

```
grafana:
  image: grafana/grafana:latest
  container_name: grafana
  restart: unless-stopped

  environment:
    - GF_SERVER_ROOT_URL=https://metrics.prof.sio-ndlp.fr/grafana/
    - GF_SERVER_SERVE_FROM_SUB_PATH=true
    - GF_SERVER_DOMAIN=metrics.prof.sio-ndlp.fr
    - GF_SECURITY_ADMIN_USER=administrateur
    - GF_SECURITY_ADMIN_PASSWORD=AdminGRAFANA
    - GF_USERS_ALLOW_SIGN_UP=false

  volumes:
    - grafana_data:/var/lib/grafana
  ports:
    - "192.168.100.110:3000:3000"

loki:
  image: grafana/loki:3.0.0
  container_name: loki
  restart: unless-stopped
  command: ["-config.file=/etc/loki/loki.yml"]
  volumes:
    - ./loki/loki.yml:/etc/loki/loki.yml:ro
    - loki_data:/loki
  ports:
    - "192.168.100.110:3100:3100"

volumes:
  prometheus_data:
  grafana_data:
  loki_data:
```

- Relancez le docker compose avec les commandes suivantes :

cd /opt/monitoring

sudo docker compose up -d

Patientez pendant la création du volume et du conteneur Loki :

```
docker@DEBIAN-DOCKER-PROF:/opt/monitoring$ sudo docker compose up -d
[+] up 11/11
✓Image grafana/loki:3.0.0      Pulled
✓Volume monitoring_loki_data  Created
✓Container grafana            Running
✓Container loki               Created
✓Container prometheus         Running
```

- Testez la connexion avec la commande suivante :

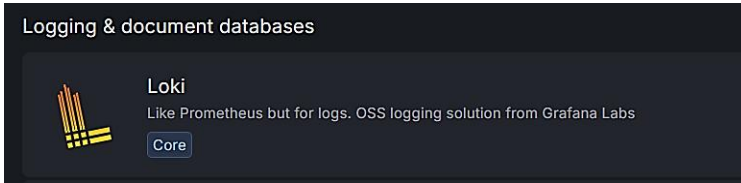
curl -s http://192.168.100.110:3100/ready

Retour attendu :

```
docker@DEBIAN-DOCKER-PROF:/opt/monitoring$ curl -s http://192.168.100.110:3100/ready
ready
```

3^{ème} étape : ajouter Loki comme datasource dans Grafana

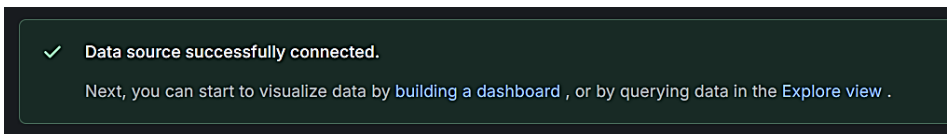
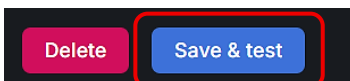
- Connectez-vous à l'interface de Grafana
- Cliquez, dans le volet de gauche, sur « **Connections** » et « **Data sources** »
- Cliquez le bouton bleu (en haut à droite) « **Add new data source** »
- Cliquez sur « **Loki** » :



- Saisissez l'URL de connexion à la source Loki (notre machine Docker avec le port 3100) :



- Cliquez le bouton bleu « **Save & test** » et vérifiez que la connexion s'est bien déroulée :



4. Mise en œuvre de Grafana Alloy sur la machine Traefik

Alloy est l'**agent de collecte** : il lit les logs (fichiers), peut les parser (JSON), extraire un timestamp, ajouter des labels, filtrer du bruit, puis **envoyer à Loki**.

