

Compte-rendu

Contexte :

La société Amine nous a demandé d'intervenir afin de mettre en place un système de surveillance sur le réseau. Ce réseau contiendra 2 LAN avec 2 routeurs.

Nous pourrons le configurer afin d'avoir plusieurs fonctionnalités

Sommaire:

- 1 - Installation de LibreNMS et configuration de la stations windows 10
- 2 - Installation de 2 routeurs Pfsense
- 3 - Configuration de Pfsense
- 4 - Configuration de Librenms

Prérequis :

- Deux routeur Pfsense
- Serveur Ubuntu 22.04 LTS
- Station Windows 10 Pro

Explication :

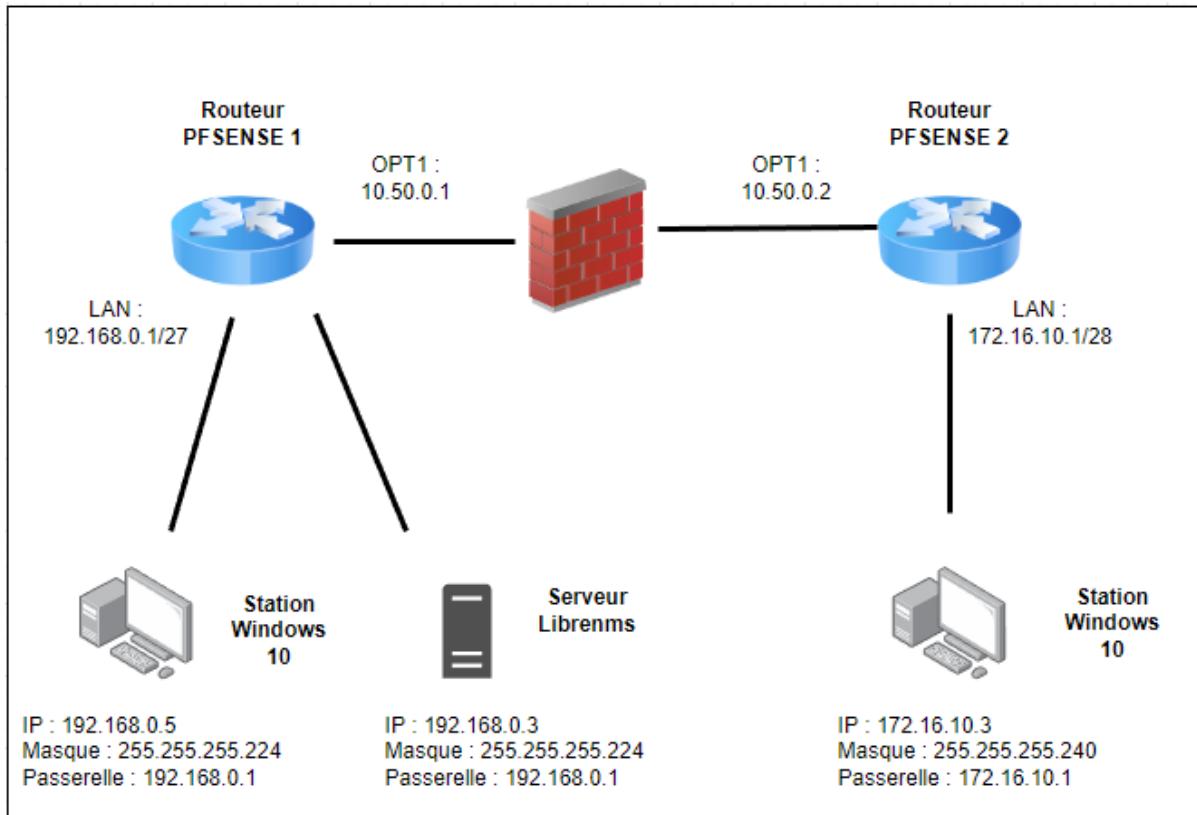
Tout d'abord, nous allons installer LibreNMS sur un serveur Ubuntu en version 22.04.

Ensuite, nous allons installer Windows 10 sur deux stations avec une configuration de la carte réseau différente pour les deux machines.

