Graylog - Installation

FRELAUT
Raphaël
BTSSIO

Sommaire :

1. Introduction

- ## Présentation de Graylog

Graylog est un outil open-source pour la gestion centralisée des logs, conçu pour collecter, indexer et analyser des données machine en temps réel.

2. Objectifs

- ➢ Centraliser les logs provenant de différentes sources.
  - Routeur PFSense
  - ActiveDirectory
  - Debian BDD
  - Suricata
  - Machine Supervision
- ➢ Installation de l'outil
- ➢ Paramétrage de la centralisation des logs
- ➢ Configurer des forwarders de log sur les sources pour définir les alertes à remonter
- ➢ Définir les règles de détection d'anomalies
- ➢ Configurer les alertes pour prévenir l'équipe de supervision.

3. Installation

Commençons par une mise à jour du cache des paquets et l'installation d'outils nécessaires pour la suite des événements.

```
apt-get update
apt-get install curl lsb-release ca-certificates gnupg2 pwgen
```

A. Installation de mangoDB

Une fois que c'est fait, nous allons commencer l'installation de MongoDB. Téléchargez la clé GPG correspondante au dépôt MongoDB :

```
curl -fsSL https://www.mongodb.org/static/pgp/server-6.0.asc | gpg -o
/usr/share/keyrings/mongodb-server-6.0.gpg --dearmor
```

Puis, ajoutez le dépôt de MongoDB 6 sur la machine Debian 12 :

```
echo "deb [ signed-by=/usr/share/keyrings/mongodb-server-6.0.gpg]
http://repo.mongodb.org/apt/debian bullseye/mongodb-org/6.0 main" | tee
/etc/apt/sources.list.d/mongodb-org-6.0.list
```

Ensuite, nous allons mettre à jour le cache des paquets et tenter d'installer MongoDB :

```
apt-get update
apt-get install -y mongodb-org
```

L'installation de MongoDB ne peut pas être effectuée, car il manque une dépendance : **libssl1.1**. Nous allons devoir installer ce paquet manuellement avant de pouvoir poursuivre parce que Debian 12 ne l'a pas dans ses dépôts.

```
Les paquets suivants contiennent des dépendances non satisfaites :
 mongodb-org-mongos : Dépend: libssl1.1 (>= 1.1.1) mais il n'est pas
installable
 mongodb-org-server : Dépend: libssl1.1 (>= 1.1.1) mais il n'est pas
installable
E: Impossible de corriger les problèmes, des paquets défectueux sont en
mode « garder en l'état ».
```

Nous allons télécharger le paquet DEB nommé "**libssl1.1_1.1.1f-1ubuntu2.23_amd64.deb**" (version la plus récente) avec la commande **wget**, puis procéder à son installation via la commande **dpkg**. Ce qui donne les deux commandes suivantes :

```
wget
http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.1f-1ubun
tu2.23_amd64.deb
dpkg -i libssl1.1_1.1.1f-1ubuntu2.23_amd64.deb
```

Relancez l'installation de MongoDB :

```
apt-get install -y mongodb-org
```

Ensuite, relancez le service MongoDB et activez son démarrage automatique au lancement du serveur Debian.

```
sudo systemctl daemon-reload
sudo systemctl enable mongod.service
sudo systemctl restart mongod.service
sudo systemctl --type=service --state=active | grep mongod
```

MongoDB est installé, nous pouvons passer à l'installation du prochain composant.

B. Installation d'OpenSearch

Nous allons passer à l'installation d'OpenSearch sur le serveur. La commande suivante permet d'ajouter la clé de signature pour les paquets OpenSearch :

```
curl -o- https://artifacts.opensearch.org/publickeys/opensearch.pgp | gpg
--dearmor --batch --yes -o /usr/share/keyrings/opensearch-keyring
```

Puis, ajoutez le dépôt OpenSearch pour que nous puissions télécharger le paquet avec **apt** par la suite :

```
echo "deb [signed-by=/usr/share/keyrings/opensearch-keyring]
https://artifacts.opensearch.org/releases/bundle/opensearch/2.x/apt stable
main" | tee /etc/apt/sources.list.d/opensearch-2.x.list
```

Mettez à jour votre cache de paquets :

```
apt-get update
```

Puis, **installez OpenSearch** en prenant soin de **définir le mot de passe par défaut pour le compte Admin** de votre instance. Ici, le mot de passe est "**IT-Connect2024!**", mais remplacez cette valeur par un mot de passe robuste. **Évitez les mots de passe faibles** du style "**P@ssword123**" et utilisez au moins **8 caractères** avec au moins un caractère de chaque type (minuscule, majuscule, chiffre et caractère spécial), sinon il y aura une erreur à la fin de l'installation. **C'est un prérequis depuis OpenSearch 2.12.**

```
env OPENSEARCH_INITIAL_ADMIN_PASSWORD=IT-Connect2024! apt-get install
opensearch
```

Patientez pendant l'installation...

