

## Documentation technique - Mise en place de Graylog

### Objectif

L'objectif de cette documentation est la mise en place de Graylog, un logiciel permettant de centraliser, d'indexer et de faciliter la lecture des logs. Il est capable de traiter les logs de plusieurs sources différentes, tout en créant des alertes.

Dans le cadre du projet Health North, cela va nous permettre d'identifier et de résoudre plus efficacement les incidents sur des services tels que le reverse proxy, les serveurs Web ou encore l'Active Directory.

### Préparation de la machine virtuelle

Pour commencer, il faut d'abord installer Graylog. Pour cela, nous avons besoin de créer une machine virtuelle sous Ubuntu, avec un serveur équipé au minimum de 8 Go de RAM. Cette machine doit être positionnée dans le VLAN 170 avec une adresse IP fixe qui est : 10.170.0.11.

### Préparation et installation de Graylog

Dans un premier temps, nous devons installer OpenJDK 17, qui est requis pour le bon fonctionnement de Graylog.

Afin que Graylog fonctionne correctement, plusieurs composants sont nécessaires. Tout d'abord MongoDB, qu'il faut ajouter à la machine virtuelle, installer puis démarrer grâce aux commandes suivantes :

```
sudo apt install -y gnupg curl
```

```
curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
```

```
sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg --dearmor
```

```
echo "deb [ arch=amd64, arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ]  
https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse" | \
```

```
sudo tee /etc/apt/sources.list.d/mongodb-org-8.0.list
```

```
sudo apt update
```

```
sudo apt install -y mongodb-org
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable mongod
```

```
sudo systemctl start mongod
```

```
sudo systemctl status mongod
```

```
Last login: Sun Apr 19 14:16:48 2026 from 10.100.0.75
emilien@hn-gdlp-gl01:~$ sudo systemctl status mongod
[sudo] password for emilien:
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; preset: e>
   Active: active (running) since Sun 2026-04-19 21:11:28 UTC; 1 week 0 days >
     Docs: https://docs.mongodb.org/manual
   Main PID: 561266 (mongod)
  Memory: 169.5M (peak: 294.1M swap: 71.0M swap peak: 72.8M)
     CPU: 2h 39min 2.896s
   CGroup: /system.slice/mongod.service
           └─561266 /usr/bin/mongod --config /etc/mongod.conf

avril 19 21:11:28 hn-gdlp-gl01 systemd[1]: Started mongod.service - MongoDB Dat>
avril 19 21:11:28 hn-gdlp-gl01 mongod[561266]: {"t":{"$date":"2026-04-19T21:11:>
lines 1-12/12 (END)
```

Dans un second temps, nous avons besoin d'installer OpenSearch qui est utilisé par Graylog afin d'indexer les logs reçus. Pour cela, nous devons ajouter le dépôt OpenSearch à Ubuntu, puis lancer l'installation. Pour finir, nous devons configurer OpenSearch à l'aide des commandes suivantes :

```
curl -o- https://artifacts.opensearch.org/publickeys/opensearch.pgp | \
```

```
sudo gpg --dearmor --batch --yes -o /usr/share/keyrings/opensearch-keyring
```

```
Echo "deb [signed-by=/usr/share/keyrings/opensearch-keyring]
```

```
https://artifacts.opensearch.org/releases/bundle/opensearch/2.x/apt stable main" | \
```

```
sudo tee /etc/apt/sources.list.d/opensearch-2.x.list
```

```
sudo apt update
```

```
sudo OPENSEARCH_INITIAL_ADMIN_PASSWORD='HNOpenSearch2026!' apt install -y
opensearch
```

Une fois l'installation réalisée, nous pouvons donc configurer OpenSearch. Pour cela, il faut ouvrir le fichier de configuration suivant :

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

À l'intérieur de ce fichier, nous pouvons copier ceci :

```
cluster.name: graylog
node.name: HN-GDLP-GL01
path.data: /var/lib/opensearch
path.logs: /var/log/opensearch
network.host: 127.0.0.1
http.port: 9200
discovery.type: single-node
action.auto_create_index: false
plugins.security.disabled: true
```

Ensuite, nous pouvons configurer la mémoire vive utilisée par Java, dans notre cas 1 Go.

