Mouslim ATTALAOUI

PAPPE

Mise en place d'un serveur SAMBA

AFIP FORMATIONS VILLEURBANNE

Sommaire :

Qu’est-ce que Samba ?

Samba est un **logiciel libre** qui permet de créer un serveur de fichiers sous Linux et Unix et d’assurer une compatibilité avec les **protocoles de partage de fichiers Windows**. Il utilise le protocole **SMB (Server Message Block) / CIFS (Common Internet File System)**, qui permet aux ordinateurs sous différents systèmes d'exploitation de partager des fichiers et des imprimantes sur un réseau.

Fonctionnalités principales de Samba

1. **Partage de fichiers et d’imprimantes**
    a. Permet aux clients Windows, Linux et Mac de partager des fichiers et des imprimantes sur un même réseau.
2. **Intégration à un domaine Active Directory (AD)**
    a. Samba peut être configuré comme un **membre d’un domaine Windows** et utiliser AD pour authentifier les utilisateurs.
3. **Gestion des permissions et des utilisateurs**
    a. Possibilité de définir des **droits d’accès** précis sur les fichiers et dossiers partagés.
4. **Serveur de domaine (PDC) ou Contrôleur de domaine Active Directory**
    a. Peut fonctionner en tant que **contrôleur de domaine principal (PDC)** ou s’intégrer à un **domaine Windows existant**.
5. **Sécurité et gestion des accès**
    a. Supporte l’authentification **NTLM, Kerberos et LDAP** pour sécuriser les connexions.

Avantages de Samba

- **Gratuit et Open Source** : Alternative libre aux solutions Microsoft.
- **Interopérabilité** : Fonctionne sur des réseaux mixtes Windows/Linux/Mac.
- **Flexibilité** : Personnalisable et adapté aux petites comme aux grandes entreprises.
- **Sécurisé** : Supporte plusieurs protocoles d’authentification et de chiffrement.

Qu’est-ce que SMB/CIFS ?

Samba repose sur le protocole **SMB (Server Message Block)**, également appelé **CIFS (Common Internet File System)**.

Définition et rôle de SMB/CIFS

- SMB est un **protocole de communication réseau** utilisé principalement par Windows pour permettre le partage de fichiers et d'imprimantes.
- CIFS est une version **étendue** de SMB développée par Microsoft et utilisée dans les systèmes modernes.

Principales fonctionnalités du protocole SMB :

1. **Accès distant aux fichiers** : Permet aux utilisateurs de lire, écrire et modifier des fichiers à distance.
2. **Partage d'imprimantes** : Accès aux imprimantes partagées sur le réseau.
3. **Authentification des utilisateurs** : Utilise Kerberos et NTLM pour sécuriser les connexions.
4. **Notifications de modification des fichiers** : Permet aux clients d'être informés en temps réel des changements dans un dossier partagé.
5. **Optimisation des performances** : Prise en charge du **caching** et des connexions persistantes pour accélérer les transferts.

Introduction

Dans le cadre de mon projet, j'ai mis en place un serveur de fichiers Samba sous Linux Debian, intégré dans un contrôleur de domaine existant. Ce serveur permet le partage sécurisé de fichiers entre utilisateurs d'un réseau. Afin d'améliorer la gestion et la sécurité de cette infrastructure, j'ai ajouté deux solutions complémentaires : **Graylog**, pour l'analyse centralisée des journaux Samba, et **Fail2Ban**, pour la protection contre les tentatives d'intrusion.

Ce dossier a pour objectif de détailler l'installation, la configuration et l'optimisation de ces services, tout en expliquant leur impact sur la gestion et la sécurité du système d'information.

Contexte

Dans un environnement professionnel, la gestion des fichiers partagés est essentielle pour assurer un travail collaboratif efficace et sécurisé. Un serveur Samba permet d'offrir cette fonctionnalité en s'intégrant facilement aux infrastructures existantes, notamment aux domaines Windows via Active Directory.

Cependant, un serveur de fichiers peut être exposé à divers risques, notamment les tentatives d'accès non autorisées et la difficulté de traçabilité des actions des utilisateurs. Pour répondre à ces enjeux, j'ai intégré :

- **Graylog** : une solution de gestion centralisée des journaux, permettant de surveiller en temps réel l'activité du serveur Samba et de détecter d'éventuelles anomalies.
- **Fail2Ban** : un outil de protection contre les attaques par force brute, qui analyse les logs du serveur et bloque automatiquement les adresses IP suspectes.

Ces améliorations permettent de renforcer la sécurité du serveur tout en offrant une meilleure visibilité sur son fonctionnement.

Installation et configuration de Samba

On commence par mettre à jour le serveur en utilisant ces commandes :

-apt update

```
root@mouslim:~# apt update
Atteint :1 http://deb.debian.org/debian bullseye InRelease
Réception de :2 http://deb.debian.org/debian bullseye-updates InRelease [44,1 kB]
Réception de :3 http://security.debian.org/debian-security bullseye-security InRelease [27,2 kB]
Réception de :5 https://artifacts.elastic.co/packages/oss-7.x/apt stable InRelease [10,4 kB]
Réception de :6 http://security.debian.org/debian-security bullseye-security/main Sources [240 kB]
Réception de :7 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [35
0 kB]
Atteint :8 http://repo.mongodb.org/apt/debian buster/mongodb-org/5.0 InRelease
Réception de :9 http://security.debian.org/debian-security bullseye-security/main Translation-en [22
7 kB]
Réception de :10 https://artifacts.elastic.co/packages/oss-7.x/apt stable/main amd64 Packages [107 k
B]
Atteint :4 https://packages.graylog2.org/repo/debian stable InRelease
1 004 ko réceptionnés en 2s (551 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
14 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
root@mouslim:~# _
```