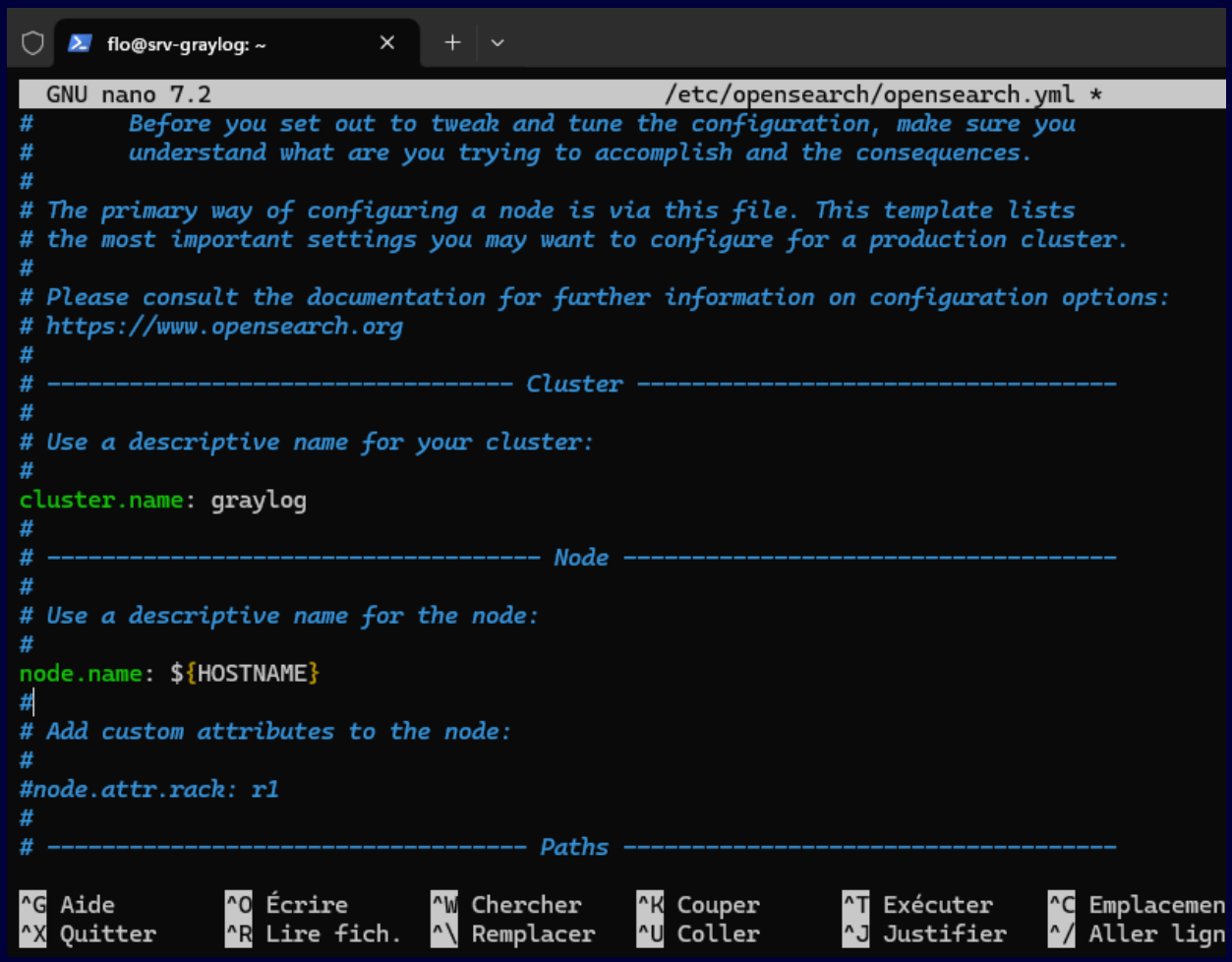
Quand c'est terminé, prenez le temps d'effectuer la configuration minimale. Ouvrez le fichier de configuration au format YAML :

```
nano /etc/opensearch/opensearch.yml
```