Pour cela il faut accéder au fichier de configuration de Java :

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

Une fois ouvert, nous pouvons ajouter les lignes suivantes :

```
-Xms1g
-Xmx1g
```

Pour configurer les paramètres système, nous devons utiliser ces commandes :

```
sudo sysctl -w vm.max_map_count=262144
echo "vm.max_map_count=262144" | sudo tee -a /etc/sysctl.conf
```

Dès que les paramètres sont modifiés, nous pouvons redémarrer OpenSearch afin d'appliquer la configuration.

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable opensearch
```

```
sudo systemctl start opensearch
```

```
sudo systemctl status opensearch
```

```
emilien@hn-gdlp-g101:~$ sudo systemctl status opensearch
● opensearch.service - OpenSearch
   Loaded: loaded (/usr/lib/systemd/system/opensearch.service; enabled; prese>
   Active: active (running) since Sat 2026-04-18 08:32:03 UTC; 1 week 1 day a>
     Docs: https://opensearch.org/
   Main PID: 11570 (java)
     Tasks: 91 (limit: 9432)
  Memory: 3.7G (peak: 6.6G swap: 506.5M swap peak: 675.8M)
     CPU: 6h 42min 20.704s
   CGroup: /system.slice/opensearch.service
           └─11570 /usr/share/opensearch/jdk/bin/java -Xshare:auto -Dopensear>

avril 18 08:32:04 hn-gdlp-g101 systemd-entrypoint[11570]:          at org.opense>
avril 18 08:32:04 hn-gdlp-g101 systemd-entrypoint[11570]:          at org.opense>
avril 18 08:32:04 hn-gdlp-g101 systemd-entrypoint[11570]:          at org.opense>
avril 18 08:32:04 hn-gdlp-g101 systemd-entrypoint[11570]:          at org.opense>
avril 18 08:32:04 hn-gdlp-g101 systemd-entrypoint[11570]:          at org.opense>
avril 18 08:32:04 hn-gdlp-g101 systemd-entrypoint[11570]:          at org.opense>
avril 18 08:32:04 hn-gdlp-g101 systemd-entrypoint[11570]:          at java.base/>
avril 18 08:32:04 hn-gdlp-g101 systemd-entrypoint[11570]:          at java.base/>
avril 18 08:32:04 hn-gdlp-g101 systemd-entrypoint[11570]:          at java.base/>
avril 18 08:32:04 hn-gdlp-g101 systemd-entrypoint[11570]: For complete error de>
lines 1-21/21 (END)
```

Nous pouvons ensuite tester le bon fonctionnement d'OpenSearch à l'aide de la commande suivante : <http://127.0.0.1:9200>

```
emilien@hn-gd1p-g101:~$ curl http://127.0.0.1:9200
{
  "name" : "node-1",
  "cluster_name" : "graylog",
  "cluster_uuid" : "GnVT4FV-QDGBX7r51KVLXg",
  "version" : {
    "distribution" : "opensearch",
    "number" : "2.19.5",
    "build_type" : "deb",
    "build_hash" : "688434c5163e9b107f339df25b4e98a96c10ddc6",
    "build_date" : "2026-03-07T02:22:25.894998559Z",
    "build_snapshot" : false,
    "lucene_version" : "9.12.3",
    "minimum_wire_compatibility_version" : "7.10.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "The OpenSearch Project: https://opensearch.org/"
}
emilien@hn-gd1p-g101:~$ █
```

Ensuite, nous pouvons installer Graylog. Pour cela, nous devons ajouter le dépôt Graylog, puis lancer l'installation en suivant les commandes comme ceci :

```
wget https://packages.graylog2.org/repo/packages/graylog-7.0-repository_latest.deb
```

```
sudo dpkg -i graylog-7.0-repository_latest.deb
```

```
sudo apt update
```

```
sudo apt install -y graylog-server
```

Une fois l'installation terminée, nous pouvons accéder au fichier de configuration du serveur afin d'y ajouter les paramètres de sécurité. Nous devons définir un mot de passe fort, puis générer le hash avec ces commandes :

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

Dans ce fichier, nous devons renseigner :

