srv-logs

```
MARK RODERICK - supervision et journalisation- logs
```

3.5.4 Prévention des attaques 3.5.5 Détection des actions malveillantes
3.5.6 Analyse d’incidents de sécurité, proposition et mise en œuvre de contre-mesures

---

1. C'est quoi un "Log" ? (Le Journal de Bord)

Un **log** (ou "journalisation" en français) est un enregistrement chronologique de tout ce qui se passe sur un système, une application ou un équipement réseau. C'est comme la boîte noire d'un avion ou un journal intime numérique.

Chaque ligne de log contient généralement :

- **L'horodatage (Timestamp) :** Quand ça s'est passé.
- **La source :** Quelle machine ou application a généré l'événement.
- **Le niveau de gravité :** (INFO, WARNING, ERROR, CRITICAL).
- **Le message :** La description de ce qui s'est passé (ex: "Échec de connexion pour l'utilisateur admin").

---

2. Pourquoi les logs sont-ils importants ?

Sans logs, tu es aveugle. Voici les quatre raisons majeures de leur importance :

1. **Dépannage (Troubleshooting) :** Quand un serveur plante à 3h du matin, le log te dit précisément quelle erreur a causé l'arrêt.
2. **Sécurité :** Si quelqu'un tente de forcer un mot de passe (Brute Force), les logs de sécurité enregistrent toutes les tentatives infructueuses. C'est la base pour détecter une intrusion.
3. **Analyse de performance :** Les logs permettent de voir si une application devient lente ou si un disque dur commence à montrer des signes de faiblesse.
4. **Conformité légale :** Dans beaucoup de pays et d'industries (banque, santé), la loi oblige à conserver les logs pendant une certaine durée (souvent 1 an) pour pouvoir enquêter en cas d'incident.

---

3. C'est quoi un Serveur de Logs ? (La Centralisation)

Dans un petit réseau, tu peux aller lire les logs directement sur chaque machine. Mais si tu as 50 serveurs, 10 switches et 5 pare-feu, c'est impossible. C'est là qu'intervient le **Serveur de Logs centralisé**.

Le principe :

Toutes tes machines envoient leurs logs en temps réel vers un serveur unique (souvent via un protocole appelé **Syslog**).

---

4. Pourquoi centraliser les logs sur un serveur dédié ?

Avoir un serveur de logs spécifique présente des avantages stratégiques :

Avantage	Pourquoi c'est crucial ?
Persistance	Si un pirate efface les logs sur le serveur qu'il a hacké pour cacher ses traces, les logs sont déjà sauvegardés en sécurité sur ton serveur de logs central.
Corrélation	Tu peux voir que l'erreur sur la base de données (Serveur A) est liée à une tentative de connexion suspecte sur le Pare-feu (Machine B).
Vitesse de recherche	Au lieu de te connecter à 10 machines, tu fais une seule recherche globale (ex: "Chercher 'Erreur 500' partout").
Alerte	Le serveur de logs peut t'envoyer un mail ou un SMS dès qu'il voit passer le mot "CRITICAL" dans un flux.

🛠️ Les outils populaires (Le "Stack")

Si tu veux monter un serveur de logs, tu entendras souvent parler de ces solutions :

- **ELK Stack :** Le trio le plus célèbre (Elasticsearch pour stocker, Logstash pour trier, Kibana pour visualiser sous forme de jolis graphiques).
- **Graylog :** Très puissant et souvent plus simple à gérer que ELK.
- **Splunk :** La Rolls-Royce (très puissant, mais payant et cher).
- **RSyslog / Syslog-ng :** Les outils de base sous Linux pour transférer les logs d'une machine à l'autre.

> **En résumé :** Le log est la **preuve** de ce qui s'est passé. Le serveur de logs est le **coffre-fort** et la **loupe** qui te permet d'analyser ces preuves rapidement.

Est-ce que tu veux que je t'explique comment on configure une machine pour qu'elle envoie ses logs vers un serveur (le protocole Syslog) ?

Objectif :
Mise en place d'une VM utilisant Debian 13 On utilisera la solution LogAnalyser Contexte TP Mettre en place et récupération de logs.

Objectif :
Centraliser les logs sur LogAnalyser (copier les logs de : --> logs de cisco, des stormshield, des AD DNS) Réalisation TP On va mettre en place une Debian 13 en utilisant une image ISO, elle sera de gen1

🛡️ Définition d’un SIEM

Un **SIEM (Security Information and Event Management)** est une solution de cybersécurité qui permet de **collecter, centraliser et analyser les journaux (logs)** provenant des systèmes, applications et équipements réseau. Ses objectifs principaux sont :

- **Détection des menaces** en temps réel grâce à la corrélation des événements.
- **Surveillance de la sécurité** et visibilité complète sur l’infrastructure.
- **Conformité réglementaire** (ex. RGPD, ISO 27001, PCI-DSS) via des rapports et audits.
- **Réponse aux incidents** plus rapide grâce aux alertes et automatisations.

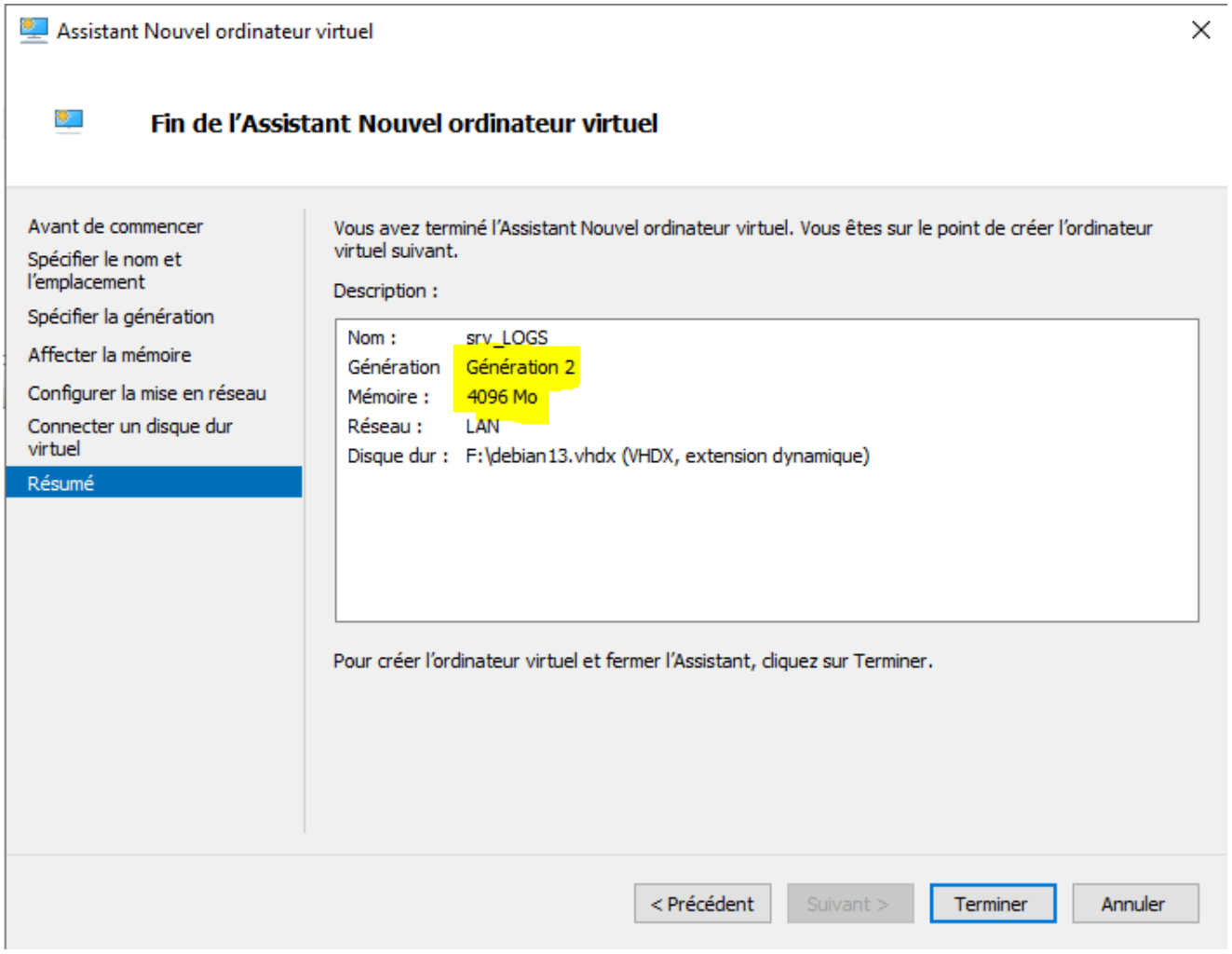
📊 Comparatif des principaux SIEM

Outil	Type / Modèle	Points forts	Cas d’usage idéal
Splunk Enterprise Security	Logiciel propriétaire, niveau entreprise	Analyses en temps réel, tableaux de bord personnalisables, nombreuses intégrations	Grandes entreprises nécessitant une visibilité complète et conformité
ELK Stack (Elastic SIEM)	Open source	Flexible, évolutif, économique, recherche puissante	Équipes techniques, startups, PME avec contraintes budgétaires
Graylog Security	Open source	Facile à utiliser, personnalisable, abordable, bonne gestion des logs	PME ou organisations cherchant un SIEM léger
IBM QRadar	Logiciel propriétaire, niveau entreprise	Détection avancée des menaces, rapports de conformité, moteur de corrélation puissant	Secteurs réglementés (finance, santé, administrations)
Exabeam Fusion SIEM	Logiciel propriétaire, nouvelle génération	Analyse du comportement des utilisateurs (UEBA), automatisation, machine learning	Entreprises axées sur la détection des menaces internes et l’automatisation

