2025

Mise en place d'un serveur de logs GRAYLOG avec Docker-compose

ASSURMER

Auteurs: DUBOIS Bastien

Validateurs: DEGEN Loïc, SAMARTANO Joseph

SOMMAIRE

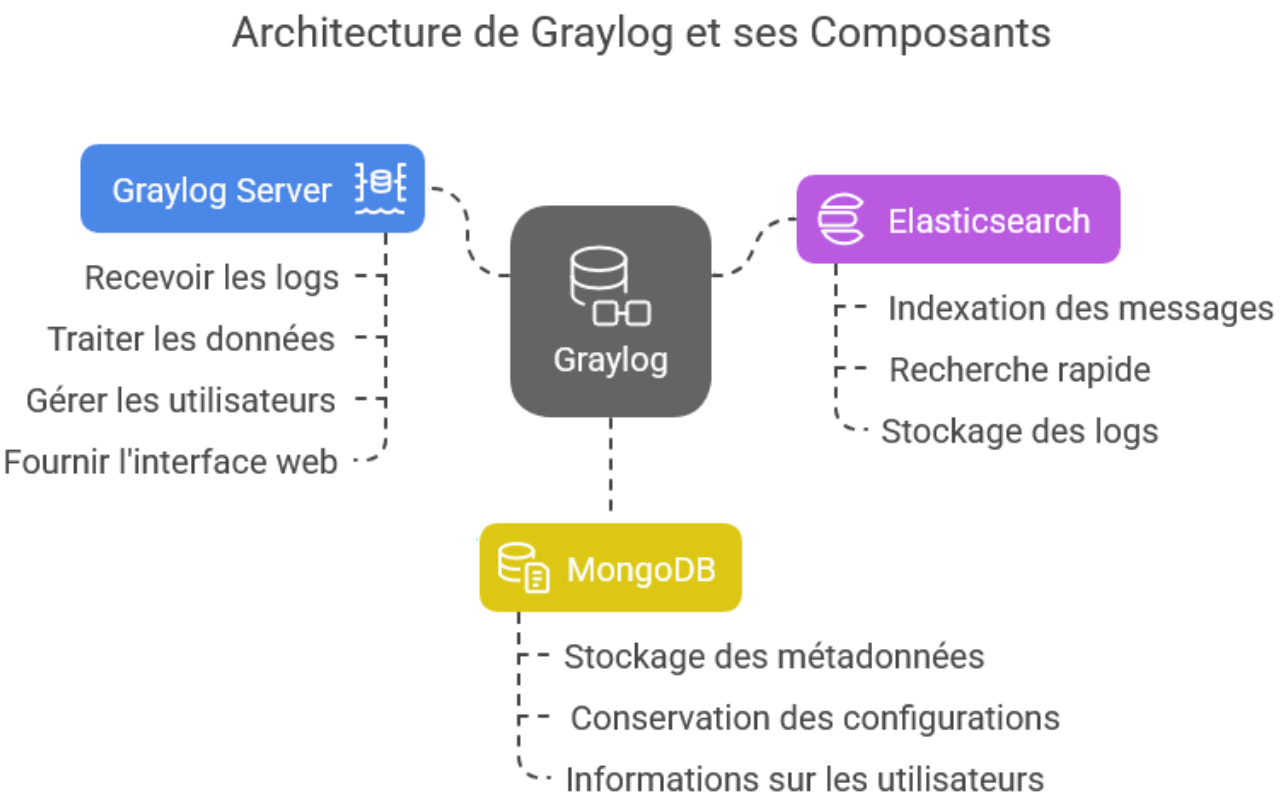
Table des matières

Fonctionnement de Graylog

Graylog est un système de gestion de logs centralisé, open source, conçu pour collecter, indexer et analyser des données de journalisation (logs) provenant de différentes sources dans une infrastructure informatique.

Il est particulièrement utile dans les environnements avec plusieurs serveurs, applications et dispositifs réseau qui génèrent tous des logs que l'on souhaite analyser de manière centralisée pour faciliter le dépannage, la sécurité et l'audit.