```
password_secret = Le mots de passe ici
```

Puis générer le hash :

```
echo -n "mots de passe ici" | sha256sum
```

Puis remplir dans le fichier de configuration :

```
root_password_sha2 = HASH_SHA256_ICI
```

Puis nous pouvons ensuite configurer l'accès à Graylog ainsi que les différents services que nous avons installés plus tôt :

```
http_bind_address = 0.0.0.0:9000
```

```
http_external_uri = http://10.170.0.11:9000/
```

```
elasticsearch_hosts = http://127.0.0.1:9200
```

```
mongodb_uri = mongodb://localhost:27017/graylog
```

Puis nous pouvons démarrer le service Graylog à l'aide des commandes suivantes :

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable graylog-server
```

```
sudo systemctl start graylog-server
```

```
sudo systemctl status graylog-server
```

```
emilien@hn-gd1p-g101:~$ pwgen -N 1 -s 96
Command 'pwgen' not found, but can be installed with:
sudo apt install pwgen
emilien@hn-gd1p-g101:~$ sudo systemctl status graylog-server
● graylog-server.service - Graylog server
   Loaded: loaded (/usr/lib/systemd/system/graylog-server.service; enabled; p>
   Active: active (running) since Sun 2026-04-19 21:11:29 UTC; 1 week 0 days >
     Docs: http://docs.graylog.org/
   Main PID: 561333 (graylog-server)
    Tasks: 298 (limit: 9432)
  Memory: 1.4G (peak: 1.5G swap: 381.9M swap peak: 822.6M)
     CPU: 5h 2min 14.643s
   CGroup: /system.slice/graylog-server.service
           └─561333 /bin/sh /usr/share/graylog-server/bin/graylog-server
             └─561334 /usr/share/graylog-server/jvm/bin/java -Xms1g -Xmx1g -ser>

avril 19 21:11:29 hn-gd1p-g101 systemd[1]: Started graylog-server.service - Gra>
lines 1-13/13 (END)
```

## Configuration de Graylog

Nous pouvons nous connecter à l'interface web via l'adresse suivante :