Lorsque le fichier est ouvert, configurez les options suivantes :

```
cluster.name: graylog
node.name: ${HOSTNAME}
path.data: /var/lib/opensearch
path.logs: /var/log/opensearch
discovery.type: single-node
network.host: 127.0.0.1
action.auto_create_index: false
plugins.security.disabled: true
```

Cette configuration OpenSearch est destinée à configurer un nœud unique.



C. Configuration JAVA (JVM)

Vous devez configurer Java Virtual Machine utilisé par OpenSearch afin d'ajuster la quantité de mémoire que peut utiliser ce service. Éditez le fichier de configuration suivant :

```
nano /etc/opensearch/jvm.options
```

Avec la configuration déployée ici, **OpenSearch démarrera avec une mémoire allouée de 2 Go et pourra atteindre jusqu'à 2 Go**, il n'y aura donc pas de variation de mémoire pendant le fonctionnement. Ici, la configuration tient compte du fait que la machine virtuelle dispose d'un total de **4 Go de RAM**. Les deux paramètres doivent avoir la même valeur. Ceci implique de remplacer ces lignes :

```
-Xms1g
-Xmx1g
```

Par ces lignes :

```
-Xms2g
-Xmx2g
```

Fermez ce fichier après l'avoir enregistré.

En complément, nous devons vérifier la configuration du paramètre "**max_map_count**" au niveau du noyau Linux. Il définit la limite des zones de mémoire mappées par processus, afin de répondre aux besoins de notre application. **OpenSearch**, au même titre que **Elasticsearch**, recommande de **fixer cette valeur à "262144" pour éviter des erreurs liées à la gestion de la mémoire**.

En principe, sur une machine Debian 12 fraîchement installée, la valeur est déjà correcte. Mais, nous allons le vérifier. Exécutez cette commande :

```
cat /proc/sys/vm/max_map_count
```

Si vous obtenez une valeur différente de "**262144**", exécutez la commande suivante, sinon ce n'est pas nécessaire.

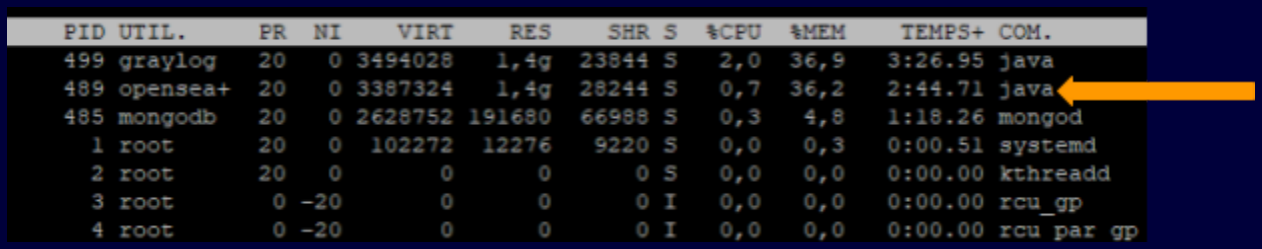
```
sudo sysctl -w vm.max_map_count=262144
```

Enfin, activez le démarrage automatique d'OpenSearch et lancez le service associé.

```
sudo systemctl daemon-reload
sudo systemctl enable opensearch
sudo systemctl restart opensearch
```

Si vous affichez l'état de votre système, vous devriez voir un processus Java avec 2 Go de RAM.

```
top
```



PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
499	graylog	20	0	3494028	1,4g	23844	S	2,0	36,9	3:26.95	java
489	opensea+	20	0	3387324	1,4g	28244	S	0,7	36,2	2:44.71	java
485	mongodb	20	0	2628752	191680	66988	S	0,3	4,8	1:18.26	mongod
1	root	20	0	102272	12276	9220	S	0,0	0,3	0:00.51	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp

D. Installation de Graylog

Pour effectuer l'**installation de Graylog 6.1** dans sa dernière version, exécutez les 4 commandes suivantes afin de **télécharger et d'installer Graylog Server** :

```
wget
https://packages.graylog2.org/repo/packages/graylog-6.1-repository_latest.d
eb
dpkg -i graylog-6.1-repository_latest.deb
apt-get update
apt-get install graylog-server
```

Quand c'est fait, nous devons apporter des modifications à la configuration de Graylog avant de chercher à le lancer.