Le serveur **Graylog** est le cœur du système qui collecte, traite et expose les données de journalisation via une interface web intuitive. Il s'appuie sur **Elasticsearch** pour le stockage et l'indexation des logs, offrant ainsi des capacités de recherche rapide et avancée sur de grands volumes de données. **MongoDB** complète l'architecture en conservant les configurations du système, les paramètres utilisateurs et les métadonnées.



Flux de traitement des logs

- **Collecte**: Les logs sont envoyés à Graylog via différents protocoles :

  Syslog (UDP/TCP)
  GELF (Graylog Extended Log Format)
  Beats (Filebeat, Winlogbeat)
  API REST
  Fichiers logs

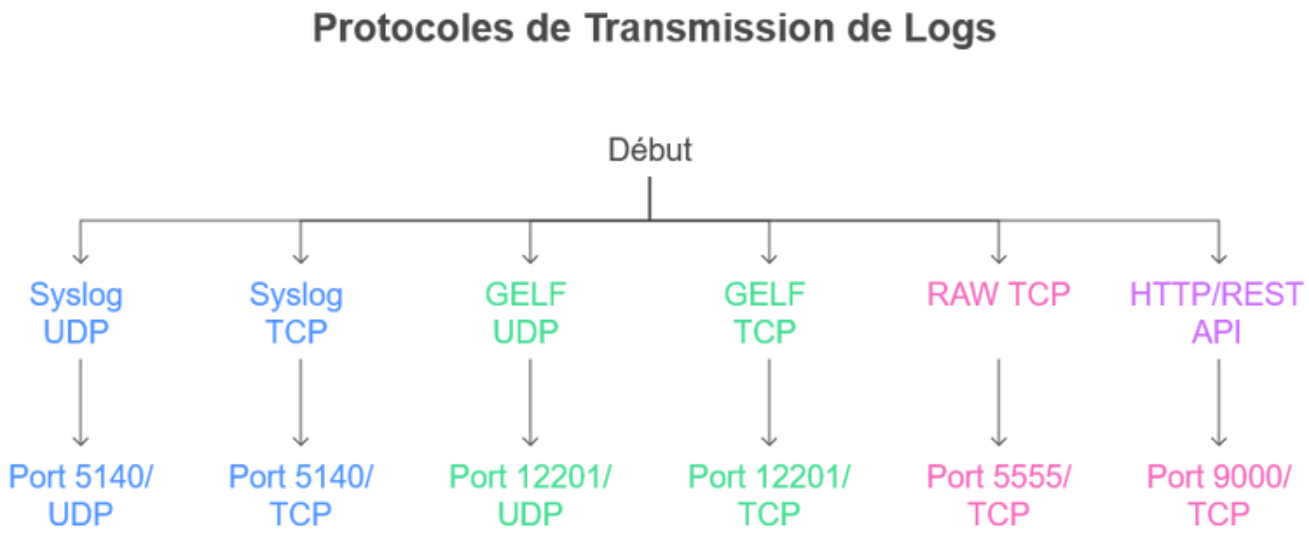
- **Traitement**: Une fois reçus, Graylog peut :

  Extraire des champs particuliers
  Enrichir les messages avec des informations supplémentaires
  Normaliser les formats
  Appliquer des règles de traitement

- **Stockage** : Les messages traités sont indexés dans Elasticsearch.

- **Recherche et analyse**: Via l'interface web, vous pouvez :

  Rechercher des logs spécifiques
  Créer des tableaux de bord
  Configurer des alertes
  Générer des rapports



Pour la transmission des logs, plusieurs couches de protocoles fonctionnent ensemble :

- **Le protocole Syslog et GELF UDP / TCP :**

Syslog et **GELF** sont des protocoles de la couche application, il définissent le format et la structure des messages de logs.
TCP et **UDP** sont des protocoles de la couche transport, ils définissent comment ces messages sont transportés sur le réseau.