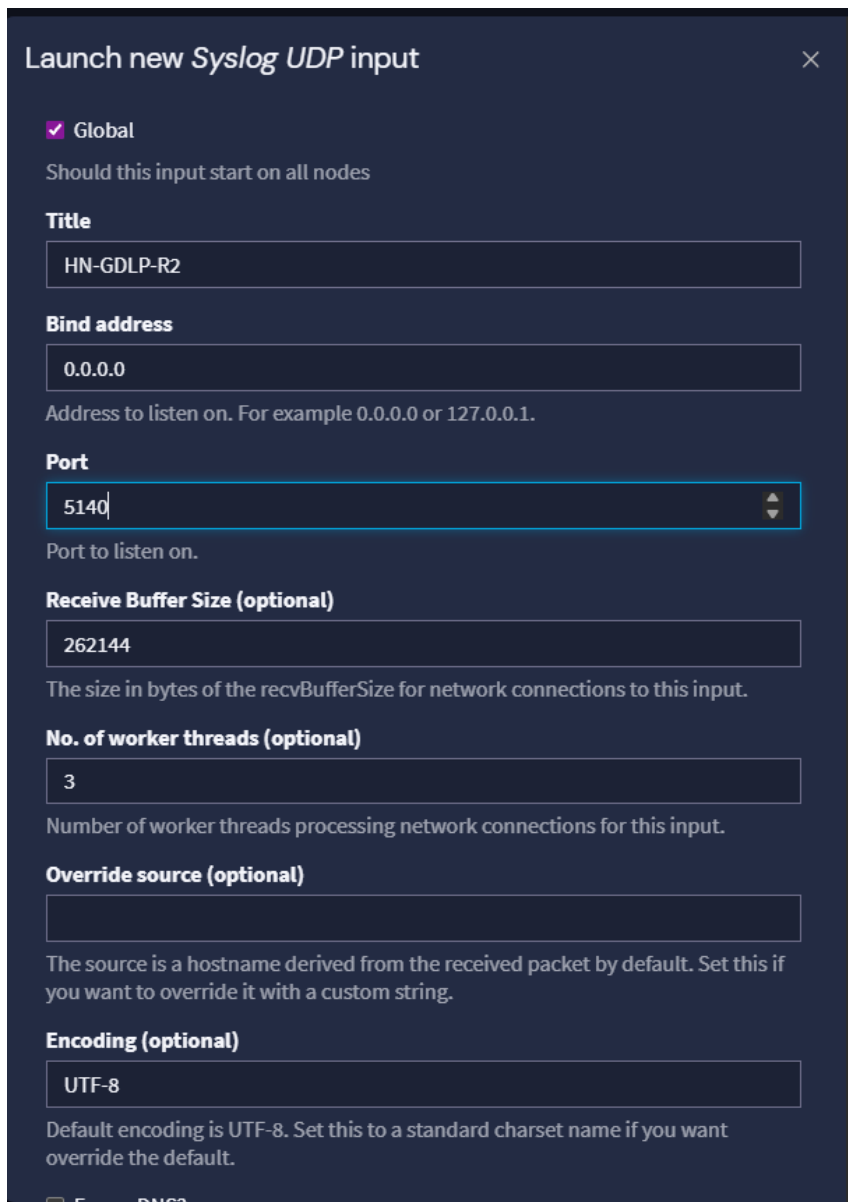
<http://10.170.0.11:9000>

L'identifiant est **admin**, et le mot de passe est celui défini précédemment.

Ensuite, nous allons pouvoir créer un *input*. Dans Graylog, un input correspond à la porte d'entrée des logs, c'est par ce biais où ils vont entrer dans Graylog avant d'être indexés.

Dans la section «Inputs», nous allons pouvoir sélectionner Syslog UDP afin de recevoir les logs provenant de pfSense.

Nous pouvons ensuite attribuer un nom à cet input, ainsi que définir le port utilisé.



Launch new Syslog UDP input

Global  
Should this input start on all nodes

**Title**  
HN-GDLP-R2

**Bind address**  
0.0.0.0  
Address to listen on. For example 0.0.0.0 or 127.0.0.1.

**Port**  
5140  
Port to listen on.

**Receive Buffer Size (optional)**  
262144  
The size in bytes of the recvBufferSize for network connections to this input.

**No. of worker threads (optional)**  
3  
Number of worker threads processing network connections for this input.

**Override source (optional)**  
  
The source is a hostname derived from the received packet by default. Set this if you want to override it with a custom string.

**Encoding (optional)**  
UTF-8  
Default encoding is UTF-8. Set this to a standard charset name if you want override the default.

Ensuite, nous allons pouvoir configurer l'envoi des logs depuis pfSense. Pour cela, nous devons nous connecter à l'interface de configuration de pfSense, puis aller dans « System Logs » et « Settings », et configurer le remote logging en cochant l'option permettant d'envoyer l'intégralité des logs.

Status / System Logs / Settings ?

System Firewall DHCP Authentication IPsec PPP PPPoE/L2TP Server OpenVPN NTP Packages Settings

### General Logging Options

**Log Message Format**   
The format of syslog messages written to disk locally and sent to remote syslog servers (if enabled). Changing this value will only affect new log messages.

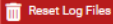
**Forward/Reverse Display**  Show log entries in reverse order (newest entries on top)

**GUI Log Entries**   
This is only the number of log entries displayed in the GUI. It does not affect how many entries are contained in the actual log files.

**Raw Logs**  Show raw filter logs  
If this is checked, filter logs are shown as generated by the packet filter, without any formatting. This will reveal more detailed information, but it is more difficult to read.

**Where to show rule descriptions**   
Show the applied rule description below or in the firewall log rows. Displaying rule descriptions for all lines in the log might affect performance with large rule sets.

**Local Logging**  Disable writing log files to the local disk  
WARNING: This will also disable Login Protection!

**Reset Log Files**   
Clears all local log files and reinitializes them as empty logs. This also restarts the DHCP daemon. Use the Save button first if any setting changes have been made.

### Logging Preferences

Default firewall "block" rules  
Log packets that are **blocked** by the implicit default block rule.

Default firewall "pass" rules  
Log packets that are **allowed** by the implicit default pass rule. Note: Packets with IP options are not affected by this option and **are logged by default**.