Commençons par configurer ces deux options :

- **password_secret** : ce paramètre sert à définir une clé utilisée par Graylog pour sécuriser le stockage des mots de passe utilisateurs (dans l'esprit d'une clé de salage). Cette clé doit être **unique** et **aléatoire**.
- **root_password_sha2** : ce paramètre correspond au mot de passe de l'administrateur par défaut dans Graylog. Il est stocké sous forme d'un hash SHA-256.

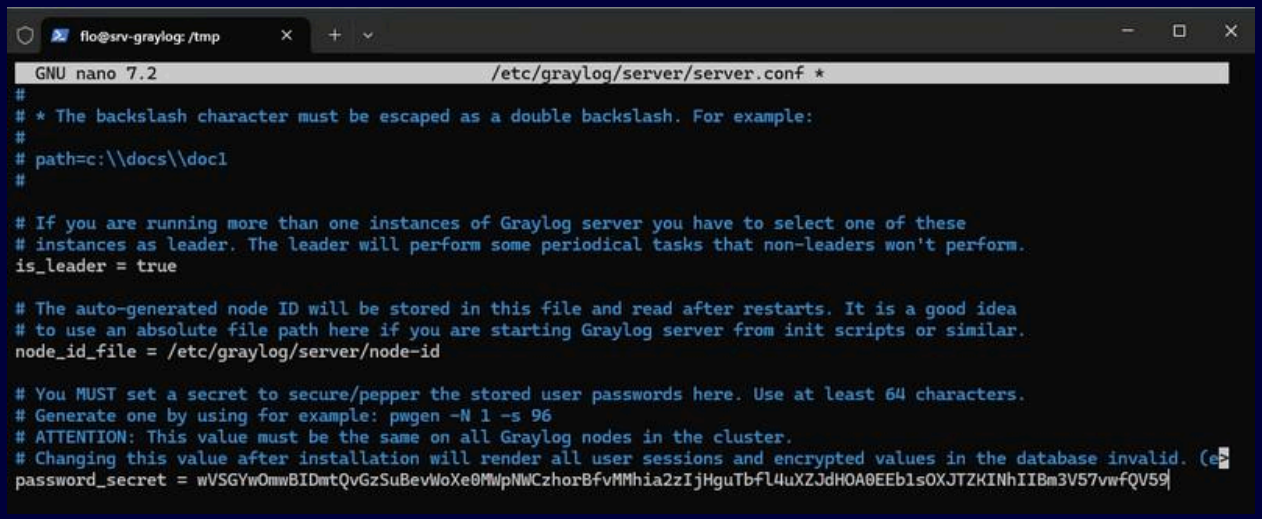
Nous allons commencer par générer une clé de 96 caractères pour le paramètre **password_secret** :

```
pwgen -N 1 -s 96
wVSGYwOmwBIDmtQvGzSuBevWoXe0MWpNWCzhorBfvMMhia2zIjHguTbfl4uXZJdHOA0EEb1sOXJ
TZKINhIIBm3V57vwfQV59
```

Copiez la valeur retournée, puis ouvrez le fichier de configuration de Graylog :

```
nano /etc/graylog/server/server.conf
```

Collez la clé au niveau du paramètre **password_secret**, comme ceci :



Enregistrez et fermez le fichier.