-apt upgrade

```
root@mouslim:~# apt upgrade
```

```
Préparation du dépaquetage de .../libkrb5-3_1.18.3-6+deb11u6_amd64.deb ...
Dépaquetage de libkrb5-3:amd64 (1.18.3-6+deb11u6) sur (1.18.3-6+deb11u5) ...
Paramétrage de libkrb5-3:amd64 (1.18.3-6+deb11u6) ...
(Lecture de la base de données... 55219 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../libgssapi-krb5-2_1.18.3-6+deb11u6_amd64.deb ...
Dépaquetage de libgssapi-krb5-2:amd64 (1.18.3-6+deb11u6) sur (1.18.3-6+deb11u5) ...
Paramétrage de libgssapi-krb5-2:amd64 (1.18.3-6+deb11u6) ...
(Lecture de la base de données... 55219 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../0-krb5-user_1.18.3-6+deb11u6_amd64.deb ...
Dépaquetage de krb5-user (1.18.3-6+deb11u6) sur (1.18.3-6+deb11u5) ...
Préparation du dépaquetage de .../1-libgssrpc4_1.18.3-6+deb11u6_amd64.deb ...
Dépaquetage de libgssrpc4:amd64 (1.18.3-6+deb11u6) sur (1.18.3-6+deb11u5) ...
Préparation du dépaquetage de .../2-libkadm5clnt-mit12_1.18.3-6+deb11u6_amd64.deb ...
Dépaquetage de libkadm5clnt-mit12:amd64 (1.18.3-6+deb11u6) sur (1.18.3-6+deb11u5) ...
Préparation du dépaquetage de .../3-libkdb5-10_1.18.3-6+deb11u6_amd64.deb ...
Dépaquetage de libkdb5-10:amd64 (1.18.3-6+deb11u6) sur (1.18.3-6+deb11u5) ...
Préparation du dépaquetage de .../4-libkadm5srv-mit12_1.18.3-6+deb11u6_amd64.deb ...
Dépaquetage de libkadm5srv-mit12:amd64 (1.18.3-6+deb11u6) sur (1.18.3-6+deb11u5) ...
Préparation du dépaquetage de .../5-krb5-locales_1.18.3-6+deb11u6_all.deb ...
Dépaquetage de krb5-locales (1.18.3-6+deb11u6) sur (1.18.3-6+deb11u5) ...
Préparation du dépaquetage de .../6-libxml2_2.9.10+dfsg-6.7+deb11u6_amd64.deb ...
Dépaquetage de libxml2:amd64 (2.9.10+dfsg-6.7+deb11u6) sur (2.9.10+dfsg-6.7+deb11u5) ...
Préparation du dépaquetage de .../7-filebeat_7.17.28_amd64.deb ...
Dépaquetage de filebeat (7.17.28) sur (7.17.27) ...
Paramétrage de krb5-locales (1.18.3-6+deb11u6) ...
Paramétrage de libgssrpc4:amd64 (1.18.3-6+deb11u6) ...
Paramétrage de filebeat (7.17.28) ...
Paramétrage de libxml2:amd64 (2.9.10+dfsg-6.7+deb11u6) ...
```

Nous allons ensuite installer Samba avec cette commande :

-apt install samba

```
oot@mouslim:~# apt install samba
ecture des listes de paquets... Fait
onstruction de l'arbre des dépendances... Fait
ecture des informations d'état... Fait
es paquets supplémentaires suivants seront installés :
  attr dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm ibverbs-providers libassuan0 libavahi-client3
  libavahi-common-data libavahi-common3 libboost-iostreams1.74.0 libboost-thread1.74.0 libcephfs2 libcups2 libfmt9 libgfapi0 libgfrpc0 libgfxdr0 libglusterfs0
  libgpgme11 libgpm2 libibverbs1 libksba8 libldb2 libncurses6 libnl-3-200 libnl-route-3-200 libnpth0 libpython3.11 librados2 librdmacm1 libtalloc2 libtdb1
  libtevent0 liburing2 libwbclient0 libyaml-0-2 pinentry-curses python3-anyio python3-cffi-backend python3-click python3-colorama python3-cryptography
  python3-dnspython python3-gpg python3-h11 python3-h2 python3-hpack python3-httpcore python3-httpx python3-hyperframe python3-ldb python3-markdown
  python3-markdown-it python3-mdurl python3-pygments python3-requests-toolbelt python3-rfc3986 python3-rich python3-samba python3-sniffio python3-talloc
  python3-tdb python3-yaml samba-ad-provision samba-common samba-common-bin samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
aquets suggérés :
  pinentry-gnome3 tor parcimonie xloadimage scdaemon cups-common gpm pinentry-doc python-cryptography-doc python3-cryptography-vectors python3-trio
  python3-aioquic python-markdown-doc python-pygments-doc ttf-bitstream-vera bind9 bind9utils ctdb ldb-tools ntp | chrony ufw winbind heimdal-clients
es NOUVEAUX paquets suivants seront installés :
  attr dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm ibverbs-providers libassuan0 libavahi-client3
  libavahi-common-data libavahi-common3 libboost-iostreams1.74.0 libboost-thread1.74.0 libcephfs2 libcups2 libfmt9 libgfapi0 libgfrpc0 libgfxdr0 libglusterfs0
  libgpgme11 libgpm2 libibverbs1 libksba8 libldb2 libncurses6 libnl-3-200 libnl-route-3-200 libnpth0 libpython3.11 librados2 librdmacm1 libtalloc2 libtdb1
  libtevent0 liburing2 libwbclient0 libyaml-0-2 pinentry-curses python3-anyio python3-cffi-backend python3-click python3-colorama python3-cryptography
  python3-dnspython python3-gpg python3-h11 python3-h2 python3-hpack python3-httpcore python3-httpx python3-hyperframe python3-ldb python3-markdown
  python3-markdown-it python3-mdurl python3-pygments python3-requests-toolbelt python3-rfc3986 python3-rich python3-samba python3-sniffio python3-talloc
  python3-tdb python3-yaml samba samba-ad-provision samba-common samba-common-bin samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
 mis à jour, 78 nouvellement installés, 0 à enlever et 0 non mis à jour.
l est nécessaire de prendre 44,2 Mo dans les archives.
près cette opération, 153 Mo d'espace disque supplémentaires seront utilisés.
ouhaitez-vous continuer ? [O/n]
```

Nous allons maintenant configurer samba selon nos besoins , pour cela nous allons :

- Entrer dans le fichier Samba avec = nano /etc/samba/smb.conf
- Une fois entrer nous allons ajouter ou modifier les paramètres dans la partit global

```
[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
   workgroup = GSB
   server string = actsrv
   netbios name = actsrv
   security = ADS
   realm = GSB.LOCAL
   password server = w2022dc.gsb.local
   idmap uid = 10000-20000
   idmap gid = 10000-20000
   winbind enum users = yes
   winbind enum groups = yes
   winbind refresh tickets = yes
   template homedir = /data/commun
   template shell = /bin/bash
   client use spnego = yes
   client ntlmv2 auth = yes
   winbind use default domain = yes
   restrict anonymous = 2
```

Ce fichier de configuration montre que le serveur Samba est intégré à un domaine Active Directory (AD) en mode ADS (Active Directory Security). Cela signifie que l'authentification des utilisateurs est entièrement gérée par le contrôleur de domaine Active Directory, ce qui permet une gestion centralisée des droits d'accès et des permissions.
Pour assurer cette intégration, Winbind est utilisé. Il permet de récupérer les comptes d'utilisateurs et de groupes depuis Active Directory et de les mapper en tant qu'utilisateurs locaux sur le serveur Samba. Grâce aux paramètres (winbind enum users = yes) et (winbind enum groups = yes), les utilisateurs et groupes du domaine sont visibles sur la machine Samba comme s'ils étaient des comptes locaux.

- Nous allons maintenant créer la section [test]:

```
[test]
   comment = mon partage linux
   path = /data/test
   guest ok = no
   browseable = yes
   read only = no
   valid users = @GSB\Adminitrateur, _
```

Cette configuration de partage Samba, qui permet un accès à tous les utilisateurs authentifiés du domaine, favorise la collaboration et l'échange de fichiers sans contraintes.