Default "Bogon Networks" block rules  
Log packets that are **blocked** by the assigned interface option "Block bogon networks".

Default "Private Networks" block rules  
Log packets that are **blocked** by the assigned interface option "Block private networks and loopback addresses".

Default "IPv4 link-local" block rules  
Log packets that are **blocked** by the default "Block IPv4 link-local" rules.

The number of log files to keep before the oldest copy is removed on rotation.

### Remote Logging Options

**Enable Remote Logging**  Send log messages to remote syslog server

**Source Address**  This option will allow the logging daemon to bind to a single IP address, rather than all IP addresses. If a single IP is picked, remote syslog servers must all be of that IP type. To mix IPv4 and IPv6 remote syslog servers, bind to all interfaces.  
NOTE: If an IP address cannot be located on the chosen interface, the daemon will bind to all addresses.

**IP Protocol**  This option is only used when a non-default address is chosen as the source above. This option only expresses a preference; if an IP address of the selected type is not found on the chosen interface, the other type will be tried.

**Remote log servers**

**Remote Syslog Contents**  Everything

- System Events
- Firewall Events
- DNS Events (Resolver/unbound, Forwarder/dnsmasq, filterdns)
- DHCP Events (DHCP Daemon, DHCP Relay, DHCP Client)
- PPP Events (PPPoE WAN Client, L2TP WAN Client, PPTP WAN Client)
- General Authentication Events
- Captive Portal Events
- VPN Events (IPsec, OpenVPN, L2TP, PPPoE Server)
- Gateway Monitor Events
- Routing Daemon Events (RADVD, UPnP, RIP, OSPF, BGP)
- Network Time Protocol Events (NTP Daemon, NTP Client)
- Wireless Events (hostapd)

Syslog sends UDP datagrams to port 514 on the specified remote syslog server, unless another port is specified. Be sure to set syslogd on the remote server to accept syslog messages from pfSense.

Pour un serveur Linux, il est également nécessaire d'ajouter un input dans Graylog, puis d'installer rsyslog et de configurer l'envoi des logs comme ceci :

```
sudo apt update
```

```
sudo apt install -y rsyslog
```

```
sudo nano /etc/rsyslog.d/90-graylog.conf
```

. @10.170.0.11: le port que l'on a choisi

Puis, nous pouvons redémarrer rsyslog :

```
sudo systemctl restart rsyslog
```

```
sudo systemctl enable rsyslog
```

Pour un serveur Windows, nous pouvons utiliser NXLog, qui permet de centraliser les logs de la machine et de les envoyer vers Graylog.

Après installation, nous pouvons modifier le fichier de configuration et y entrer :

```
define ROOT C:\Program Files\nxlog
```

```
define CERTDIR %ROOT%\cert  
define CONFDIR %ROOT%\conf  
define LOGDIR %ROOT%\data
```

```
Moduledir %ROOT%\modules  
CacheDir %ROOT%\data  
Pidfile %ROOT%\data\nxlog.pid  
SpoolDir %ROOT%\data  
LogFile %ROOT%\data\nxlog.log
```

```
<Input eventlog>
```

```
    Module im_msvistalog
```

```
</Input>
```

```
<Output graylog>
```

```
    Module om_udp
```

```
    Host 10.170.0.11
```

```
    Port 1514
```

```
    Exec to_syslog_bsd();
```

```
</Output>
```

```
<Route route1>
```

```
    Path eventlog => graylog
```

```
</Route>
```

Une fois les trois différents inputs créés, nous pouvons créer des streams dans Graylog afin d'organiser et de filtrer les logs selon leur provenance ou leur type.

## Création des Stream

Les streams permettent de séparer les logs selon leur source.

Pour cela, nous devons nous rendre dans « Streams », et cliquer sur « Create Stream ».

Ensuite, il suffit de donner un nom à notre stream, et sélectionner l'index pfSense.

**Create stream** [X]

**Title**

Pfsense

A descriptive name of the new stream

**Description (Opt.)**

R1 Log

What kind of messages are routed into this stream?

**Index Set**

Pfsense [X] [v]

Messages that match this stream will be written to the configured index set.