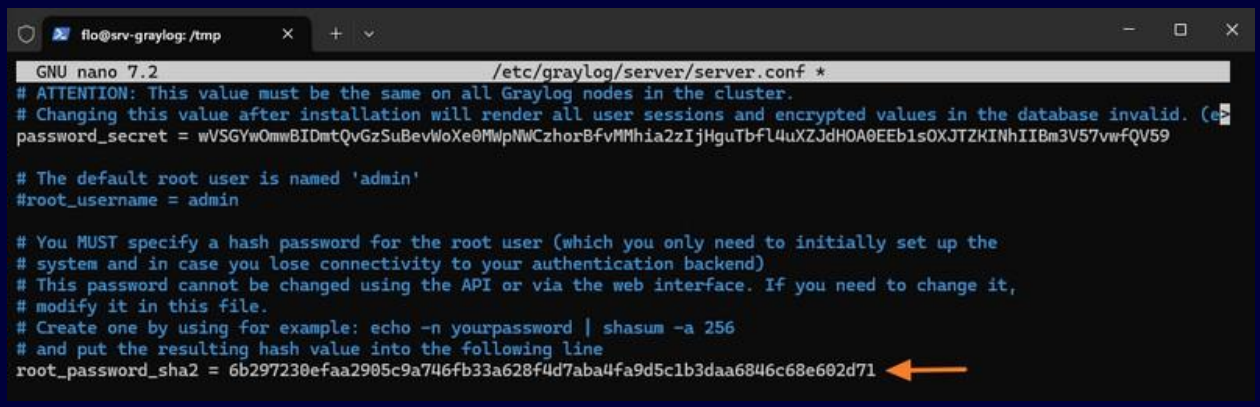
Ensuite, vous devez définir le mot de passe du compte "**admin**" créé par défaut. Dans le fichier de configuration, c'est le hash du mot de passe qui doit être stocké, ce qui implique de le calculer. L'exemple ci-dessous permet d'obtenir le hash du mot de passe "**PuitsDeLogs@**" : adaptez la valeur avec votre mot de passe.

```
echo -n "PuitsDeLogs@" | shasum -a 256
6b297230efaa2905c9a746fb33a628f4d7aba4fa9d5c1b3daa6846c68e602d71
```

Copiez la valeur obtenue en sortie. Ouvrez de nouveau le fichier de configuration de Graylog :

```
nano /etc/graylog/server/server.conf
```

Collez la valeur au niveau de l'option **root_password_sha2** comme ceci :



Profitez d'être dans le fichier de configuration pour configurer l'option nommée "**http_bind_address**". Indiquez "**0.0.0.0:9000**" pour que l'interface web de Graylog soit accessible sur le port **9000**, via n'importe quelle adresse IP du serveur.

```
###############
# HTTP settings
###############

#### HTTP bind address
#
# The network interface used by the Graylog HTTP interface.
#
# This network interface must be accessible by all Graylog nodes in the cluster and by all clients
# using the Graylog web interface.
#
# If the port is omitted, Graylog will use port 9000 by default.
#
# Default: 127.0.0.1:9000
http_bind_address = 0.0.0.0:9000
#http_bind_address = [2001:db8::1]:9000
```

Puis, configurez l'option "**elasticsearch_hosts**" avec la valeur "**http://127.0.0.1:9200**" pour déclarer notre instance locale OpenSearch. Ceci est nécessaire, car nous n'utilisons pas de **Graylog Data Node**. Et sans cette option, il ne sera pas possible d'aller plus loin...