Etape 1. Création du répertoire de travail

En premier lieu, créer un dossier graylog avec la commande **mkdir -p ~/graylog** et aller dessus avec **cd ~/graylog :**

```
assurmer@SRV-GRAYLOG:~$ mkdir -p ~/graylog
```

```
assurmer@SRV-GRAYLOG:~$ cd ~/graylog
assurmer@SRV-GRAYLOG:~/graylog$
```

Etape 2. Générer un mot de passe admin

Pour améliorer la sécurité, créer un mot de passe fort puis générer son hashage et ensuite copier le hashage affiché:

```
assurmer@SRV-GRAYLOG:~/graylog$ echo -n "votremotdepasse" | sha256sum
```

Etape 3. Création du fichier .env

Ensuite, créer un fichier ".env" toujours dans ~/graylog, il va servir à stocker les variables sensibles (hashage des mots de passe), par mesure de sécurité.
Renforcer la sécurité en restreignant les permissions :

```
assurmer@SRV-GRAYLOG:~/graylog$ sudo touch .env
```

```
assurmer@SRV-GRAYLOG:~/graylog$ chmod 600 .env
```

Entrer dans le fichier avec la commande **nano .env** et tapez, puis sauvegarder et quitter :
GRAYLOG_PASSWORD_SECRET=UnePhraseSecrète123456!
GRAYLOG_ROOT_PASSWORD_SHA2=[coller ici le hashage du mot de passe admin]
GRAYLOG_HTTP_EXTERNAL_URI=http://[IPduserveur]:9000/
GRAYLOG_ROOT_TIMEZONE=Europe/Paris

Etape 4. Création du fichier docker-compose.yml

Ensuite, il faut créer le fichier “docker-compose.yml” dans ce répertoire et y incorporer ce contenu, il comporte les différents services du serveur : MongoDB, Elasticsearch et Graylog.
Une fois le fichier docker-compose.yml créé, il faut renseigner les différentes sections correspondantes. Les versions de mongoDB et Elasticsearch dans ce fichier sont anciennes, les paramètres risquent d’être légèrement différents en fonction de la version utilisée.

```
version: '3.8'
services:
  mongo:
    image: mongo:4.4
    container_name: graylog-mongo
    restart: unless-stopped
    networks:
      - graylog
    volumes:
      - mongo_data:/data/db
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.10.2
    container_name: graylog-elasticsearch
    environment:
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
      - "discovery.type=single-node"
    ulimits:
      memlock:
        soft: -1
        hard: -1
    restart: unless-stopped
    networks:
      - graylog
    volumes:
      - es_data:/usr/share/elasticsearch/data
  graylog:
    image: graylog/graylog:4.3-jre11
    container_name: graylog-server
    env_file:
      - .env
    environment:
      - GRAYLOG_HTTP_BIND_ADDRESS=0.0.0.0:9000
      - GRAYLOG_HTTP_EXTERNAL_URI=http://172.16.0.101:9000/
    depends_on:
      - mongo
      - elasticsearch
    restart: unless-stopped
    ports:
      - "9000:9000"
      - "5140:5140/udp"
      - "5140:5140/tcp"
      - "12201:12201/udp"
      - "12201:12201/tcp"
    networks:
      - graylog
    volumes:
      - graylog_config:/usr/share/graylog/data/config
      - graylog_journal:/usr/share/graylog/data/journal
```

Dans ce fichier, des volumes sont déclarés pour chaque service, assurant ainsi la persistance des données entre les redémarrages de Docker.

Tous les conteneurs partagent un réseau Docker nommé “graylog” pour communiquer facilement entre eux sans exposer tous les ports.

Le redémarrage automatique est activé avec “restart : unless-stopped”.

Dans la section Graylog est renseignée le port d’écoute Graylog ainsi que l’IP externe pour accéder à son interface.

Les divers ports et protocoles correspondant sont aussi renseignés pour la remontée de logs.

Enfin, sur la page suivante, les volumes du fichier sont déclarés.

```
networks:
  graylog:
    driver: bridge

volumes:
  graylog_config:
  graylog_journal:
  mongo_data:
  es_data:
```