Après être sortis du fichier, exécuter la commande "testparm", elle permet de vérifier la syntaxe du fichier et de s'assurer qu'il soit valide avant l'utilisations.

```
root@mouslim:~# testparm_
```

```
[homes]
        browseable = No
        comment = Home Directories
        create mask = 0700
        directory mask = 0700
        valid users = %S


[printers]
        browseable = No
        comment = All Printers
        create mask = 0700
        path = /var/spool/samba
        printable = Yes


[print$]
        comment = Printer Drivers
        path = /var/lib/samba/printers


[test]
        comment = mon partage linux
        path = /data/test
        read only = No
        valid users = @GSB\Adminitrateur
```

Suite à notre configuration dans notre fichier samba, nous allons créer le dossier “data”, dans lequel nous allons créer le dossier “test” :

```
root@actsrv:~/data# mkdir /data
```

```
root@actsrv:~/data# mdkdir test
```

```
root@actsrv:~/data# ls
test
```

La commande ci-dessous permet d’installer tout les paquets utils pour l’integrations au domaine :

```
root@actsrv:/home/mouslim# apt install cifs-utils samba-client
```

On redemarre le systéme aprés installations :

```
root@actsrv:/home/mouslim# systemctl restart smbd
root@actsrv:/home/mouslim# exit
déconnexion
mouslim@actsrv:/$
```

Avec la commande ci-dessous on installe le kerberos:

```
root@actsrv:~# apt install krb5-user libpam-krb5
```

Elle permet :

- Installe les outils nécessaires pour utiliser Kerberos en tant que client Kerberos sur une machine Linux.
- Permet d'obtenir des tickets Kerberos et d'authentifier des utilisateurs sur un réseau qui utilise Kerberos, comme un domaine Active Directory.

Nous entrons dans le fichier de configurations de krb5 avec :

nano /etc/krb5.conf ==> `root@actsrv:~# nano /etc/krb5.conf`

Une fois à l'intérieur nous allons tout effacer et ajouter cette configuration :

- kdc = w022dc.gsb.local = Cette section définit les serveurs Kerberos pour le domaine GSB.LOCAL.
- admin_server = w022dc.gsb.local = Le serveur w022dc.gsb.local est le serveur Kerberos principal (KDC) qui gère les tickets d'authentification.
- .domain.com = GSB.LOCAL = Ce même serveur gère aussi les tâches administratives, comme la gestion des mots de passe.
- domain.com = GSB.LOCAL = Tous les sous-domaines de domain.com (comme sub1.domain.com) sont reliés au domaine Kerberos GSB.LOCAL.
  Le domaine principal domain.com est aussi lié à GSB.LOCAL

Pour ne pas être géné par le DHCP quand nous utiliserons le resolveur, nous allons configurer notre adresse ip en statique , et ensuite nous allon supprimer le dhcp.

Avec la commande suivant nous accedons au fichier de configurations de l'adresse ip = nano /etc/network/interfaces = `root@mouslim:~# nano /etc/network/interfaces`

Et nous allons la modifier par celle-ci :

```
  GNU nano 5.4                          /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto ens33
iface ens33 inet static
        address 192.168.126.180
        netmask 255.255.255.0
        gateway 192.168.126.2
        dns-nameservers 8.8.8.8 8.8.4.4
```

Pour verifier si tout va bien , nous allon taper la commande “ip a”:

```
root@mouslim:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 10
00
    link/ether 00:0c:29:db:10:e5 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.126.180/24 brd 192.168.126.255 scope global ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fedb:10e5/64 scope link
       valid_lft forever preferred_lft forever
root@mouslim:~#
```

Avec la commande suivante nous allons supprimer le DHCP:

```
root@mouslim:~# apt remove isc-dhcp-client -y
```

Nous allons maintenant configurer le resolveur dns:

- Ce rendre dans le fichier , nano /etc/resolv.conf = `root@actsrv:~# nano /etc/resolv.conf_`
- Modifier ce fichier comme ceci :

```
  GNU nano 5.4                          /etc/resolv.conf
domain gsb.local
search gsb.local
nameserver 192.168.126.5
nameserver 8.8.8.8
```

nameserver 192.168.126.5 : Définit l'adresse IP du serveur DNS principal que votre système doit utiliser pour résoudre les noms de domaine, dans notre cas se sera celui du controleurs de domaine.

Nous allons installer :

- kinit , c’est une commande utilisée dans un environnement Kerberos pour obtenir et renouveler un ticket d'authentification auprès du serveur Kerberos = `root@mouslim:~# apt install kinit_`
- Winbind, permet à un serveur Linux de se comporter comme une machine cliente dans un environnement Windows, en permettant une gestion centralisée

des utilisateurs et de l'accès aux ressources tout en restant intégré dans l'Active Directory. =

```
root@mouslim:~# apt install winbind samba_
```

- Ufw , un moyen facile et pratique de gérer les règles de sécurité du pare-feu sur un système Linux, en simplifiant l'utilisation des règles iptables.
  =

```
root@mouslim:~# apt install ufw_
```

Nous allons autoriser les port dont samba a besoin pour fonctionner , avec ufw et iptables :

- ufw allow 445/tcp
- ufw allow 137,138,139/udp
  Ensuite verifier avec la commande ==> ufw status
- iptables -A INPUT -p tcp --dport 445 -j ACCEPT
- iptables -A INPUT -p udp --dport 137 -j ACCEPT
- iptables -A INPUT -p udp --dport 138 -j ACCEPT
- iptables -A INPUT -p udp --dport 139 -j ACCEPT

Maintenant nous allons vérifier la connexion avec le serveur en utilisant kinit, ensuite klist pour générer un ticket :

```
root@mouslim:~# kinit administrateur
Password for administrateur@GSB.LOCAL:
root@mouslim:~# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrateur@GSB.LOCAL

Valid starting       Expires              Service principal
26/02/2025 16:38:24  27/02/2025 02:38:24  krbtgt/GSB.LOCAL@GSB.LOCAL
        renew until 27/02/2025 16:38:17
root@mouslim:~# _
```

Nous pouvons l'intégrer au domaine du controleur de domaine, avec cette commande
=

```
root@actsrv:~# net ads join -U Administrateur
Enter Administrateur's password:
Using short domain name -- GSB
Joined 'ACTSRV' to dns domain 'gsb.local'
```

La configurations du fichier nsswitch , qui configure la façon dont le système Linux recherche des informations pour diverses bases de données (utilisateurs, groupes, hôtes, etc.). pour y acceder = nano /etc/nsswitch.conf.