Remove matches from 'Default Stream'

Don't assign messages that match this stream to the 'Default Stream'.

**Add Collaborator**

Search for a User or Team to add as collaborator on this stream.

Search for users and teams [v] Viewer [v] Add Collaborator

**Collaborators**

*i* This stream has no collaborators.

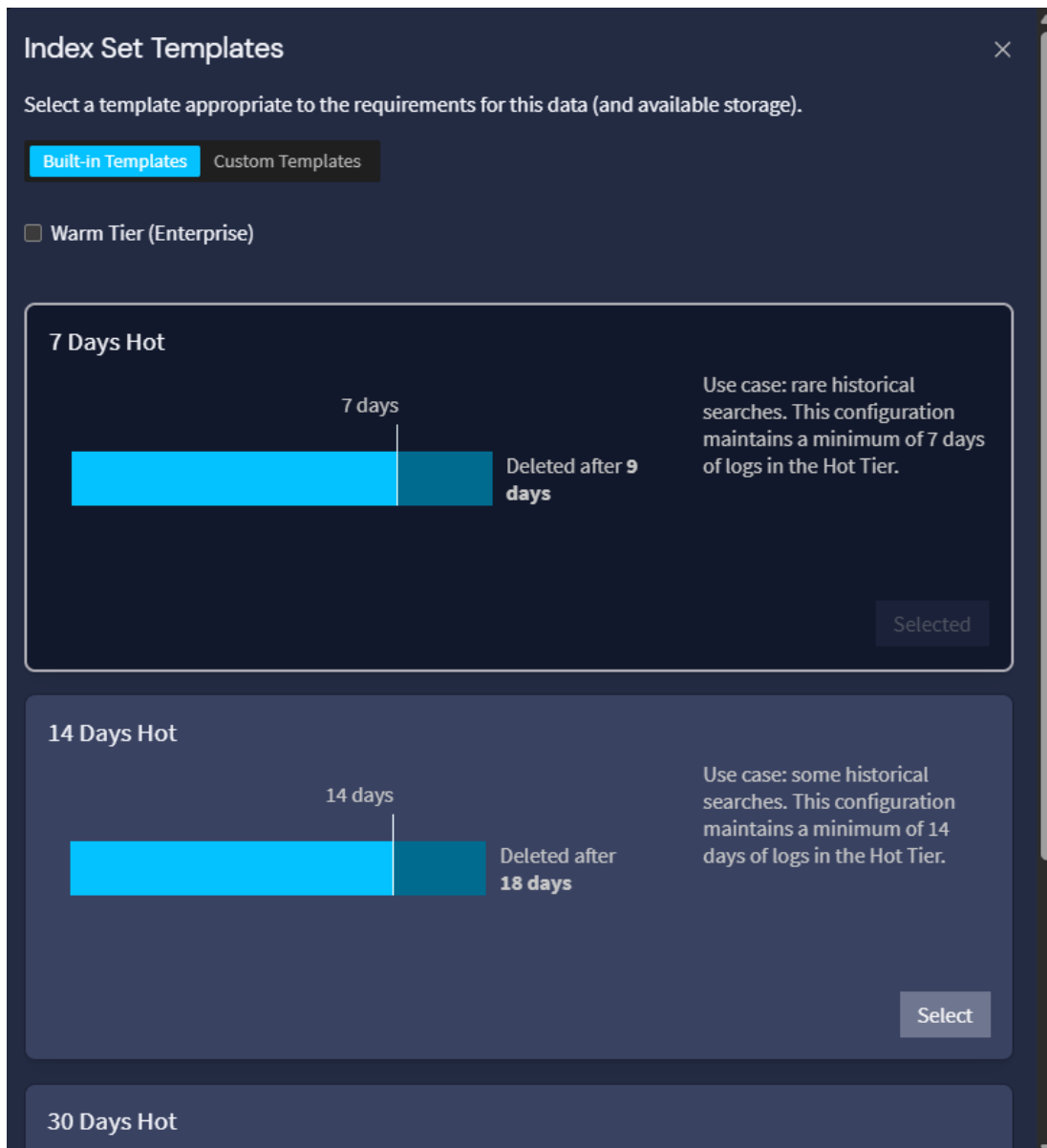
Cancel Create stream

## Création des index

Les index vont permettre de gérer la durée de stockage des logs.