1^{ère} étape : installation de Alloy avec les binaires (**sur la machine Traefik**)

- Installez Grafana Alloy, sur la machine traefik, avec les commandes suivantes :

sudo apt update

sudo apt install -y gpg wget

La commande ci-dessous tient sur une ligne (tout sélectionner) :

```
sudo mkdir -p /etc/apt/keyrings/  
wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee  
/etc/apt/keyrings/grafana.gpg > /dev/null  
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" |  
sudo tee /etc/apt/sources.list.d/grafana.list
```

sudo apt update

sudo apt install -y alloy

- Vérifiez la version Alloy installée avec la commande ci-dessous :

alloy --version

```
traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:~$ alloy --version
alloy, version v1.12.1 (branch: HEAD, revision: 293b514)
  build user:   root@c264d3686852
  build date:   2025-12-15T21:18:32Z
  go version:   go1.25.1
  platform:    linux/amd64
  tags:        slicelabels,netgo,builtinassets,promtail_journal_enabled,slicelabels
```

2^{ème} étape : configurer Alloy pour envoyer les logs de Traefik vers Loki ([sur la machine Traefik](#))

- Editez le fichier « config.alloy » avec la commande suivante :

sudo nano /etc/alloy/config.alloy

Contenu à coller à la fin du fichier :

```
local.file_match "traefik_access" {
  path_targets = [{
    __path__ = "/var/log/traefik/access.log*",
    job      = "traefik",
    node     = "traefik-109",
  }]
  sync_period = "5s"
}

loki.source.file "traefik_access" {
  targets      = local.file_match.traefik_access.targets
  forward_to   = [loki.process.traefik_access.receiver]
  tail_from_end = true
}

loki.process "traefik_access" {
  stage.json {
    expressions = {
      ts      = "time",
      status  = "status",
      method  = "requestMethod",
      vhost   = "requestHost",
      path    = "requestPath",
      router  = "routerName",
      service = "serviceName",
    }
    drop_malformed = true
  }

  stage.timestamp {
    source = "ts"
    format = "RFC3339Nano"
  }

  stage.labels {
```

```

values = {
  vhost = "",
  router = "",
  service = "",
  method = "",
  status = "",
}
}

forward_to = [loki.write.lan.receiver]
}

loki.write "lan" {
  endpoint {
    url = "http://192.168.100.110:3100/loki/api/v1/push"
  }
}
}

```

- Ajoutez, sur la machine Traefik, Alloy au groupe Traefik et faites redémarrer Alloy :

```

sudo usermod -aG traefik alloy
sudo systemctl restart alloy

```

- Faites redémarrer Alloy avec les commandes suivantes :

```

sudo systemctl restart alloy
sudo journalctl -u alloy -n 100 --no-pager

```

Résultat attendu :

```

traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:~$ sudo systemctl status alloy --no-pager
● alloy.service - Vendor-agnostic OpenTelemetry Collector distribution with programmable pipelines
   Loaded: loaded (/usr/lib/systemd/system/alloy.service; disabled; preset: enabled)
   Active: active (running) since Thu 2025-12-25 14:22:01 CET; 11s ago
     Invocation: 8d14292922234a966818994c6745ffb29
       Docs: https://grafana.com/docs/alloy
    Main PID: 8981 (alloy)
      Tasks: 6 (limit: 1095)
     Memory: 34.7M (peak: 34.9M)
        CPU: 140ms
    CGroup: /system.slice/alloy.service
            └─8981 /usr/bin/alloy run --storage.path=/var/lib/alloy/data /etc/alloy/config.alloy

```

3^{ème} étape : activer Alloy au démarrage de la machine

```

sudo systemctl enable alloy

```

- Vérifiez que Loki est prêt avec la commande suivante :

```

curl -s http://192.168.100.110:3100/ready

```

Résultat attendu :

```

traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:~$ curl -s http://192.168.100.110:3100/ready
ready

```

- Vérifiez qu'Alloy lit bien le fichier de log Traefik avec la commande suivante :

```
sudo ls -l /var/log/traefik/access.log
```