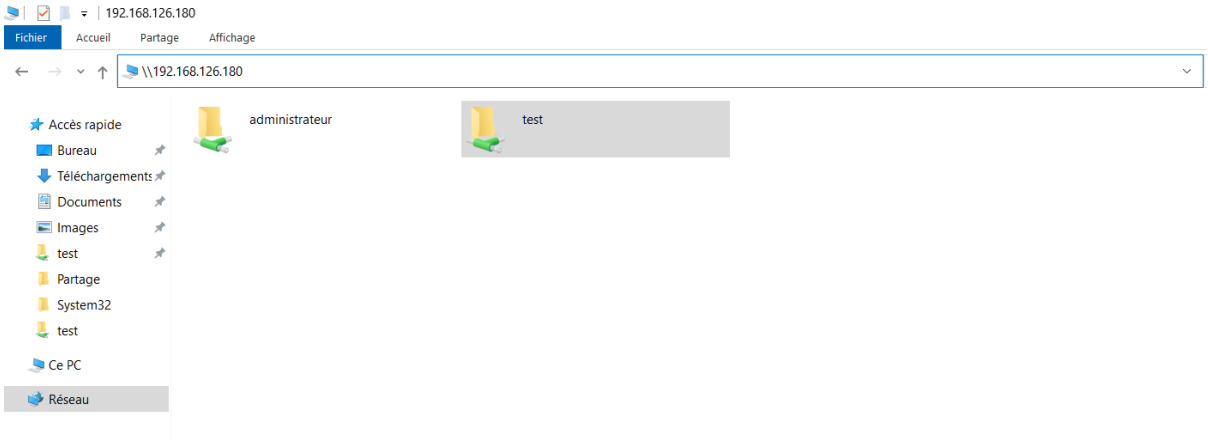
Ensuite modifier le comme ci-dessous :

```
passwd:         compat winbind
group:          compat winbind
shadow:         compat
gshadow:        files

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

- compat : Utilise les fichiers locaux comme /etc/passwd.
- winbind : Recherche également les informations d'utilisateurs dans un domaine Active Directory via Winbind (un service Samba permettant l'intégration avec un AD)

UTILISER LA COMMANDE “systemctl restart samba” pour redémarrer le service samba et actualiser les modifications.

Nous devons aller vérifier si notre partage de dossier et de fichier fonctionne. Dans notre machine client, nous allons dans l’explorateurs de fichier, puis sur la barre de recherche entrer deux antislashs et l’adresse IP du serveur fichier.

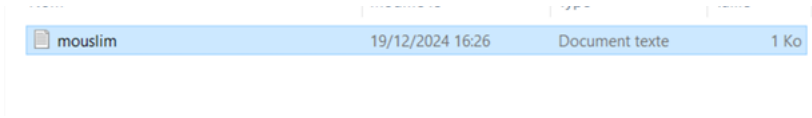


Comme ci-dessus .

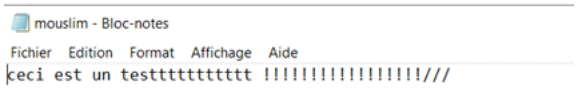
Dans notre serveur fichier nous allons creer un fichier dans notre dossier partager avec la commande “touch”, puis ajouter une ligne de test, .

-
```
systemctl restart smbd
cd /data/test
touch mouslim.txt
```
-
```
GNU nano 5.4                                    mouslim.txt
ceci est un testttttttttttt !!!!!!!!!!!!!!!!!!///
```

Nous retournons maintenant sur windows, et nous pouvons constater le fichier partager :



- Avec a l'interrieur nos lignes de test.



- Notre partage fichier fonctionne donc trés bien.

Nous allons donc maintenant ajouter nos améliorations.

Installations et configurations de Graylog

Sur un Debian 11 nous allons taper les commandes suivantes (car Debian 12 n'est pas compatible avec Graylog) :

Dans l'installation de Graylog, la version Java recommandée est toute version supérieure à Java 8 donc ==>

- apt update
- apt install -y apt-transport-https openjdk-11-jre-headless uuid-runtime pwgen curl dirmngr

```
root@mouslim:~# apt update
```

```
root@mouslim:~# apt install -y apt-transport-https openjdk-11-jre-headless uuid-
runtime pwgen curl dirmngr
```

Je vérifie la version installée de Java avec :

- java –version

```
root@mouslim:~# java -version
openjdk version "11.0.26" 2025-01-21
OpenJDK Runtime Environment (build 11.0.26+4-post-Debian-1deb11u1)
OpenJDK 64-Bit Server VM (build 11.0.26+4-post-Debian-1deb11u1, mixed mode, shar
ing)
```

Maintenant j'installe ElasticSearch:

- Ajoutons d'abord la clé Elasticsearch GPG avec ==> wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

```
root@mouslim:~# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
sudo apt-key add -
```

- Ajoutons ensuite le dépôt Elasticsearch avec ==> echo "deb https://artifacts.elastic.co/packages/oss-7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list

```
root@mouslim:~# echo "deb https://artifacts.elastic.co/packages/oss-7.x/apt stab
le main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list
```

- Enfin j'installe avec ==> apt install -y elasticsearch-oss

```
root@mouslim:~# apt install -y elasticsearch-oss
```

Effectuez ensuite les configurations du fichier YAML et définissez le nom du cluster sur **graylog** comme ci-dessous:

- Nous entrons dans le fichier de configurations avec ==> nano /etc/elasticsearch/elasticsearch.yml
- Je recherche et défini le nom du cluster et ajoutez les lignes ci-dessous :

```
  GNU nano 5.4          /etc/elasticsearch/elasticsearch.yml
# ======================== Elasticsearch Configuration =========================
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluste>
#
# Please consult the documentation for further information on configuration opt>
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# ---------------------------------- Cluster -----------------------------------
#
# Use a descriptive name for your cluster:
#
cluster.name: graylog
action.auto_create_index: false
#
```

- Je recharge et démarre le service Elasticsearch comme ci-dessous:

```
root@mouslim:~# systemctl daemon-reload
root@mouslim:~# systemctl start elasticsearch
root@mouslim:~# systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /lib/
systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
```

- Je vérifie l'état du service :

```
root@mouslim:~# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor>
     Active: active (running) since Tue 2025-03-04 16:49:46 CET; 24min ago
       Docs: https://www.elastic.co
   Main PID: 681 (java)
      Tasks: 56 (limit: 2280)
     Memory: 609.7M
        CPU: 59.985s
     CGroup: /system.slice/elasticsearch.service
             └─681 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.netw>

mars 04 16:49:26 mouslim systemd[1]: Starting Elasticsearch...
mars 04 16:49:46 mouslim systemd[1]: Started Elasticsearch.
lines 1-13/13 (END)
```

Je vérifie que Elasticsearch s'exécute sur le port 9200 a l'aide de cURL ==> curl -X GET http://localhost:9200:

```
root@mouslim:~# curl -X GET http://localhost:9200
{
  "name" : "mouslim",
  "cluster_name" : "graylog",
  "cluster_uuid" : "wXA0cd5wS-GMwApqwvzKQQ",
  "version" : {
    "number" : "7.10.2",
    "build_flavor" : "oss",
    "build_type" : "deb",
    "build_hash" : "747e1cc71def077253878a59143c1f785afa92b9",
    "build_date" : "2021-01-13T00:42:12.435326Z",
    "build_snapshot" : false,
    "lucene_version" : "8.7.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

J'installe MongoDB:

- Nous installerons MongoDB en ajoutant des référentiels comme ci-dessous.

```
root@mouslim:~# wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | s
udo apt-key add -
```

- Ajoutons ensuite le dépôt:

```
root@mouslim:~# echo "deb http://repo.mongodb.org/apt/debian buster/mongodb-org/
5.0 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list
```

- Je mets à jour et j'installe MongoDB comme ci-dessous :

```
root@mouslim:~# apt-get update
```

```
root@mouslim:~# apt-get install -y mongodb-org mongodb-org-database mongodb-org-
server mongodb-org-shell mongodb-org-mongos mongodb-org-tools
```

- Ensuite, je démarre et j'active le service MongoDB pour qu'il s'exécute au démarrage.

```
root@mouslim:~# systemctl start mongod
root@mouslim:~# systemctl enable mongod
```