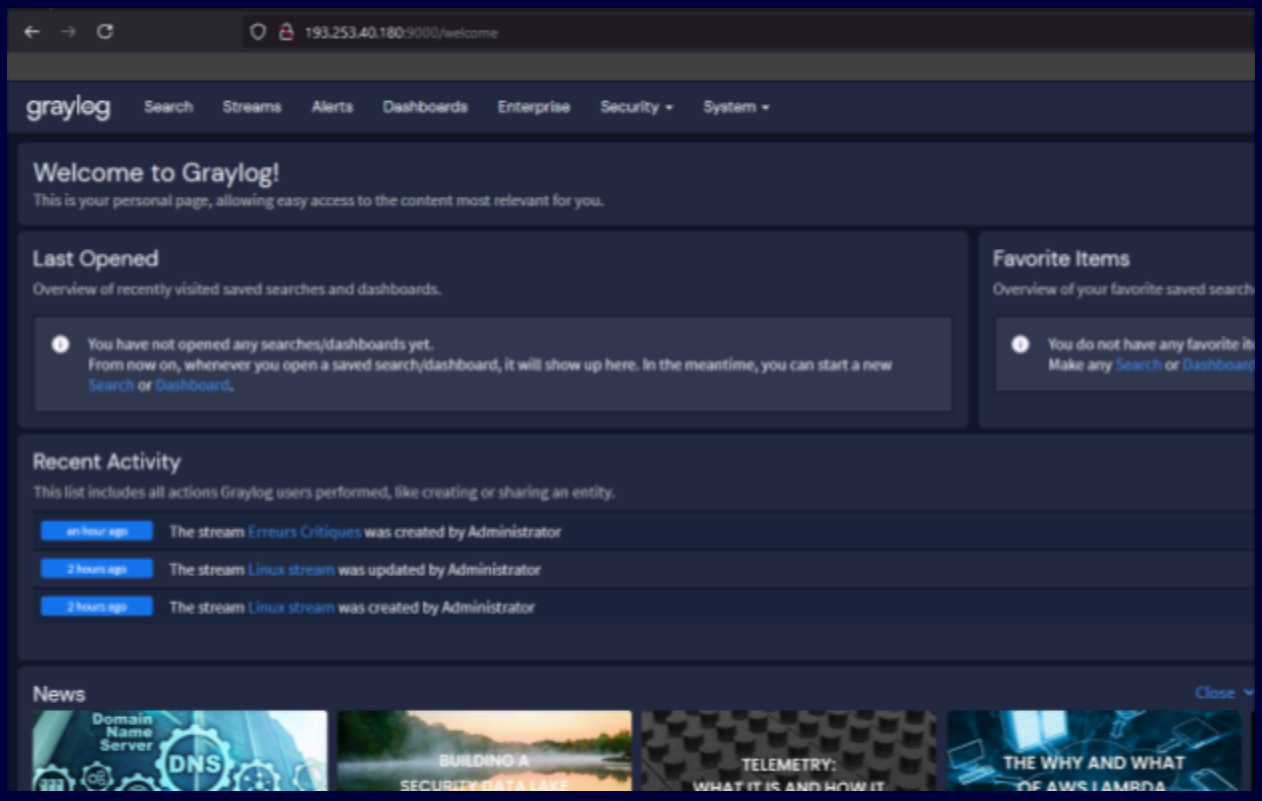
```
# List of Elasticsearch hosts Graylog should connect to.
# Need to be specified as a comma-separated list of valid URIs for the http ports of your elasticsearch nodes.
# If one or more of your elasticsearch hosts require authentication, include the credentials in each node URI that
# requires authentication.
#
# Default: http://127.0.0.1:9200
elasticsearch_hosts = http://127.0.0.1:9200
```

Enregistrez et fermez le fichier.

Cette commande active Graylog pour qu'il démarre automatiquement au prochain démarrage et elle lance immédiatement le serveur Graylog.

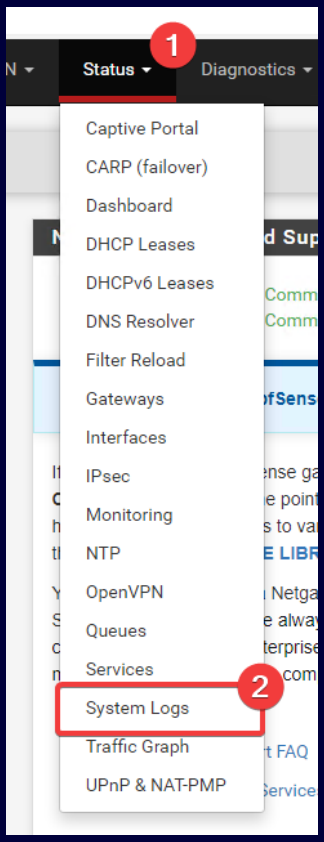
```
systemctl enable --now graylog-server
```

Une fois que c'est fait, tentez une connexion à Graylog à partir d'un navigateur. Indiquez l'adresse IP du serveur (ou son nom) et le port 9000.