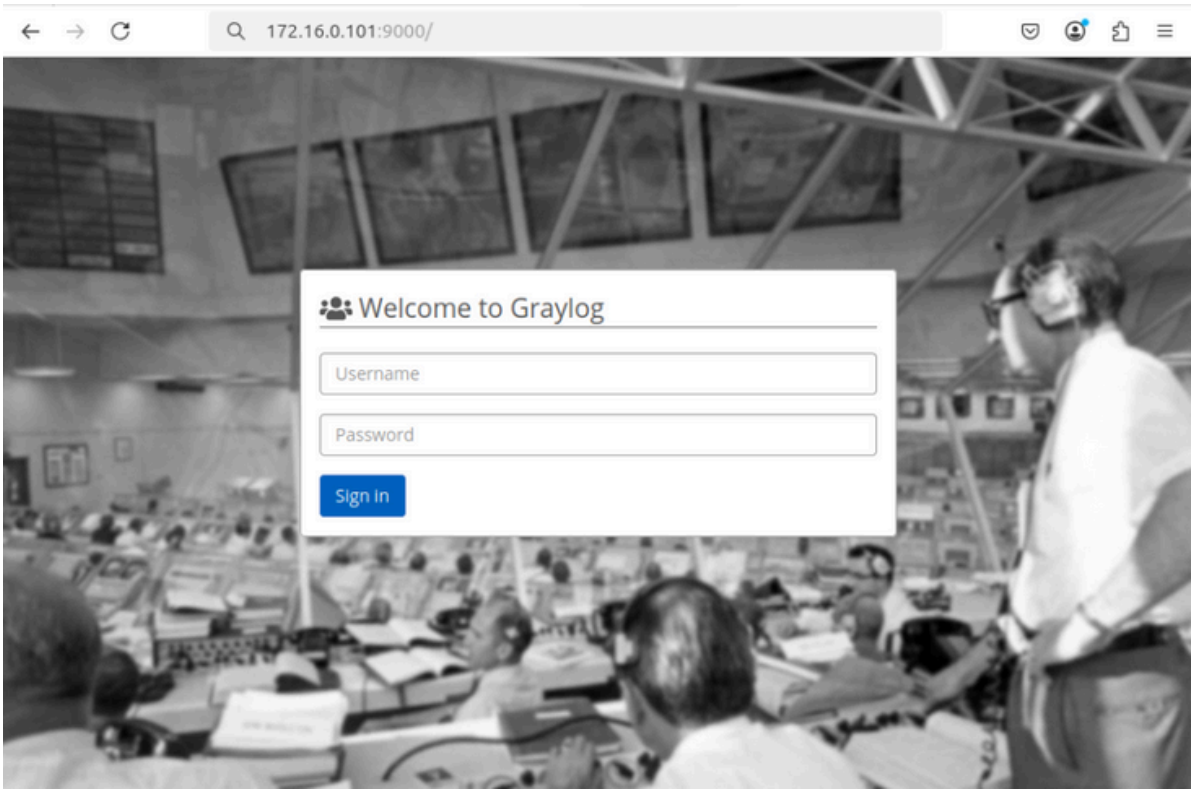
Etape 5. Lancement des conteneurs

Une fois la configuration terminée, enregistrer le fichier et lancer tout simplement les conteneurs avec la commande **docker-compose up -d.**

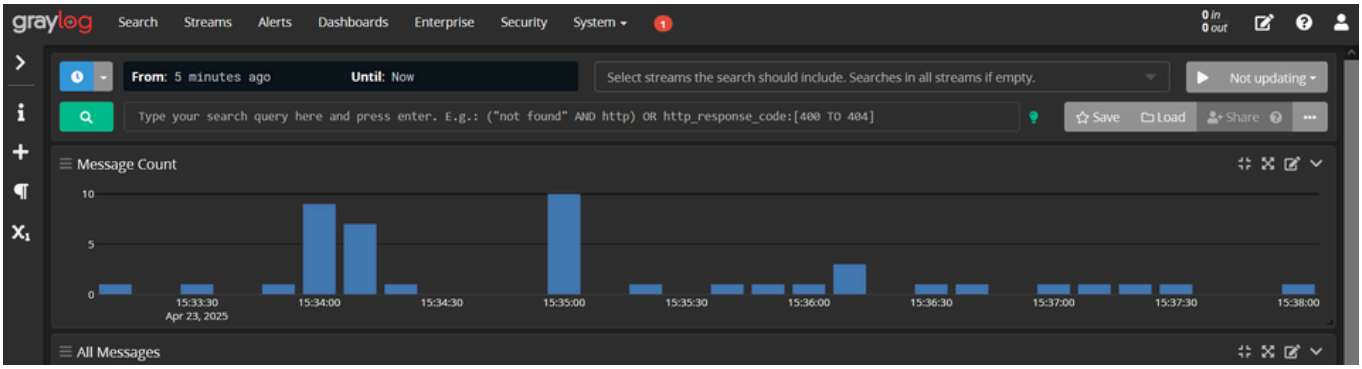
```
root@SRV-GRAYLOG:~# docker-compose up -d
Creating network "root_graylog" with driver "bridge"
Creating graylog-mongo         ... done
Creating graylog-elasticsearch ... done
Creating graylog-server        ... done
```