- Je vérifie que le service est en cours d'exécution :

```
root@mouslim:~# systemctl status mongod
● mongod.service - MongoDB Database Server
     Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset>
     Active: active (running) since Tue 2025-03-04 16:49:26 CET; 37min ago
       Docs: https://docs.mongodb.org/manual
   Main PID: 685 (mongod)
     Memory: 91.1M
        CPU: 36.671s
     CGroup: /system.slice/mongod.service
             └─685 /usr/bin/mongod --config /etc/mongod.conf

mars 04 16:49:26 mouslim systemd[1]: Started MongoDB Database Server.
mars 04 16:49:27 mouslim mongod[685]: {"t":{"$date":"2025-03-04T15:49:27.257Z"}>
```

J'installe le serveur Graylog:

- Je télécharge et j'installe le package du référentiel Graylog:

```
root@mouslim:~# wget https://packages.graylog2.org/repo/packages/graylog-4.2-rep
ository_latest.deb
root@mouslim:~# dpkg -i graylog-4.2-repository_latest.deb
```

- Je mets à jour les référentiels de cache et j'installe le serveur Graylog:

```
root@mouslim:~# apt-get update
```

```
root@mouslim:~# apt install -y graylog-server
```

- Ensuite, nous utiliserons la commande pwgen ci-dessous pour générer un secret pour sécuriser les mots de passe des utilisateurs comme ci-dessous:

```
root@mouslim:~# pwgen -N 1 -s 96
17x24z96X2OQ8VM72AMcIuEWQPiYv8gjtbQzdG6xBmoxHbI7MOtgaWqyMGQuBHjLon7JfBPnaZRR26gr
MHb2SGEAi4EsvcNh
```

- Je copie le code secret et je le mets ci-dessous:

```
root@mouslim:~# nano /etc/graylog/server/server.conf
```

```
# You MUST set a secret to secure/pepper the stored user passwords here. Use at>
# Generate one by using for example: pwgen -N 1 -s 96
# ATTENTION: This value must be the same on all Graylog nodes in the cluster.
# Changing this value after installation will render all user sessions and encr>
password_secret = OvTwEWjwXvxxn824pG5vRorsA0PCs0v7g49goiM4uZSZHBwjbdMMmpIX6lDML>
```

- Dans le fichier **.conf,** j'ajoute également les lignes ci-dessous :

```
rest_listen_uri = http://127.0.0.1:9000/api/
web_listen_uri = http://127.0.0.1:9000/
```

Je sauvegarde et je quitte .

- Je continue et je créer un mot de passe sha256 pour l'administrateur, il me servira pour l'interface web :

```
root@mouslim:~# echo -n Str0ngPassw0rd | sha256sum
```

Il devrait afficher une clefs comme celle ci ,

```
root@mouslim:~# echo -n Str0ngPassw0rd | sha256sum
434e27fac24a15cbf8b160b7b28c143a67d9e6939cbb388874e066e16cb32d75  -
```

- Je copie cette sortie et je l'utilise à l'étape ci-dessous:

```
root@mouslim:~# nano /etc/graylog/server/server.conf
```

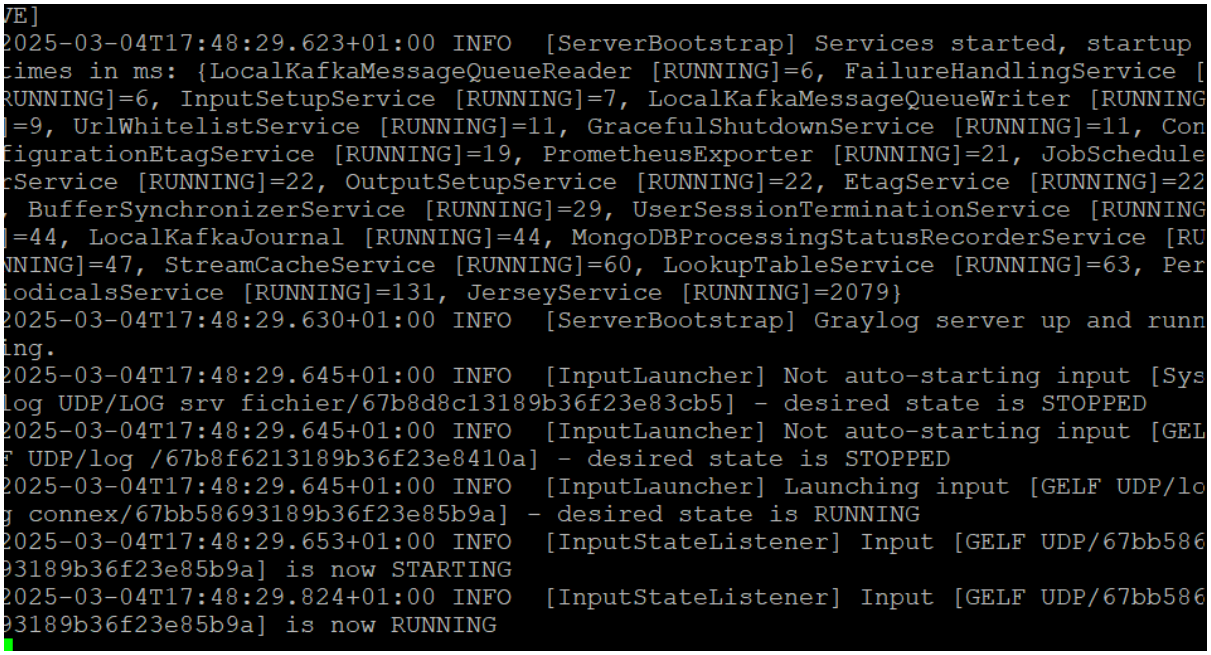
```
root_password_sha2 = 3b5e7e46e71916c2cf08a9882b5609d627e2e78380371b6408060532eb
```

- Ceci fait, le serveur Graylog est maintenant prêt à être utilisé, je démarre et active le service comme ci-dessous :

```
root@mouslim:~# systemctl daemon-reload
root@mouslim:~# systemctl restart graylog-server
root@mouslim:~# systemctl enable graylog-server
Synchronizing state of graylog-server.service with SysV service script with /lib
/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable graylog-server
```

- Je vérifie le journal comme ci-dessous:

```
root@mouslim:~# tail -f /var/log/graylog-server/server.log
```



Si le serveur fonctionne correctement, vous devriez voir le résultat ci-dessus.

- Pour que j'accéde à l'interface Web Graylog avec une adresse IP et un port de serveur :

```
root@mouslim:~# nano /etc/graylog/server/server.conf
```

```
# Default: 127.0.0.1:9000
http_bind_address = 127.0.0.1:9000
#http_bind_address = [2001:db8::1]:9000
```

Je redémarre le service pour que les modifications apportées soient appliquées:

```
root@mouslim:~# systemctl restart graylog-server
```

Nous devons maintenant assurer la bonne configuration pour permettre à samba d'envoyer les journaux a graylog.

Pour cela nous allons utiliser filebeat et rsyslog.

Avant nous devons modifier le fichier de configurations de samba, avec ==>

nano /etc/samba/smb.conf.