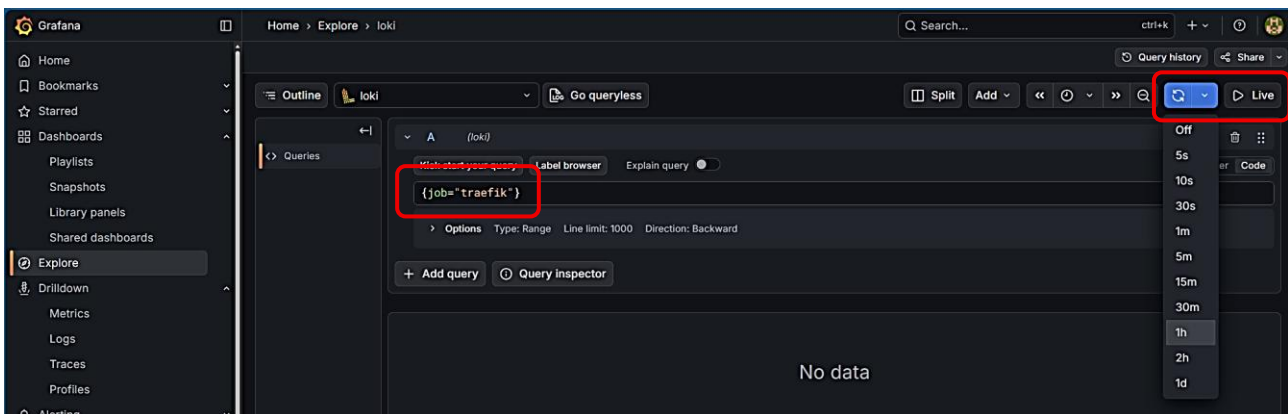
Résultat attendu :

```
traefik@PVESIO2-DEBIAN-TRAEFIK-PROF:~$ sudo ls -l /var/log/traefik/access.log
-rw-r--r-- 1 traefik traefik 690783 25 déc. 17:15 /var/log/traefik/access.log
```

Tout est prêt ! Il faut maintenant vérifier que Grafana reçoit bien des logs !

5. Récupération des logs dans Grafana

- Connectez-vous à Grafana
- Cliquez, dans le volet de gauche, « **Explore** »
- Saisissez, dans la barre de recherche, le « job » pour Traefik : **{job="traefik"}**
- Cliquez le bouton bleu (en haut à droite) et choisissez une période de temps :



- Cliquez sur le bouton « **Live** » en haut à droite
- Depuis un navigateur, connectez-vous à l'interface de votre Portainer CE ; le log s'affiche !