Etape 6. Accès à l'interface web de Graylog

Nous pouvons maintenant accéder à l'interface web du serveur, en rentrant dans le navigateur web l'adresse IP du serveur suivit du port 9000. Y rentrer l'identifiant admin ainsi que le mot de passe précédemment hashé.

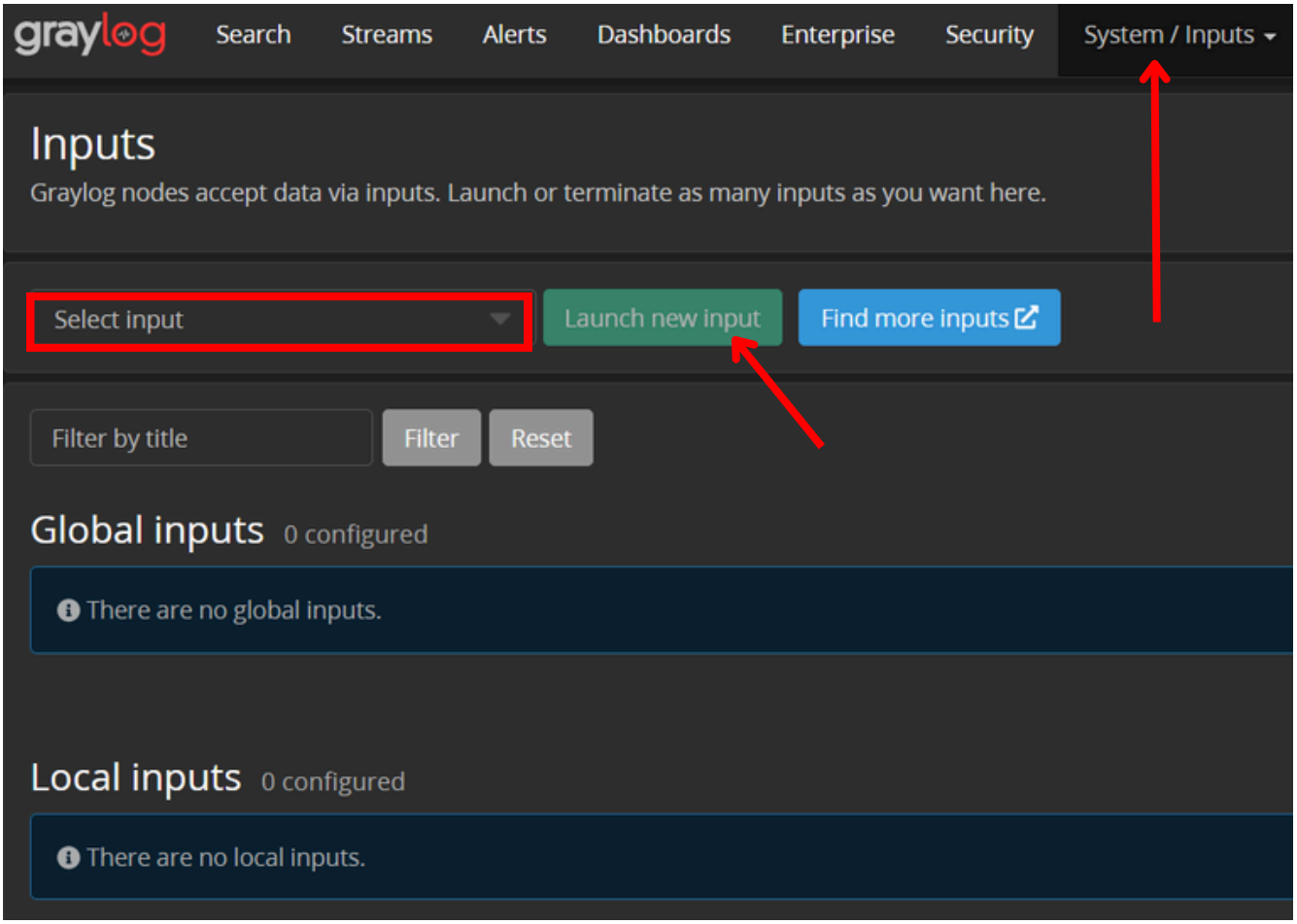


Après l'authentification, un guide de première utilisation est proposé à l'utilisateur, il est possible de le suivre ou de le passer. Enfin, nous arrivons sur l'interface du serveur et avons accès aux différentes fonctionnalités de Graylog.



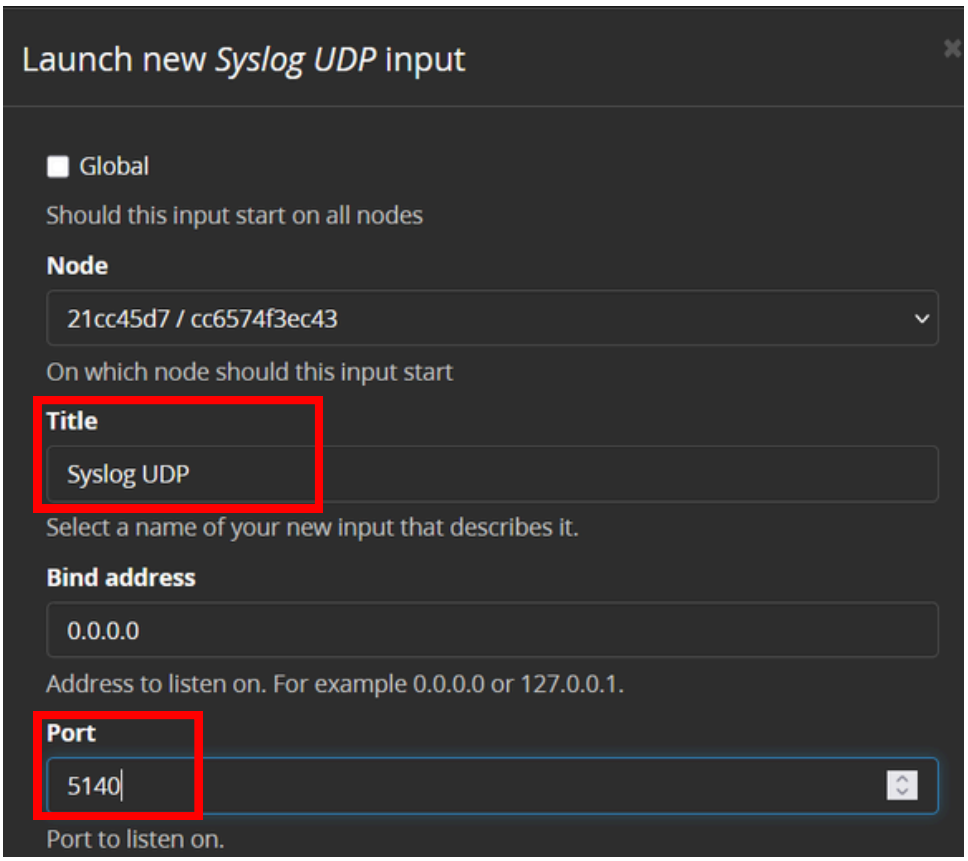
Etape 5. Configuration des entrées (input)

Maintenant, il faut paramétrer les différentes entrées pour faire remonter les logs.
Pour cela, sur le menu, cliquer sur System/inputs. Ensuite cliquer sur “Select input”.
Sélectionner “Syslog UDP” dans la liste déroulante puis cliquer sur “Launch new Input”.



Etape 5. Configuration des entrées (input)

Un menu de configuration va apparaître, le nommer Syslog UDP et y renseigner le port 5140 et cliquer sur “Save”.



On peut constater que notre input UDP pour les logs linux syslog a été enregistré.

Répéter l’opération pour les inputs Syslog TCP et GELF UDP/TCP (port 12201).