Et ajouter ces lignes :

```
log file = /var/log/samba/log.%m
max log size = 5000
log level = 3
vfs objects = full_audit
full_audit:prefix = %u|%I|%S  #
full_audit:success = open,close,write,unlink
full_audit:failure = none
full_audit:facility = local5
full_audit:priority = notice
```

Redémarrez Samba :systemctl restart smbd nmbd

```
root@mouslim:~# systemctl restart smbd nmbd
```

Créer un fichier log spécifique :

```
root@mouslim:~# touch /var/log/samba/audit.log
```

Maintenant les commandes pour donner les autorisations :

```
root@mouslim:~# chmod 640 /var/log/samba/audit.log
root@mouslim:~# chown root:adm /var/log/samba/audit.log
```

Maintenant je vais modifier le fichier /etc/rsyslog.conf pour permettre d'envoyer `local5` vers le fichier spécifique :

```
local5.*    /var/log/samba/audit.log
```

Redémarrer Rsyslog et Samba avec ==> systemctl restart rsyslog , systemctl restart smbd nmbd

Installations de Filebeat:

- Ajouter la clé : wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -

```
root@mouslim:~# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
apt-key add -
```

- Ajouter le dépôt : echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | tee /etc/apt/sources.list.d/elastic-7.x.list

```
root@mouslim:~# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable m
ain" | tee /etc/apt/sources.list.d/elastic-7.x.list
```

- Mettre à jour et installer : apt update && apt install filebeat

```
root@mouslim:~# apt update && apt install filebeat
```

Configurer Filebeat pour envoyer les logs à Graylog :

- Editer le fichier suivant :

```
root@mouslim:~# nano /etc/filebeat/filebeat.yml
```

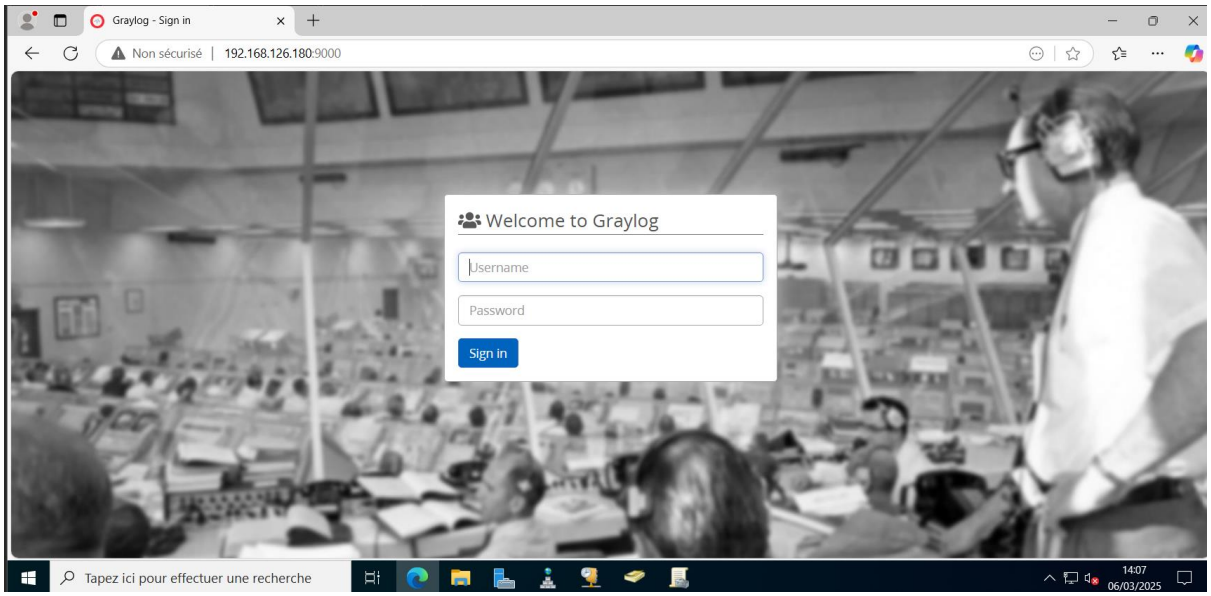
```
 GNU nano 5.4                  /etc/filebeat/filebeat.yml *
filebeat.inputs:
  - type: log
    enabled: true
    paths:
      - /var/log/samba/log.*
      - /var/log/samba/audit.log
output.gelf:
  host: 192.168.126.180:9000
  port: 12201
```

- Redémarrez Filebeat:

```
root@mouslim:~# systemctl restart filebeat
root@mouslim:~# systemctl enable filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/syste
md/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
root@mouslim:~#
```

Configurer Graylog pour recevoir les logs:

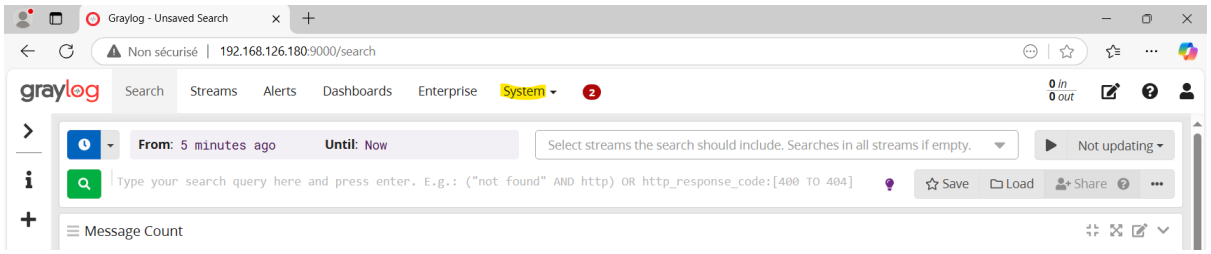
- Dans une machine Windows sur le même réseau, sur un moteur de recherche, taper dans la barre de recherche l'adresse IP suivi du port :



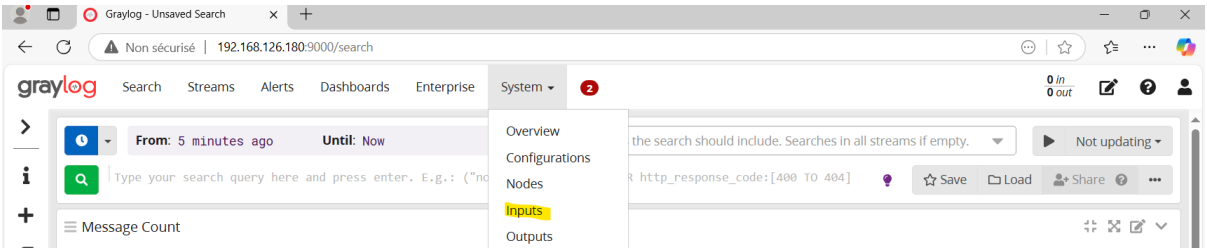
Puis se connecter à l'aide de l'identifiant "admin" par défaut et le mot de passe configurer plus tôt .