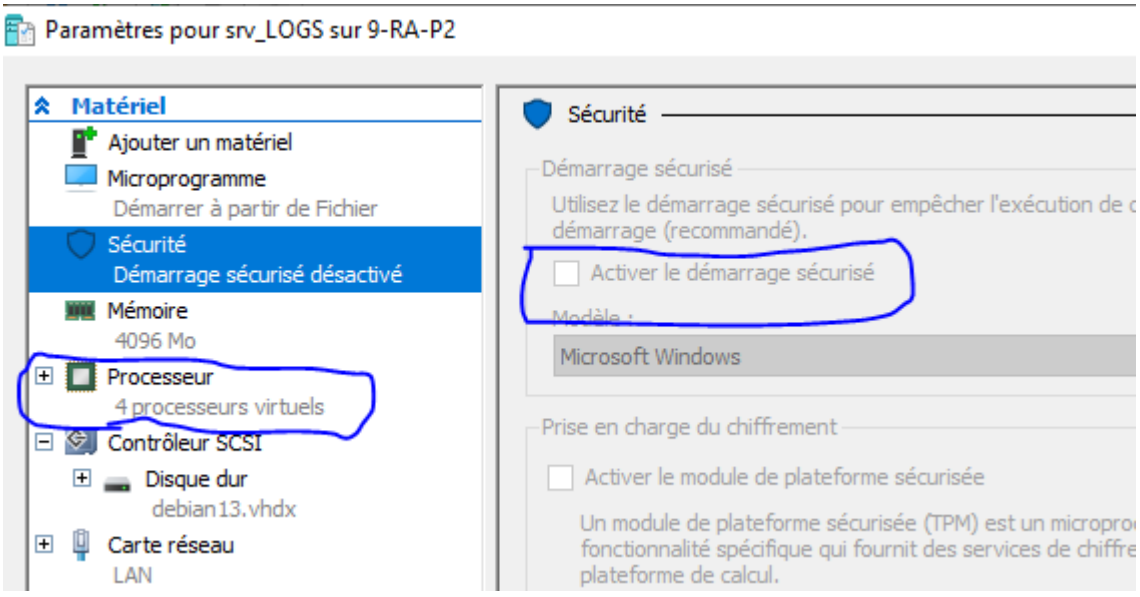
- **Splunk & IBM QRadar** → puissants, adaptés aux grandes organisations et aux environnements réglementés.

- **ELK & Graylog** → flexibles, open source, économiques, parfaits pour PME et équipes techniques.
- **Exabeam** → moderne, centré sur l'automatisation et l'analyse comportementale.