Pour cela, nous devons nous rendre dans « System », puis « Indices », et créer des index en cliquant sur « Create Index ».

Ici, nous allons pouvoir choisir la durée minimum et maximale de conservation des logs.



**Index Set Templates**

Select a template appropriate to the requirements for this data (and available storage).

**Built-in Templates** Custom Templates

Warm Tier (Enterprise)

**7 Days Hot**

7 days

Deleted after **9 days**

Use case: rare historical searches. This configuration maintains a minimum of 7 days of logs in the Hot Tier.

Selected

**14 Days Hot**

14 days

Deleted after **18 days**

Use case: some historical searches. This configuration maintains a minimum of 14 days of logs in the Hot Tier.

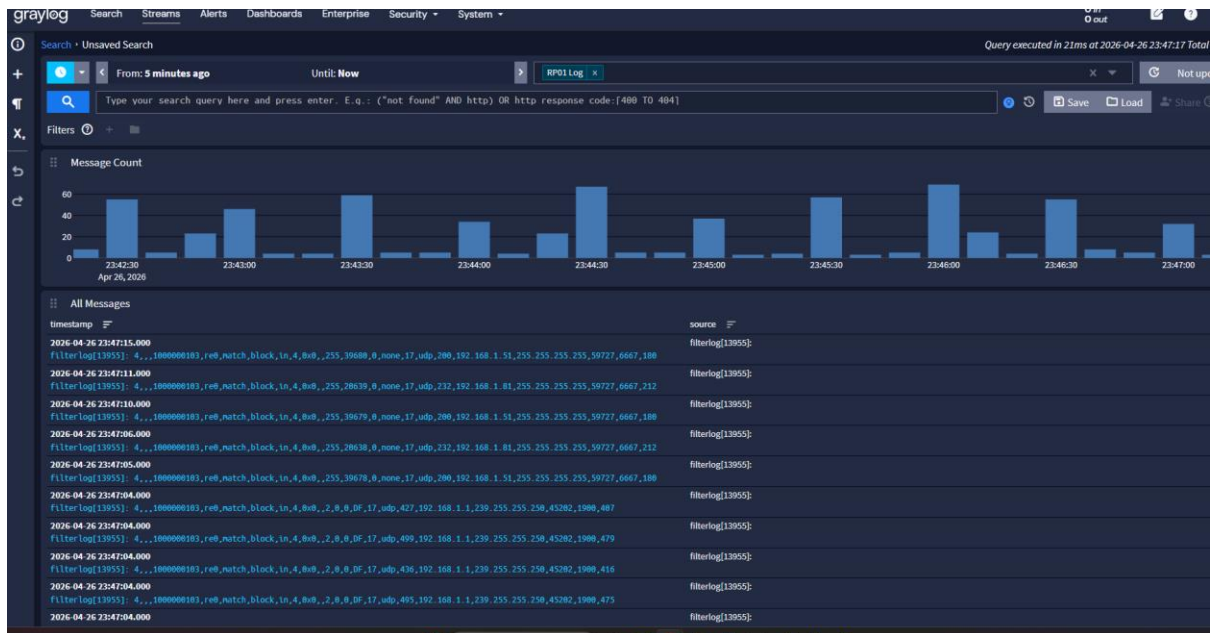
Select

**30 Days Hot**

Ensuite, nous choisir un nom à notre index et définir si l'on souhaite modifier la durée de rétention des logs.

### **Utilisation de Graylog**

Une fois nos stream correctement configurés, nous pouvons nous rendre dans l'onglet «Streams», puis cliquer sur le stream dans lequel on souhaite consulter les logs afin de visualiser les logs de la machine.



Un des axes d'amélioration possible est la création d'alertes afin d'être prévenu en cas de logs spécifiques.

Également, il serait pertinent d'ajouter l'ensemble de nos serveurs afin d'obtenir un regroupement complet des logs, ainsi que de pouvoir augmenter la capacité de stockage du serveur pour améliorer la rétention des logs.