Sur le serveur Graylog, créez une entrée Beats :

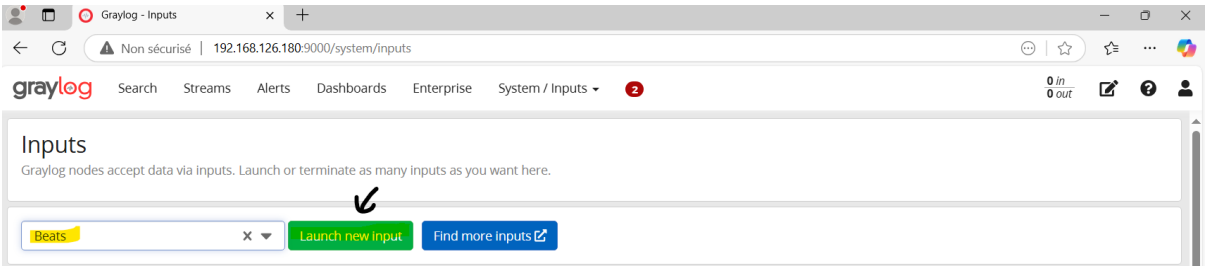
- Aller dans system:



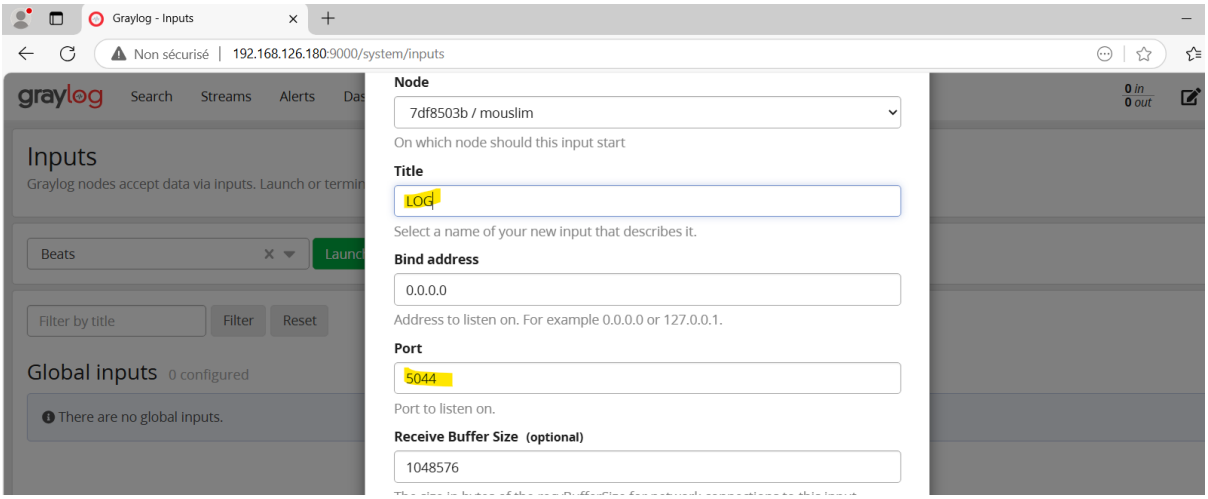
- Puis dans inputs:



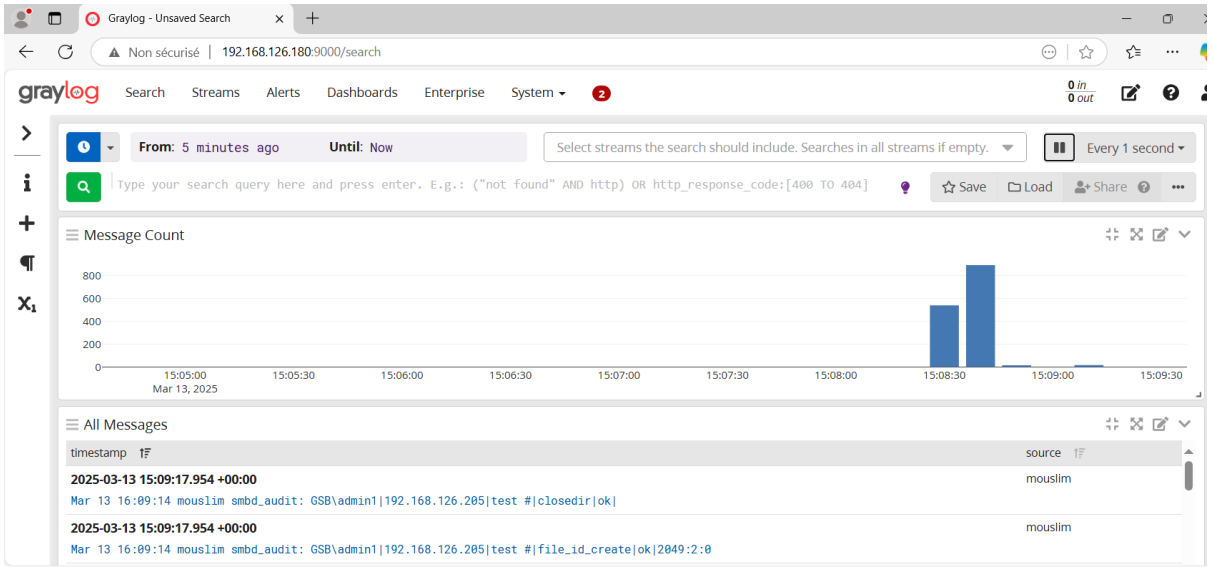
- Sélectionnez **Beats (Filebeat)** et cliquez sur **Launch new input :**



- Configurez un titre et l'entrée avec le port 5044 :



- Entrer dans le search pour visualiser les logs:



Nous pouvons bien constatez que les logs de connexions sont présents avec les détails de connexion .

Installation et configurations de FAIL2BAN

Avant de commencer, mettre à jour mon serveur : apt update && apt upgrade -y

- Puis installe Fail2Ban : `apt update && sudo apt upgrade -y`

```
root@mouslim:~# apt install fail2ban -y
```

- Vérifie que Fail2Ban est bien installé et actif : systemctl status fail2ban

```
root@mouslim:~# systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor pres>
     Active: active (running) since Tue 2025-03-04 16:49:26 CET; 2 days ago
       Docs: man:fail2ban(1)
   Main PID: 696 (fail2ban-server)
      Tasks: 3 (limit: 2280)
     Memory: 7.2M
        CPU: 2.540s
     CGroup: /system.slice/fail2ban.service
             └─696 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

mars 04 16:49:26 mouslim systemd[1]: Starting Fail2Ban Service...
mars 04 16:49:26 mouslim systemd[1]: Started Fail2Ban Service.
mars 04 16:49:26 mouslim fail2ban-server[696]: Server ready
lines 1-14/14 (END)
```

- Maintenant démarre-le :

```
root@mouslim:~# systemctl start fail2ban
root@mouslim:~# systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /lib/syste
md/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
```

Configuration de Fail2Ban :

- Créer le fichier de configuration personnalisé : cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

```
root@mouslim:~# cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

- Puis édite le fichier : nano /etc/fail2ban/jail.local

```
root@mouslim:~# nano /etc/fail2ban/jail.local
```

- Ajoute ou modifie la section suivante pour activer la protection de Samba :