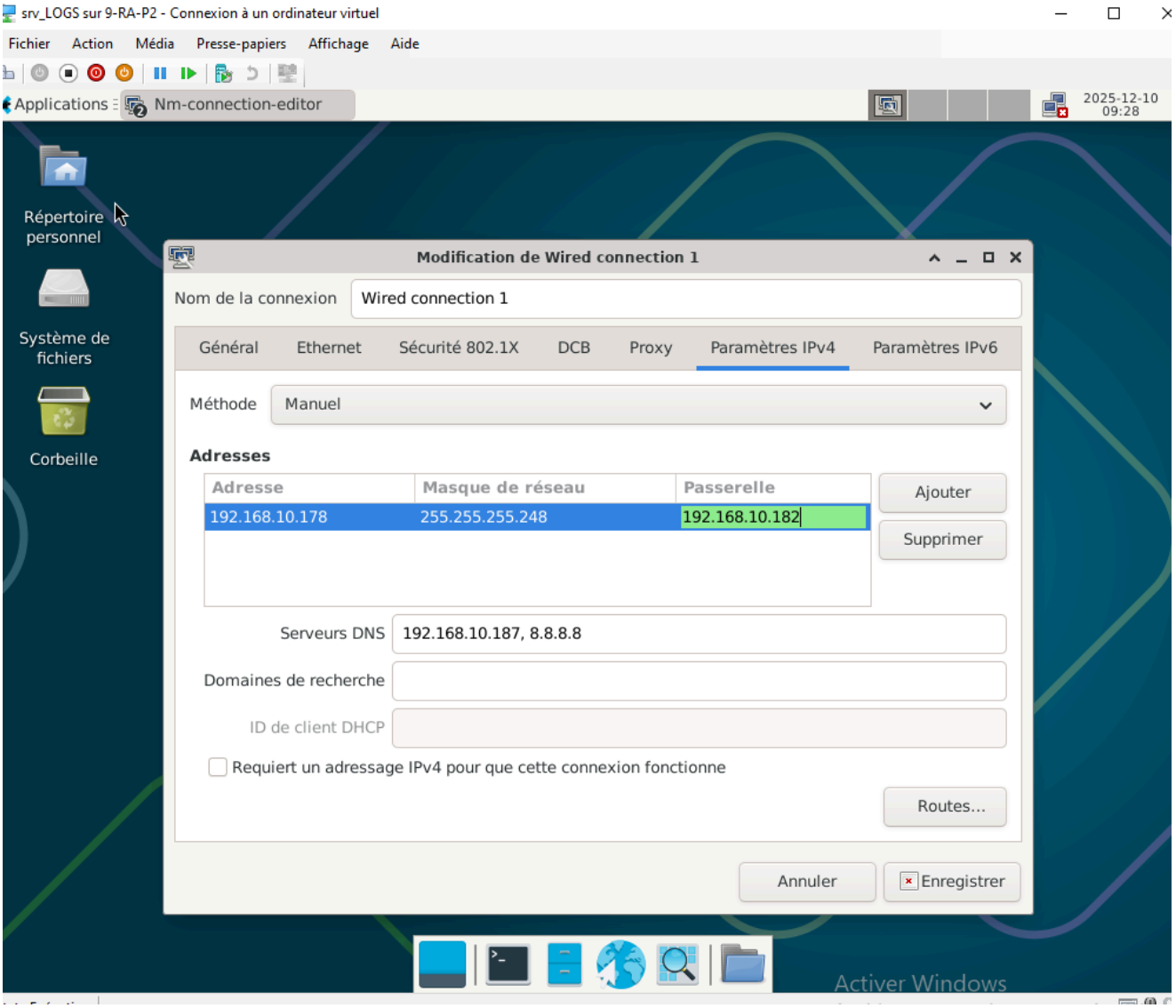
Debian 13 srv de logs



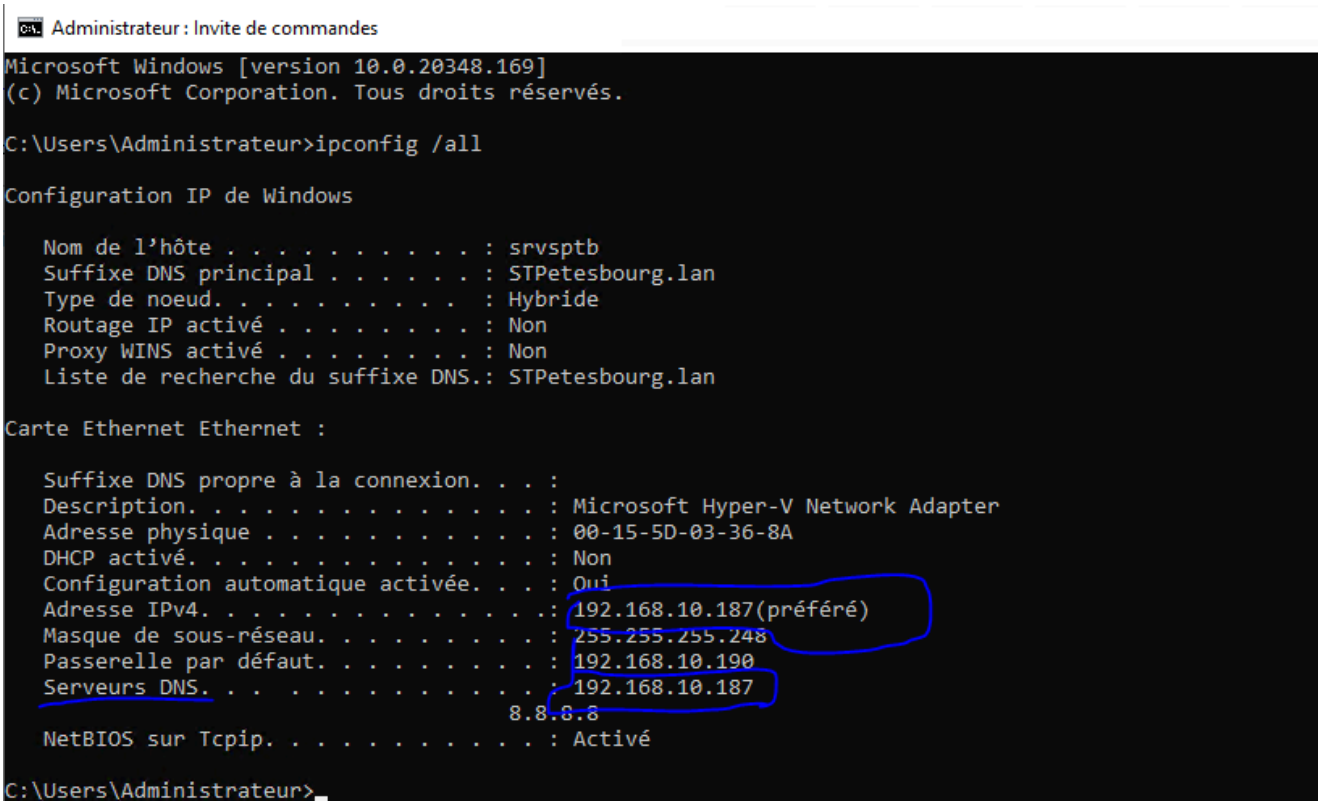
POUR FACILITER LA DEMARRAGE DU VM



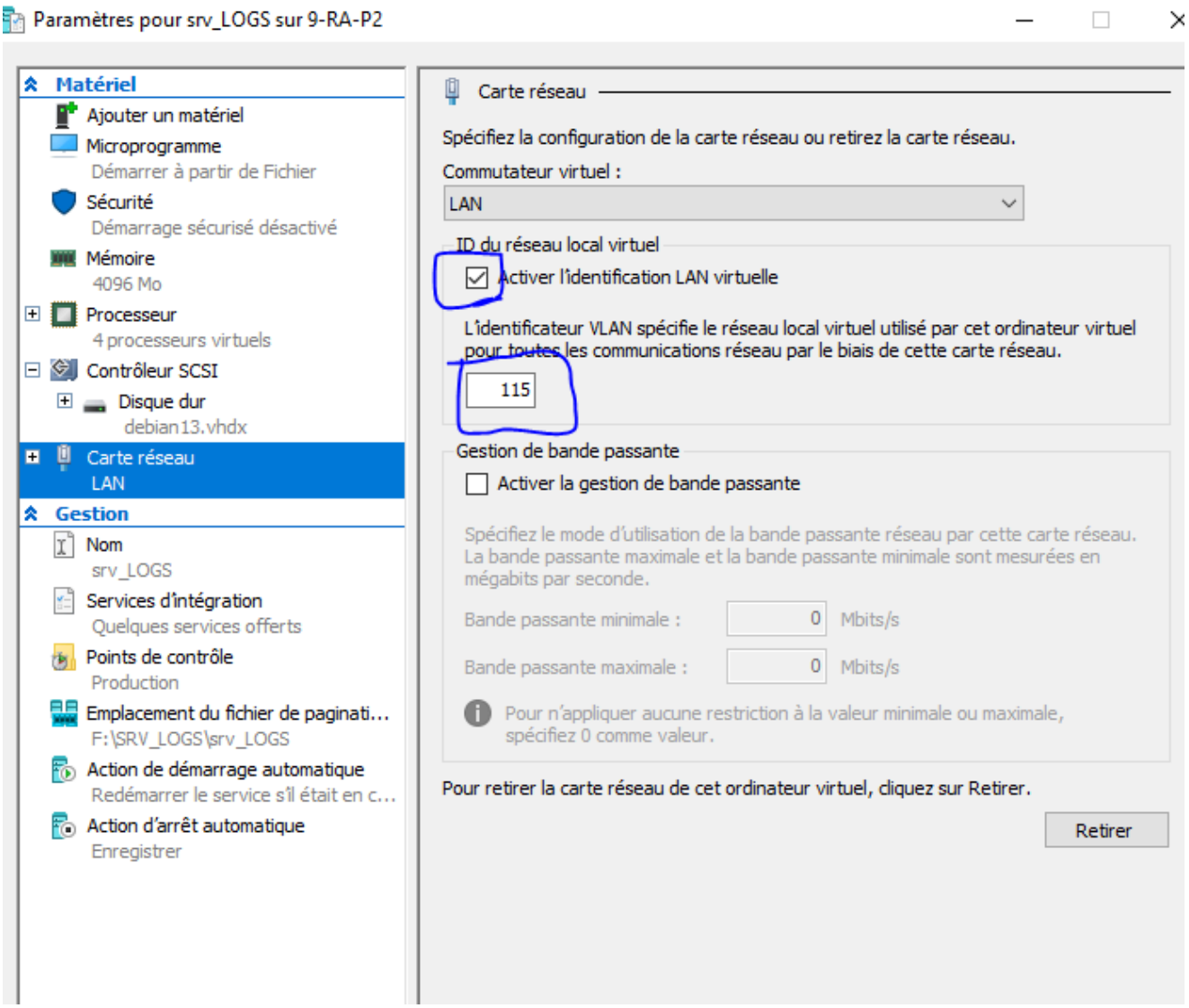
connexion reseau



passerelle



srv de logs dans le vlan it = 115



Installation apache 2

A. Installer Apache sous Debian 13

Nous commençons par mettre à jour le cache des paquets :

```
sudo apt update
```

Ensuite, nous installons le paquet **apache2** afin d'obtenir la dernière version d'Apache 2.4.

```
sudo apt install -y apache2
```

Pour qu'Apache démarre automatiquement au démarrage de la machine, saisissez la commande ci-dessous (même si normalement c'est déjà le cas) :

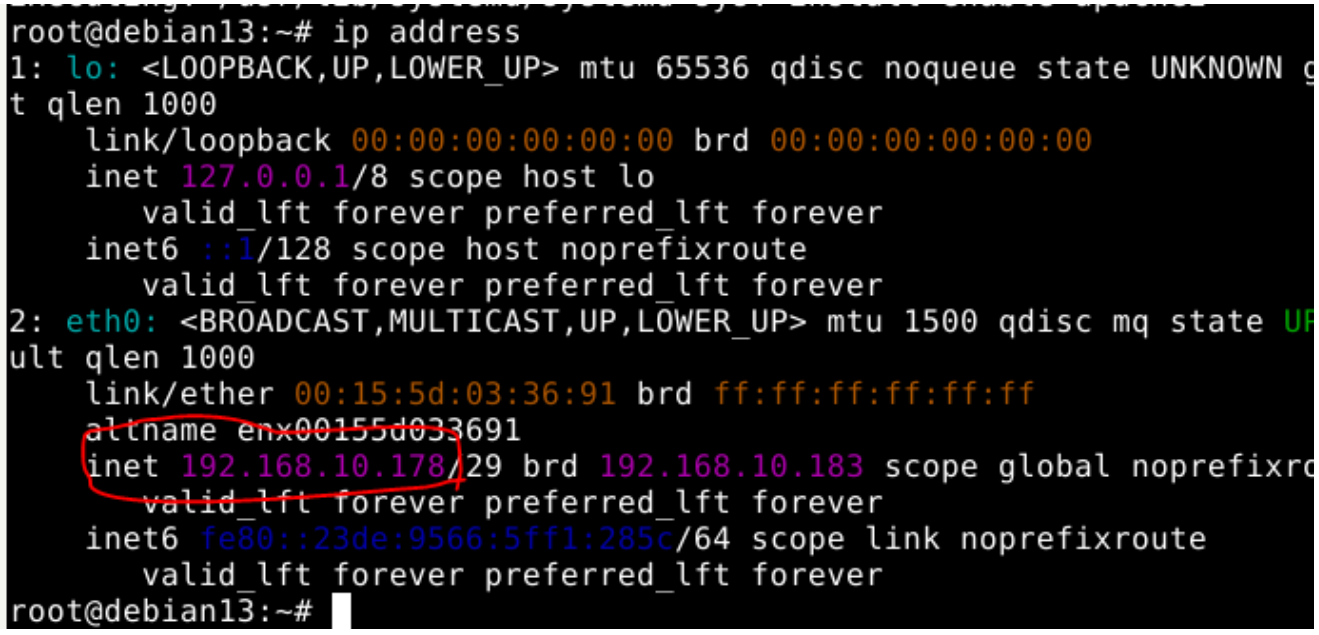
```
sudo systemctl enable apache2

Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
```

Suite à l'installation du paquet, le serveur Apache démarre directement. D'ailleurs, nous devrions pouvoir accéder à la page par défaut du serveur Web Apache2. Pour cela, il suffit de récupérer l'adresse IP du serveur :

```
ip address
```

Puis, à l'aide d'une machine équipée d'un navigateur, nous pouvons accéder à notre serveur Apache

ip du debian 13

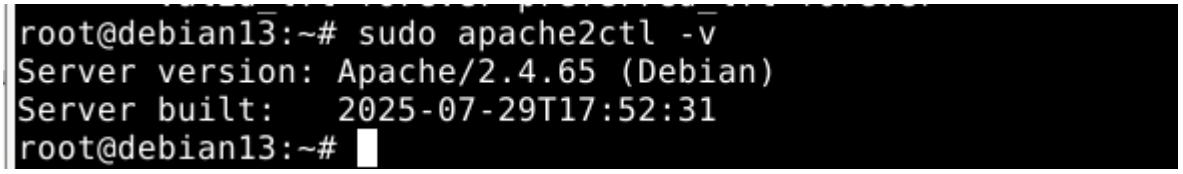


sur une navigateur web

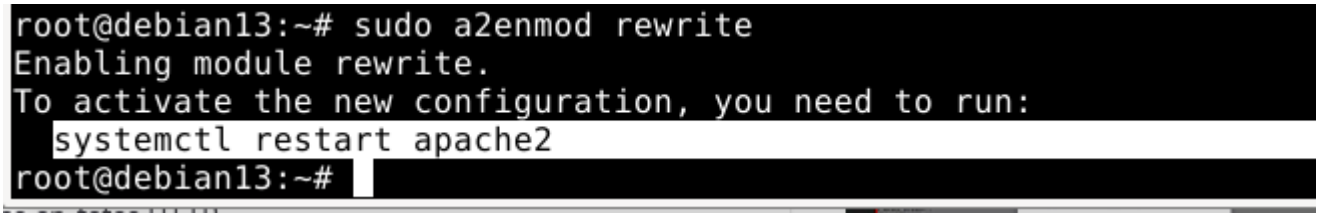
http://192.168.10.178 pour aller sur l'interface graph du debian

pour visualiser apache qu'on viens d'installer

avec la commande sudo apache2ctl -v



activate new config



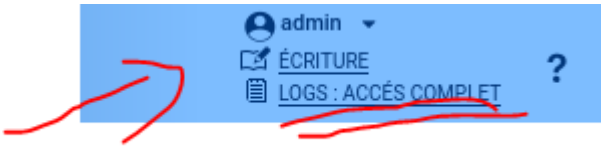
Activons quatre autres modules :