Résultat attendu :

```
2025-12-25 17:34:25 {"ClientAddr":"90.58.71.216:53358","ClientHost":"90.58.71.216","ClientPort":"53358","ClientUsername":"","DownstreamContentSize":564,"DownstreamStatus":200,"Duration":47614127,"OriginContentSize":564,"OriginDuration":47477930,"OriginStatus":200,"Overhead":136207,"RequestAddr":"portainer.prof.sio-ndlp.fr","RequestContentSize":0,"RequestCount":611,"RequestHost":"portainer.prof.sio-ndlp.fr","RequestMethod":"GET","RequestPath":"/api/system/version","RequestPort":"","RequestProtocol":"HTTP/3.0","RequestScheme":"https","RetryAttempts":0,"RouterName":"portainer@file","ServiceAddr":"192.168.100.110:9443","ServiceName":"portainer-svc@file","ServiceURL":"https://192.168.100.110:9443","SpanId":"7b06bc2a475b966e","StartLocal":"2025-12-25T17:34:25.810354237+01:00","StartUTC":"2025-12-25T16:34:25.810354237Z","TLSCipher":"TLS_CHACHA20_POLY1305_SHA256","TLSVersion":"1.3","TraceId":"4497b77818c252cdc7c777c56f91b88b","entryPointName":"websecure","level":"info","msg":"","request_Referer":"https://portainer.prof.sio-ndlp.fr/","request_User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36 Edg/143.0.0.0","request_X-Real-IP":"90.58.71.216","time":"2025-12-25T17:34:25+01:00"}
2025-12-25 17:34:26 {"ClientAddr":"90.58.71.216:53358","ClientHost":"90.58.71.216","ClientPort":"53358","ClientUsername":"","DownstreamContentSize":180436,"DownstreamStatus":200,"Duration":679010717,"GzipRatio":1.0889456649449112,"OriginContentSize":196485,"OriginDuration":678692089,"OriginStatus":200,"Overhead":318628,"RequestAddr":"portainer.prof.sio-ndlp.fr","RequestContentSize":0,"RequestCount":613,"RequestHost":"portainer.prof.sio-ndlp.fr","RequestMethod":"GET","RequestPath":"/api/motd","RequestPort":"","RequestProtocol":"HTTP/3.0","RequestScheme":"https","RetryAttempts":0,"RouterName":"portainer@file","ServiceAddr":"192.168.100.110:9443","ServiceName":"portainer-svc@file","ServiceURL":"https://192.168.100.110:9443","SpanId":"ece7c3e5b443460b","StartLocal":"2025-12-25T17:34:25.81160901+01:00","StartUTC":"2025-12-25T16:34:25.81160901Z","TLSCipher":"TLS_CHACHA20_POLY1305_SHA256","TLSVersion":"1.3","TraceId":"8aeb778d25b8ed3dfd381fc6380a65ab","entryPointName":"websecure","level":"info","msg":"","request_Referer":"https://portainer.prof.sio-ndlp.fr/","request_User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
```

Exemples de requêtes à lancer depuis Grafana (module « Explore ») :

1) Afficher tous les logs Traefik

```
{job="traefik"}
```

2) Filtrer sur un domaine (web1)

```
{job="traefik"} |= "web1.teamX.sio-ndlp.fr"
```

3) Filtrer sur une URL (ex: /grafana)

```
{job="traefik"} |= "RequestPath\":" /grafana"
```

4) Filtrer sur une méthode (GET / POST)

Variante “texte”

```
{job="traefik"} |= "\"RequestMethod\":" POST\""
```

5) Voir uniquement les erreurs (4xx/5xx)

Variante “texte”

```
{job="traefik"} |= "\"DownstreamStatus\":" 404"
```

ou plus large :

```
{job="traefik"} |~ "\"DownstreamStatus\":" (4|5)[0-9]{2}"
```

Requêtes “métriques à partir des logs” (très utiles en dashboard)

Ces requêtes retournent des séries temporelles (panels “Time series”).

6) Volume de logs (nb de lignes / minute)

```
sum(count_over_time({job="traefik"}[1m]))
```

7) Requêtes par hôte (Top 10 sur 5 min)

Si le log contient bien le host dans le texte, on peut faire une extraction via regex.

```
topk(10,
  sum by (host) (
    count_over_time(
      {job="traefik"}
      | regexp "RequestHost\":" (?P<host>[^\"]+)"
    [5m]
  )
)
```

8) Top 10 des URLs les plus demandées (sur 5 min)

```
topk(10,
  sum by (path) (
    count_over_time(
      {job="traefik"}
      | regexp "RequestPath\":" (?P<path>[^\"]+)"
    [5m]
  )
)
```

9) Taux d'erreurs (ratio 4xx/5xx)**a) Erreurs (4xx+5xx) par minute**

```
sum(count_over_time({job="traefik"} |~ "\"DownstreamStatus\":(4|5)[0-9]{2}"[1m]))
```

b) Ratio erreurs / total

```
(  
  sum(count_over_time({job="traefik"} |~ "\"DownstreamStatus\":(4|5)[0-9]{2}"[5m]))  
)  
/  
(  
  sum(count_over_time({job="traefik"}[5m]))  
)
```

Les possibilités sont multiples ; à vous de jouer !