```
  GNU nano 5.4                          /etc/fail2ban/jail.local
# "backend" specifies the backend used to get files modification.
# Available options are "pyinotify", "gamin", "polling", "systemd" and "auto".
# This option can be overridden in each jail as well.
#
# pyinotify: requires pyinotify (a file alteration monitor) to be installed.
#              If pyinotify is not installed, Fail2ban will use auto.
# gamin:     requires Gamin (a file alteration monitor) to be installed.
#              If Gamin is not installed, Fail2ban will use auto.
# polling:   uses a polling algorithm which does not require external libraries.
# systemd:   uses systemd python library to access the systemd journal.
#              Specifying "logpath" is not valid for this backend.
#              See "journalmatch" in the jails associated filter config
# auto:      will try to use the following backends, in order:
#              pyinotify, gamin, polling.
#
# Note: if systemd backend is chosen as the default but you enable a jail
#       for which logs are present only in its own log files, specify some other
#       backend for that jail (e.g. polling) and provide empty value for
#       journalmatch. See https://github.com/fail2ban/fail2ban/issues/959#issuecomment-74901200
backend = systemd

# "usedns" specifies if jails should trust hostnames in logs,
#   warn when DNS lookups are performed, or ignore all hostnames in logs
#
```

Pour que fail2ban puisse se baser sur systemd, le backend doit être modifié dans la configuration de fail2ban .

Insataller syslog celui-ci fournira les logs dans /var/log/ :

-
```
root@mouslim:~# apt install syslog
```

Décommenter la ligne commençant par ignoreip :

Ces paramètres de Fail2Ban définissent les règles de bannissement :

- bantime = 1d → L'adresse IP est bannie pendant 1 jour (24 heures) après avoir dépassé le nombre de tentatives autorisées.
- findtime = 10m → La fenêtre de détection des échecs est de 10 minutes : si un utilisateur échoue plusieurs fois dans ce laps de temps, il sera banni.

- maxretry = 5 → Une IP est bannie après 5 échecs de connexion détectés dans la période findtime.

```
#
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned.
bantime  = 1d

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime  = 10m

# "maxretry" is the number of failures before a host get banned.
maxretry = 5
```

- Après toute modification de la configuration et l'installation de rsyslog, il est important de redémarrer le service fail2ban pour que les changements soient pris en compte :

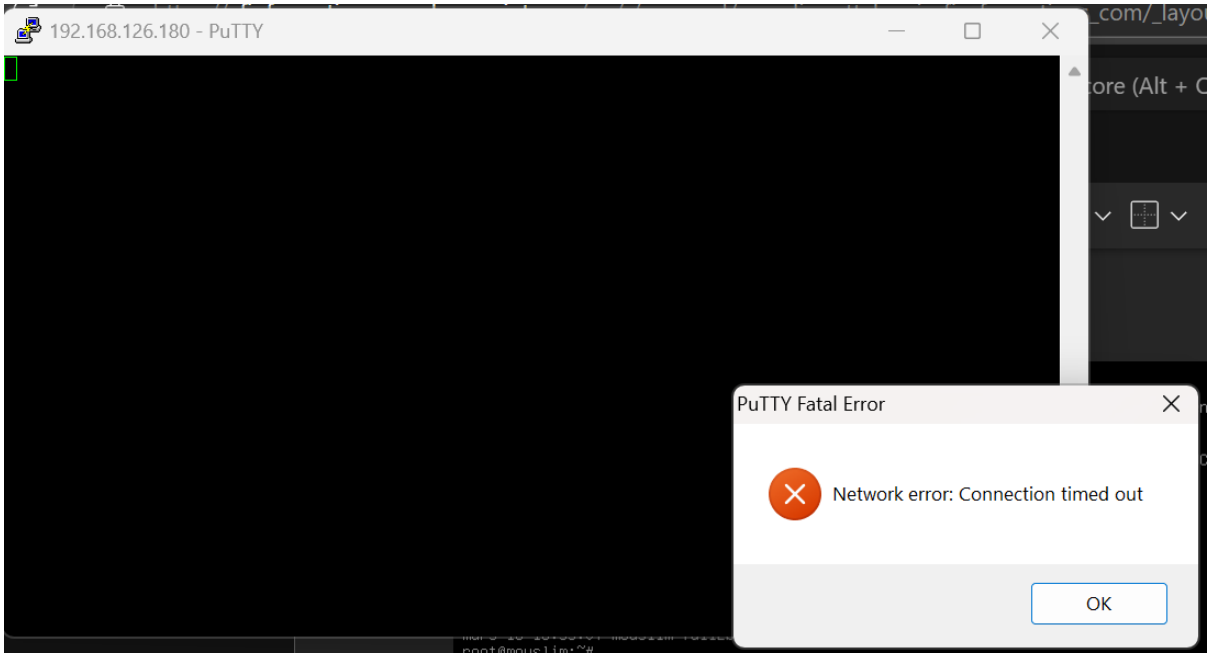
```
root@debian:~# systemctl restart fail2ban
```

- Nous allons vérifier si tout fonctionne bien :

```
root@mouslim:~# systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-03-13 13:55:06 CET; 1h 50min ago
       Docs: man:fail2ban(1)
    Process: 1612 ExecStartPre=/bin/mkdir -p /run/fail2ban (code=exited, status=0/SUCCESS)
   Main PID: 1613 (fail2ban-server)
      Tasks: 5 (limit: 4618)
     Memory: 18.2M
        CPU: 8.131s
     CGroup: /system.slice/fail2ban.service
             └─1613 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

mars 13 13:55:06 mouslim systemd[1]: Starting Fail2Ban Service...
mars 13 13:55:06 mouslim systemd[1]: Started Fail2Ban Service.
mars 13 13:55:07 mouslim fail2ban-server[1613]: Server ready
root@mouslim:~# _
```

Pour s’assurer que fail2ban bannis bien les IP :

- Se connecter avec puTTY mettre le bon identifiant et 5 mauvais mot de passe :



Fail2ban a bien fonctionné, l'IP est banni pendant 1 jour.

Conclusion

La mise en place d'un serveur Samba, couplé à l'intégration de Graylog et de Fail2Ban, a permis d'optimiser à la fois la gestion des fichiers partagés et la sécurité du serveur. Grâce à l'ajout de Graylog, il est désormais possible de centraliser et d'analyser les journaux d'événements du serveur Samba, facilitant ainsi la détection d'anomalies et d'incidents. L'implémentation de Fail2Ban renforce la sécurité en bloquant automatiquement les adresses IP suspectes, réduisant ainsi les risques d'attaques par force brute et autres tentatives malveillantes.

Les différentes améliorations apportées, notamment l'automatisation de la collecte des logs et la mise en place d'une protection dynamique contre les tentatives d'intrusion, contribuent à la stabilité, à la sécurité et à la performance du serveur Samba. Ces évolutions font de ce projet une solution robuste et performante pour la gestion de fichiers partagés dans un environnement sécurisé.

Enfin, l'intégration de ces outils dans un contexte de réseau d'entreprise ou de laboratoire offre une base solide pour la gestion de fichiers tout en assurant une surveillance continue et une protection renforcée contre les menaces potentielles. Ce projet met en lumière l'importance d'une configuration soignée et d'une vigilance constante pour garantir la sécurité des infrastructures informatiques.