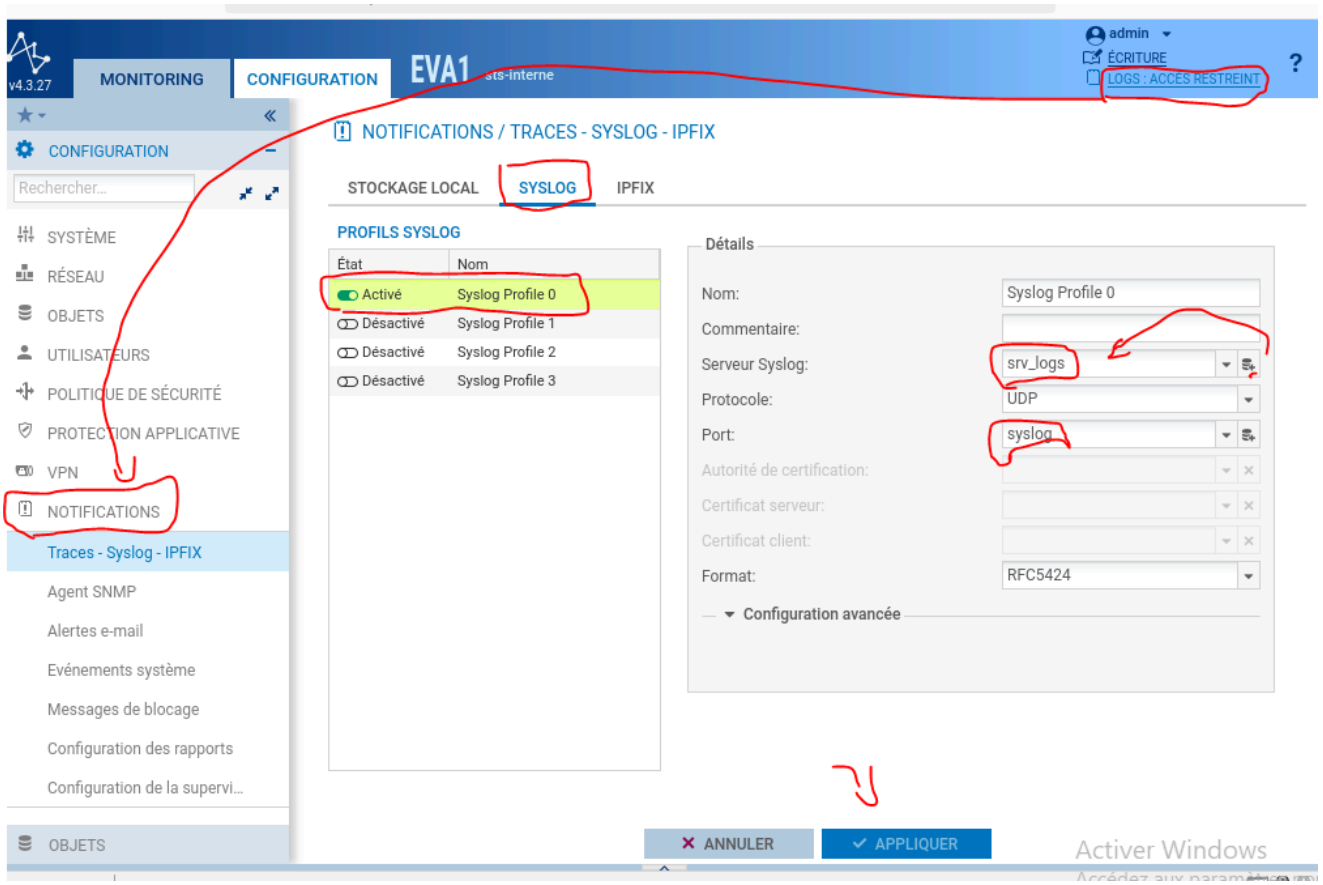
- `deflate` pour la gestion de la compression, notamment en gzip, pour utiliser la mise en cache des pages sur votre site
- `headers` afin de pouvoir agir sur les en-têtes HTTP
- `ssl` pour gérer les certificats SSL et donc l'utilisation du protocole HTTPS
- `http2` pour gérer les connexions HTTP/2

```
sudo a2enmod deflate headers ssl http2
```

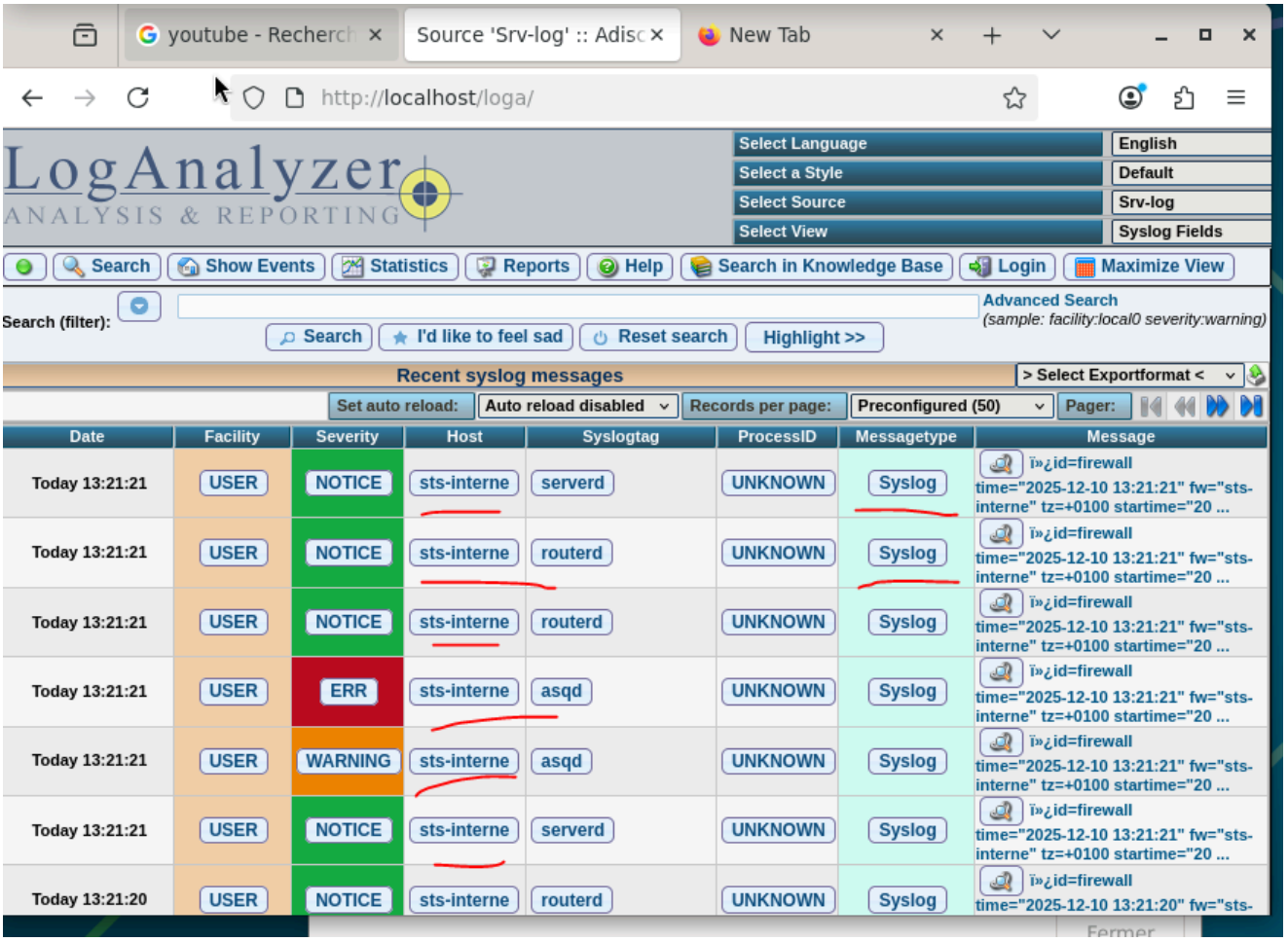
```
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure S
elf-signed certificates.
Enabling module http2.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian13:~# sudo a2enmod rewrite
Module rewrite already enabled
root@debian13:~# sudo a2enmod deflate headers ssl http2
Considering dependency filter for deflate:
Module filter already enabled
Module deflate already enabled
Module headers already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
Module http2 already enabled
root@debian13:~#
```

POUR joinde le stormshield interne aux serveur de logs
ip srv de logs = 192.168.10.178
sts_interne=172.16.10.250