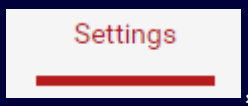


4. Envoie des logs

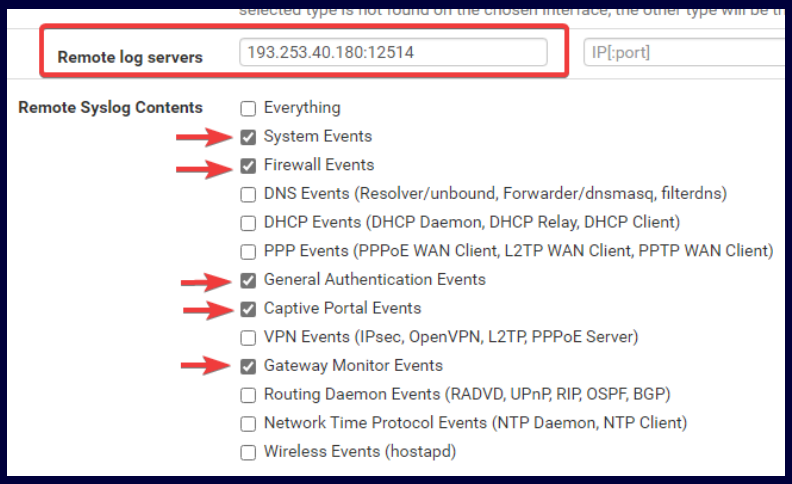
1. Depuis un routeur PFsense :



Se rendre sur l'interface web de pfsense, puis dans Status > System logs.

Ensuite, aller dans l'onglet settings 

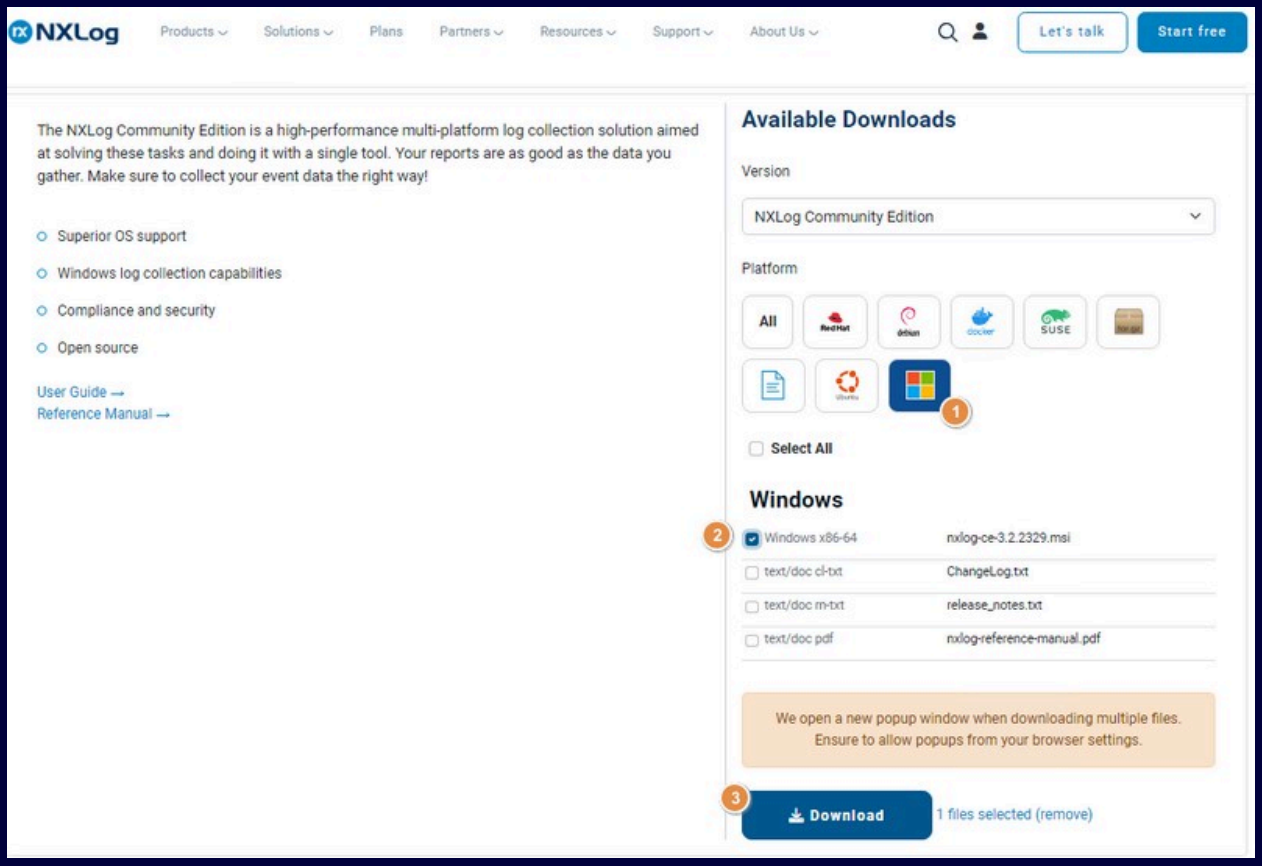
, a la fin de la page, compléter la configuration suivante :