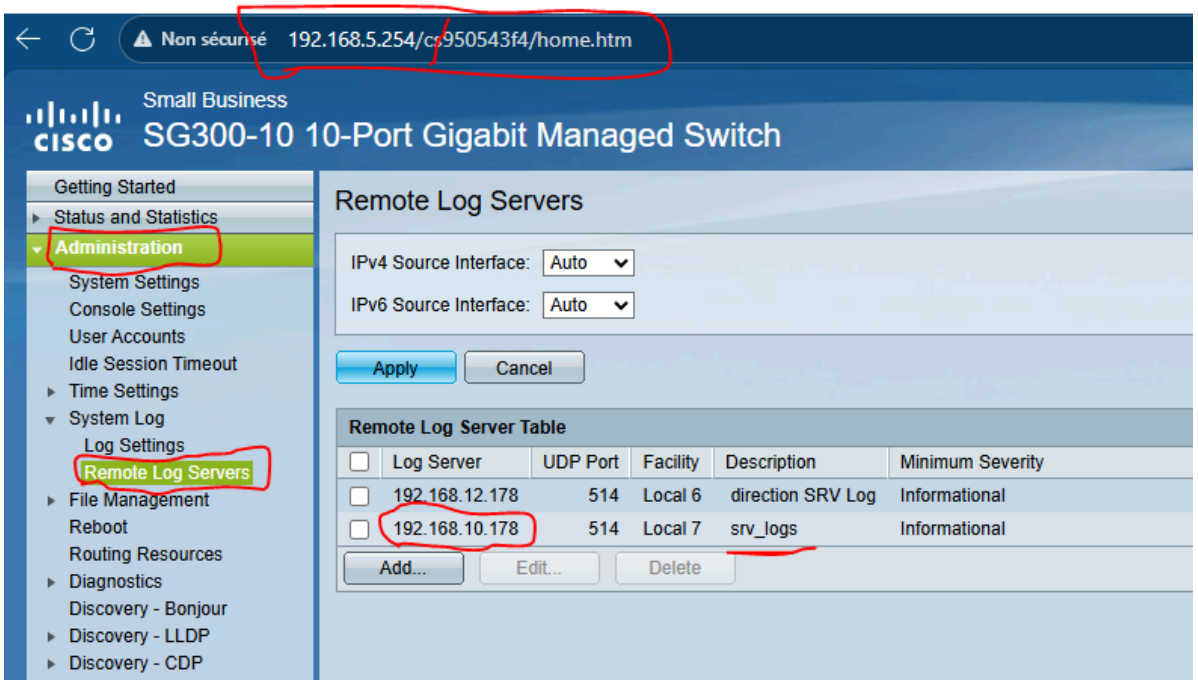


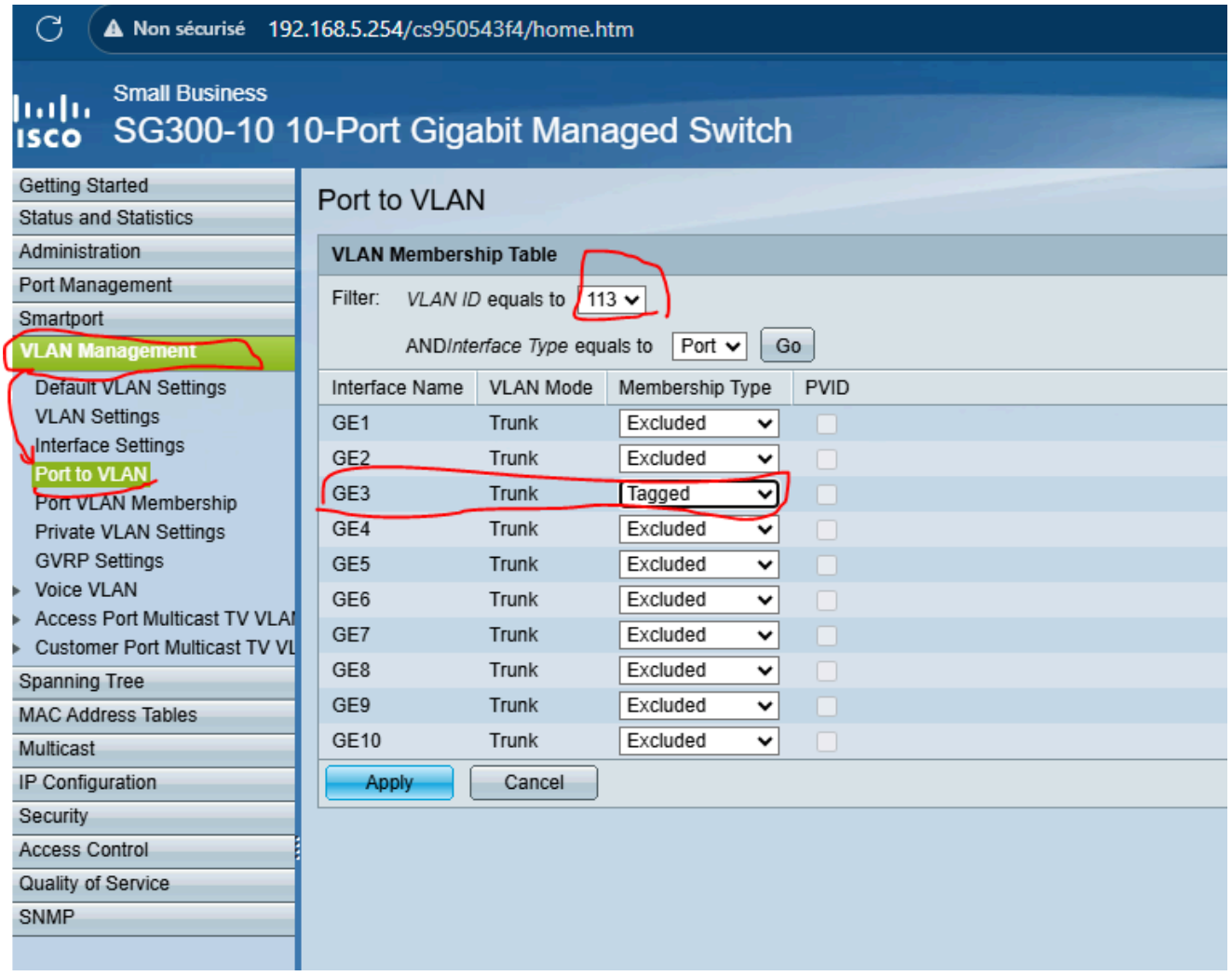
sur le srv de logs apres la commutation



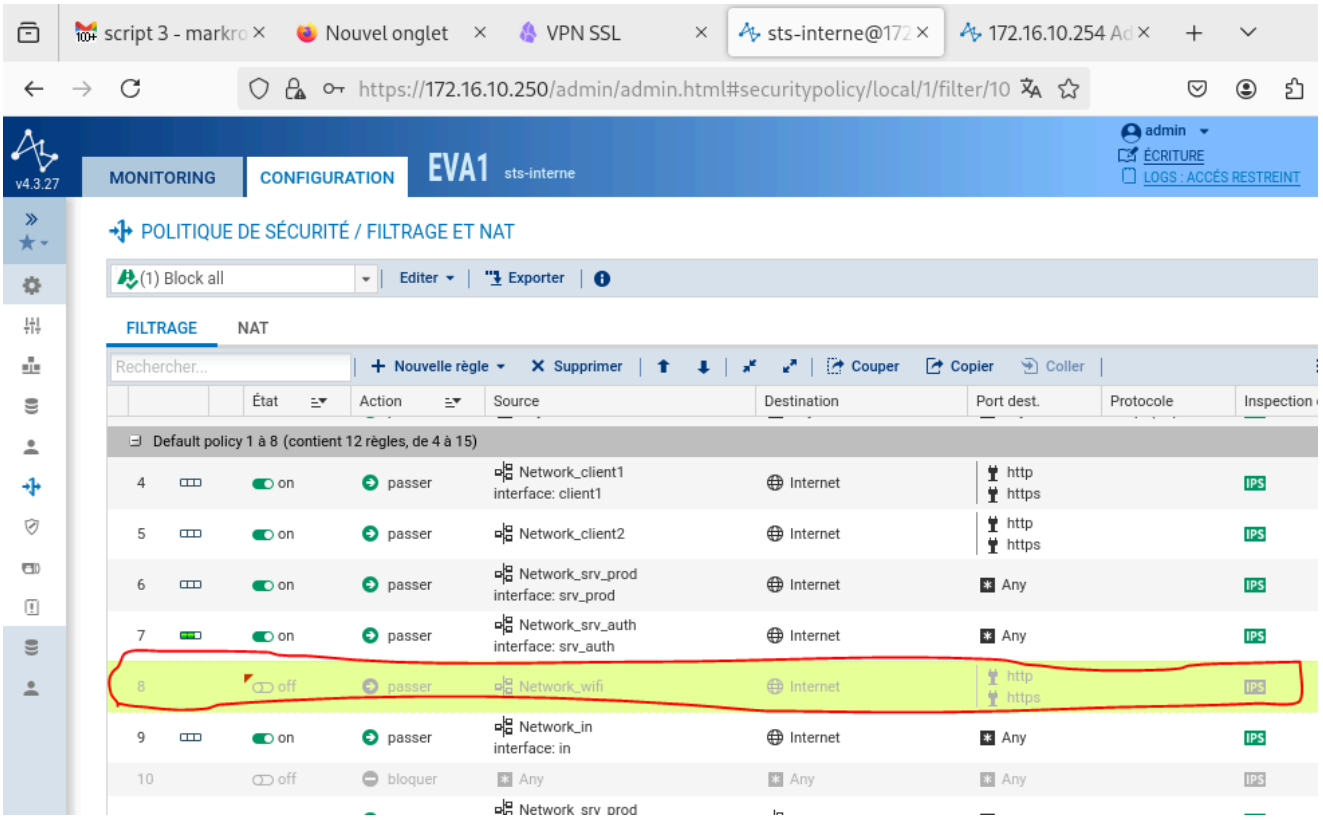
switch cisco

Pour connecter son cisco au srv_logs

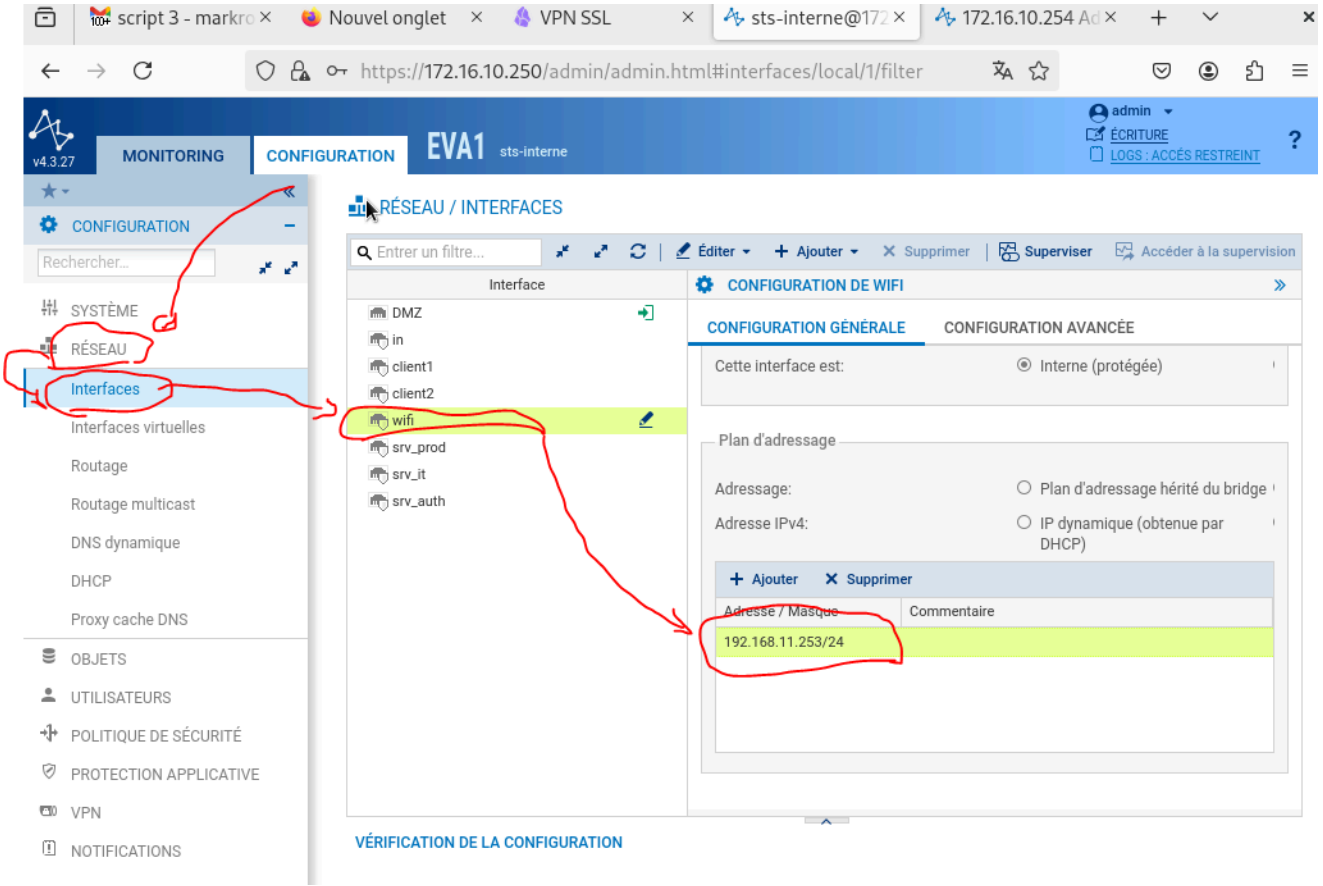
- Faut laisser une carte reseau sans vlan
-
  - 



puis sur le sts interne
on desactive la ligne wifi

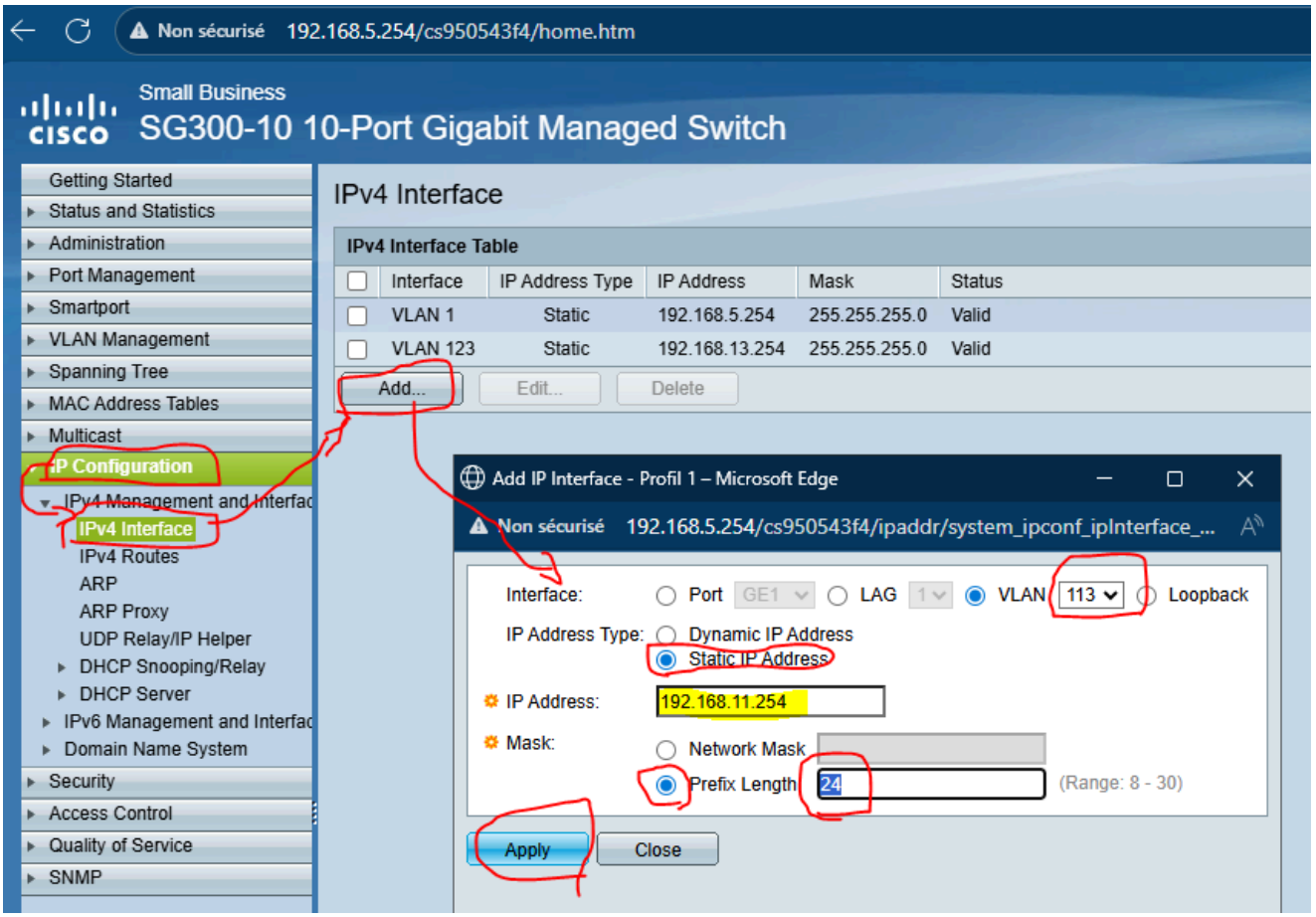


dans reseau sur sur sts interne
on change ladress ip au 192.168.11.253/24

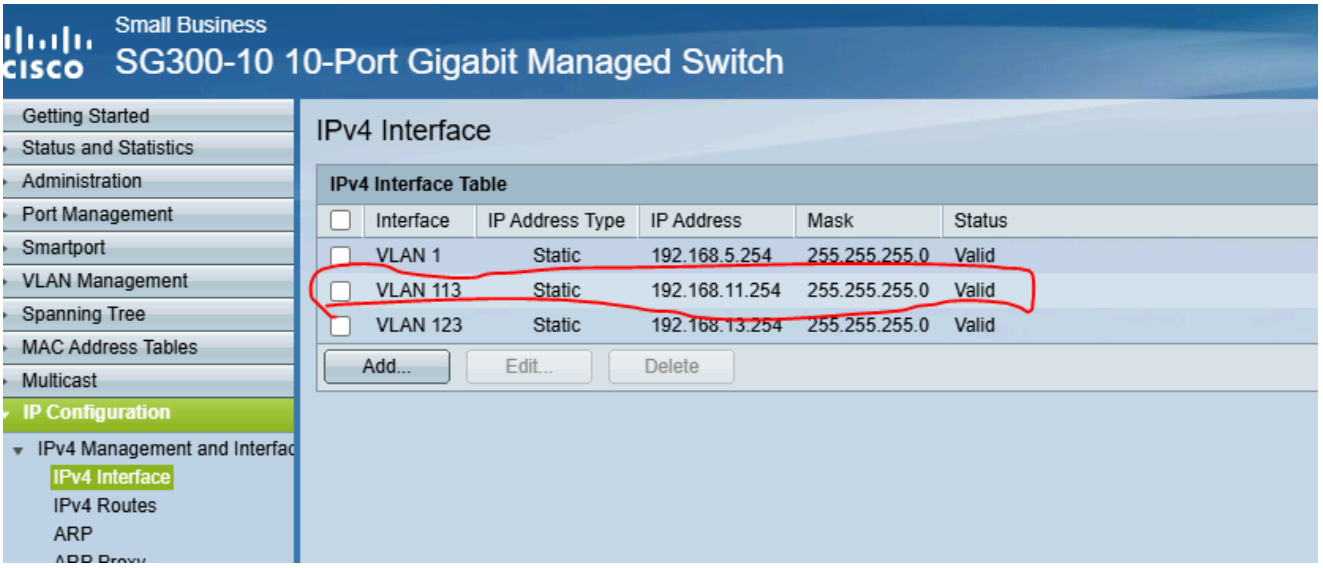


encoere sur le cisco

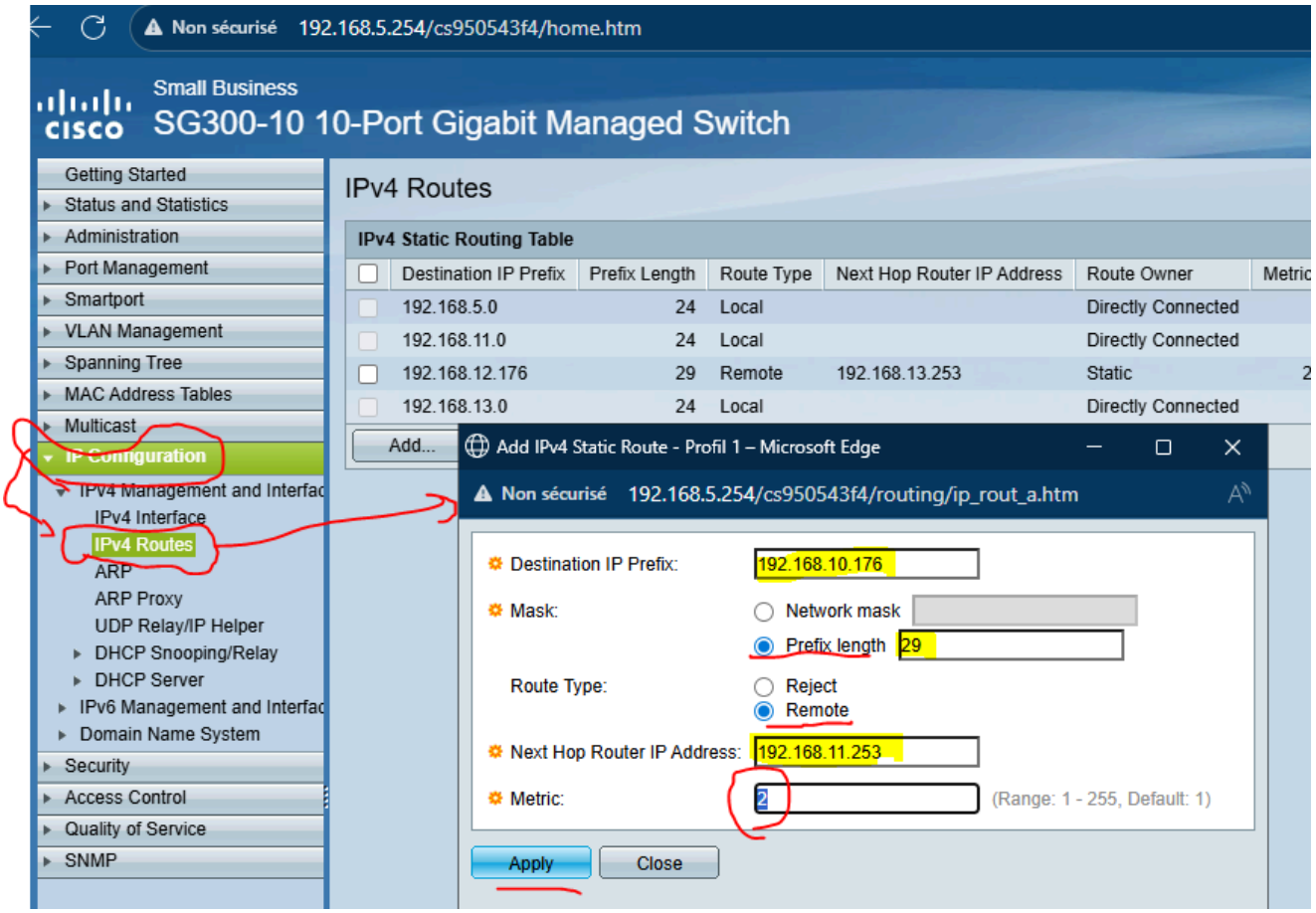
ipv4 interface



resultat

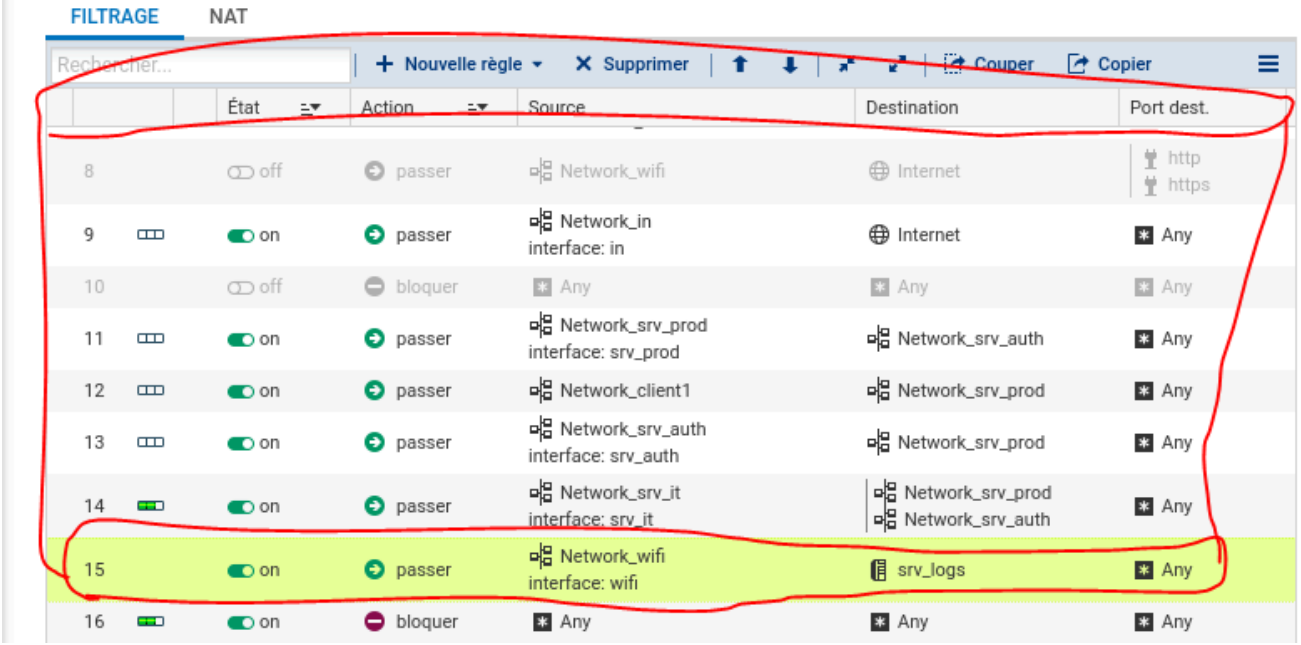


ipv4 routes



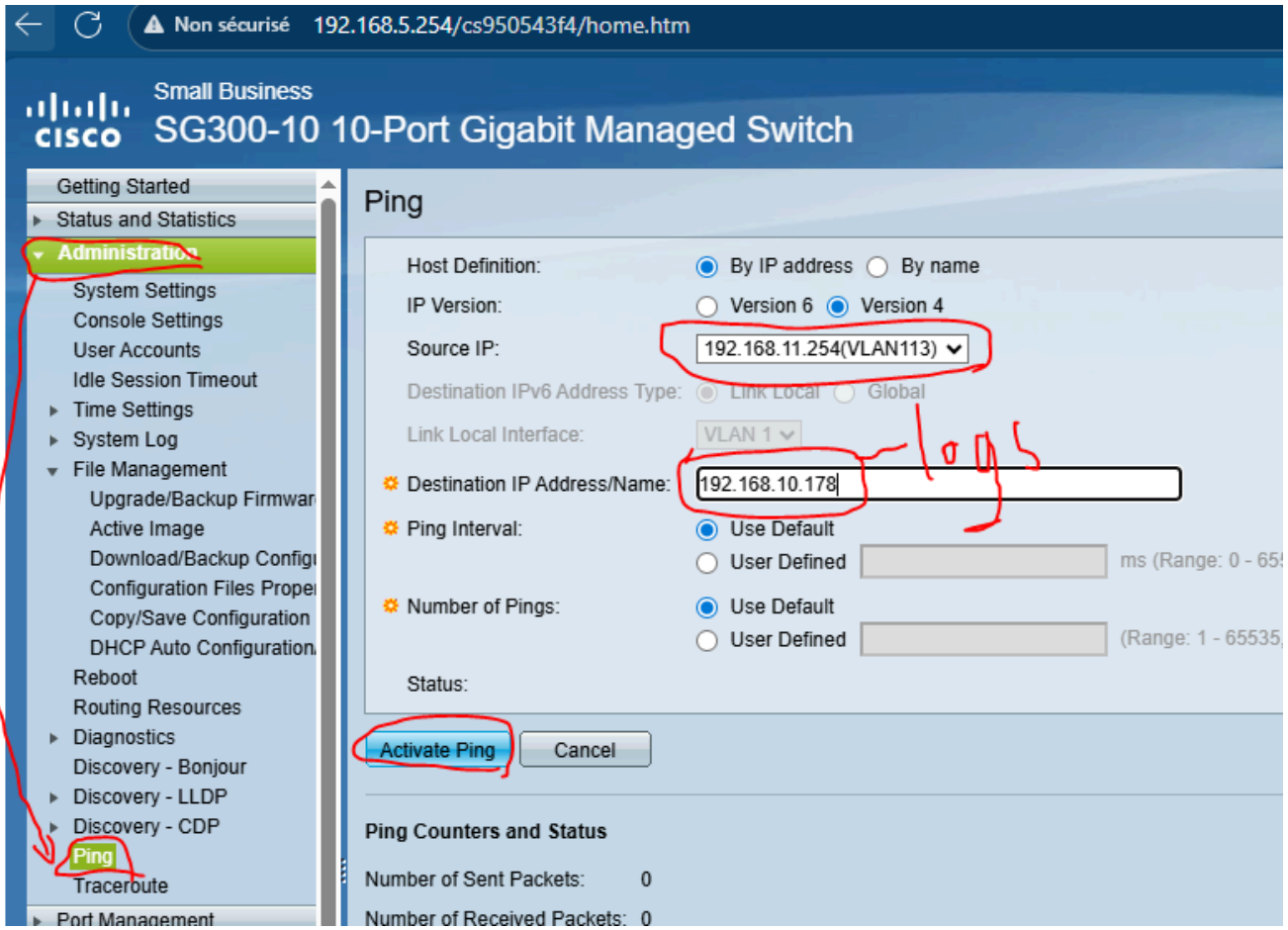
sur le sts interne

faut ajouter une regle encore