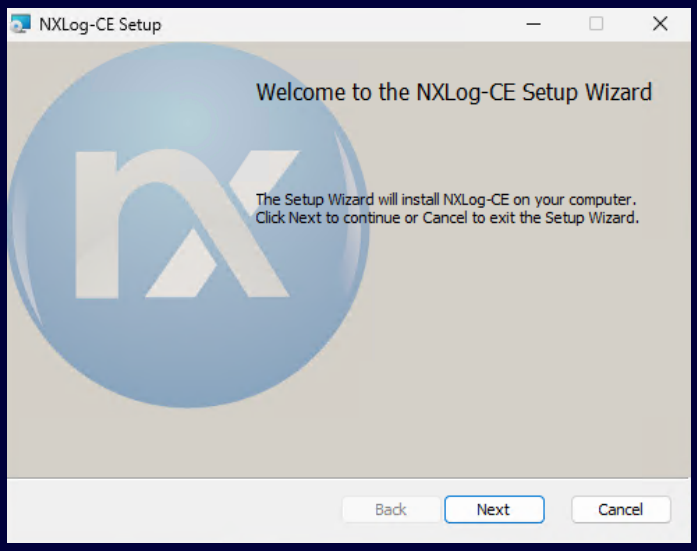
2. Depuis une machine Windows

> Cliquez sur ce nxlog : https://nxlog.co/downloads/

Puis, sélectionnez "Windows", cochez la case "Windowz x86-64" et lancez le téléchargement.



Sur votre machine Windows, lancez l'installation via le package "nxlog-ce-3.2.2329.msi". Suivez l'assistant et effectuez l'installation... La configuration s'effectuera par la suite. Puisqu'il s'agit d'un package MSI, nous pourrions le déployer facilement sur un ensemble de machines pour automatiser l'installation.



NXLog étant installé sur la machine, nous pouvons éditer son fichier de configuration situé à l'emplacement suivant :

> `C:\Program Files\nxlog\conf\nxlog.conf`

En complément de la configuration déjà présente dans le fichier "nxlog.conf", vous devez ajouter ces lignes à la fin :

```
# Récupérer les journaux de l'observateur d'événements
<Input in>
    Module      im_msvistalog
</Input>

# Déclarer le serveur Graylog (selon input)
<Extension gelf>
    Module        xm_gelf
</Extension>

<Output graylog_udp>
    Module        om_udp
    Host          192.168.10.220
    Port      12201
    OutputType    GELF_UDP
</Output>

# Routage des flux in vers out
 <Route 1>
     Path        in => graylog_udp
 </Route>
```

Quelques explications s'imposent pour vous aider à comprendre :

- im_msvistalog : il s'agit du module déclaré comme entrée (Input) pour récupérer les journaux dans l'Observateur d'événements de Windows, compatible à partir de Windows Server 2008 et Windows Vista. Il est toujours compatible avec les dernières versions, à savoir Windows 11 et Windows Server 2025. Pour les versions antérieures à Windows Server 2008, utilisez le module "im_mseventlog".
- om_udp : il s'agit du module déclaré comme sortie (Output graylog_udp). Dans ce bloc, vous devez modifier l'adresse IP, car elle correspond au serveur Graylog (192.168.10.220) et éventuellement le port. Nous utilisons le type de sortie "GELF_UDP" pour rester cohérent vis-à-vis de l'Input déclaré dans Graylog.

- ➢ Route 1 : il s'agit d'une règle « de routage » dans NXLog pour prendre ce qui correspond à l'Input "in" (les logs Windows) et les envoyer vers la sortie "graylog_udp", à savoir notre Graylog.

Sauvegardez les changements et redémarrez le service NXLog à partir d'une console PowerShell ouverte en tant qu'administrateur (ou via la console Services).

```
Restart-Service nxlog
```

Aides Complémetaires:

https://www.it-connect.fr/tuto-graylog-sur-debian-centraliser-et-analyser-logs/#I_Presentation
Comment envoyer les logs Windows vers Graylog avec NXLog ?