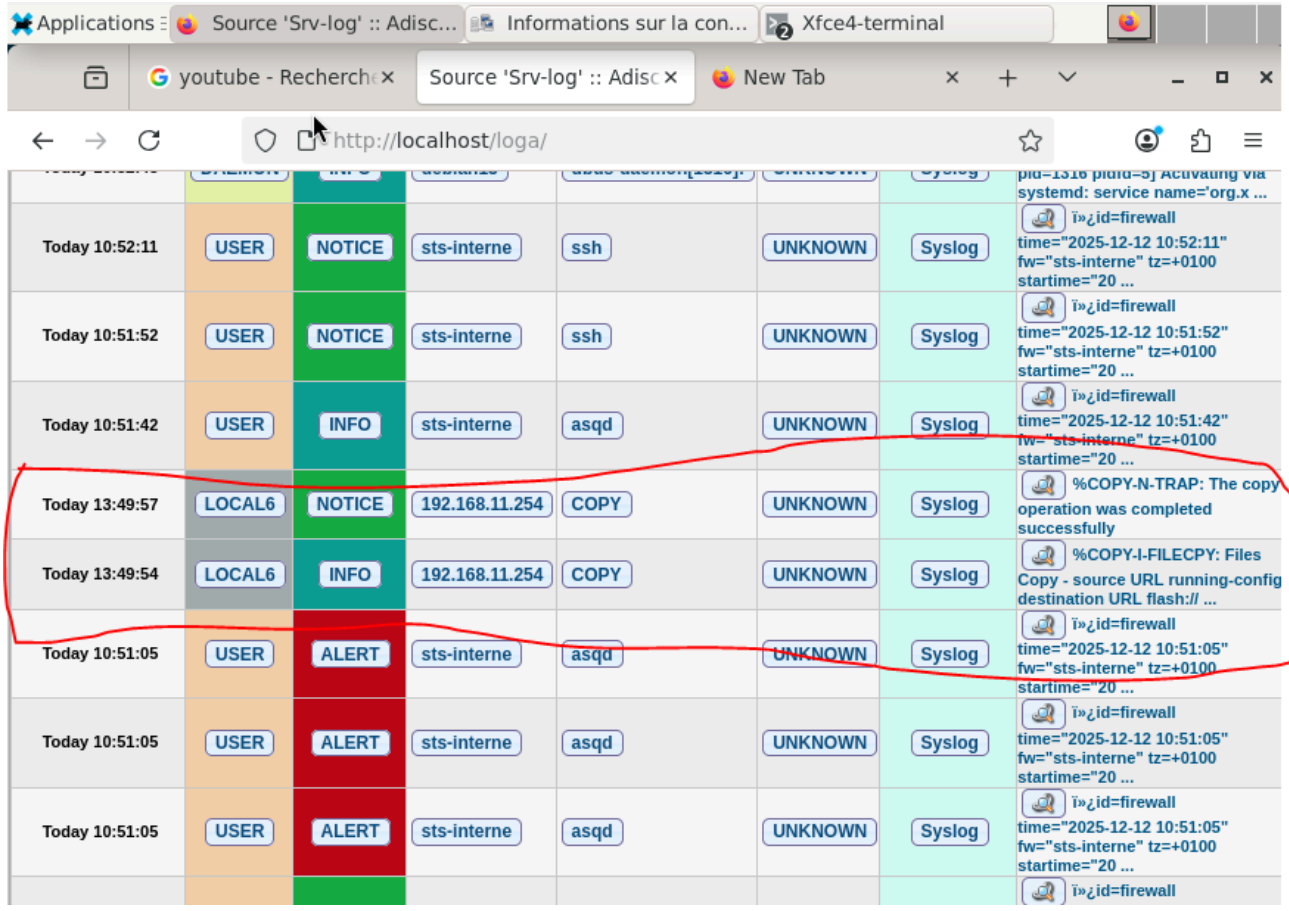
sur le csico

faut activer les pings pour etablir une connexion au ciscosu srv de logs



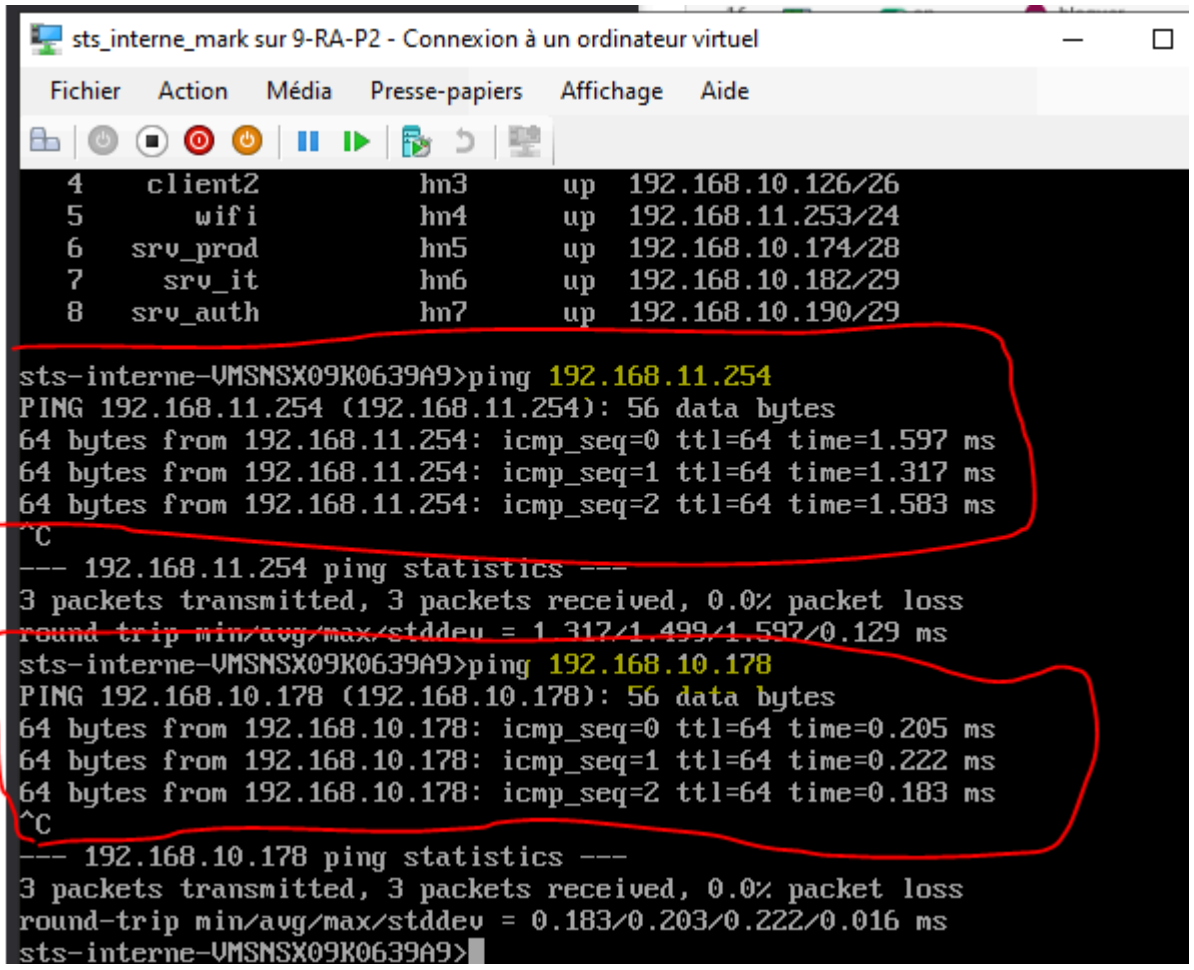
sur srv_logs

on peut voir les logs de cisco avec son ip



sur le sts_interne

on peut voirles pings reussir entre les cisco et sr_iogs



Conclusion

SRV de Log opérationnel.

*Log de notre Stormshield interne à notre SRV Log

Log de notre Switch Cisco à notre SRV Log