Puis, on installera deux routeurs Pfsense avec une configuration pour les deux sous réseaux.

Enfin, nous allons configurer LibreNMS avec plusieurs outils et gadgets très intéressants dont des graphes.

Schéma de la mission :



Tutoriel :

1- Installation de Librenms et configuration des stations en windows 10

```
serverlinux@serverlinux: ~
serverlinux@serverlinux:~$ sudo apt install php7.4 php7.4-common libapache2-mod-php7.4 php7.4-cli
```

Tout d'abord, on installe php ainsi que ses modules.

```
serverlinux@serverlinux: ~
serverlinux@serverlinux:~$ serverlinux@serverlinux:~$ sudo apt install apache2
```

On installe ensuite apache2 pour avoir l'interface de librenms.

```
serverlinux@serverlinux: ~
serverlinux@serverlinux:~$ sudo apt install software-properties-common
serverlinux@serverlinux: ~
serverlinux@serverlinux:~$ sudo add-apt-repository universe
root@serverlinux: /home/serverlinux
root@serverlinux:/home/serverlinux# apt update
root@serverlinux:~# apt install acl curl apache2 composer fping git graphviz imagemagick libapache2-mod-fcgid mariadb-client mariadb-s^
server mtr-tiny nmap php7.4-cli php7.4-curl php7.4-fpm php7.4-gd php7.4-gmp php7.4-json php7.4-mbstring php7.4-mysql php7.4-snmp php7.4-xml php7.4-zip nrdd
ool snmp snmpd whois python3-pymysql python3-dotenv python3-redis python3-setuptools python3-systemd python3-pip
```

On vient de télécharger tous les packages nécessaires pour installer librenms.

```
root@serverlinux: ~
root@serverlinux:~# useradd librenms -d /opt/librenms -M -r -s "$(which bash)"
```

On ajoute l'utilisateur libreNMS.

```
root@serverlinux: /opt
root@serverlinux:~# cd /opt
root@serverlinux:/opt# git clone https://github.com/librenms/librenms.git
```

On télécharge libreNMS.

```
root@serverlinux:/opt
root@serverlinux:/opt# chown -R librenms:librenms /opt/librenms
root@serverlinux:/opt# chmod 771 /opt/librenms
root@serverlinux:/opt# setfacl -d -m g::rwx /opt/librenms/rrd /opt/librenms/logs /opt/librenms/bootstrap/cache/ /opt/librenms/storage/
root@serverlinux:/opt# setfacl -R -m g::rwx /opt/librenms/rrd /opt/librenms/logs /opt/librenms/bootstrap/cache/ /opt/librenms/storage/
```

On modifie les droits et les permissions pour l'installation de libreNMS.

```
Sélection root@serverlinux: /opt
root@serverlinux:/opt# su - librenms
/librenms@serverlinux:/opt# /scripts/composer_wrapper.php install --no-dev
exitlibrenms@serverlinux:~$ ./scripts/composer_wrapper.php install --no-dev
```

On installe les dépendances php.

```
root@serverlinux: /opt
root@serverlinux:/opt# wget https://getcomposer.org/composer-stable.phar
root@serverlinux:/opt# mv composer-stable.phar /usr/bin/composer
root@serverlinux:/opt# chmod +x /usr/bin/composer
```

On fait ça aussi pour installer composer.

```
root@serverlinux:/opt# sudo nano /etc/php/7.4/fpm/php.ini
```

```
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
;date.timezone = Europe/Paris
```

On modifie le fichier php pour modifier l'heure de la zone en heure de Paris

```
root@serverlinux:/opt# sudo nano /etc/php/7.4/cli/php.ini
```

```
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
;date.timezone = Europe/Paris
```

On fait cela pour les deux fichiers

```
root@serverlinux:/opt# timedatectl set-timezone Europe/Paris
```

On fait cela pour le système.

```
root@serverlinux:/opt# sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

```
[mysqld]
#
# * Basic Settings
#
#user          = mysql
pid-file      = /run/mysqld/mysqld.pid
basedir       = /usr
#datadir      = /var/lib/mysql
#tmpdir        = /tmp
innodb_file_per_table=1
lower_case_table_names=0
```

On modifie la configuration de mysql en ajoutant ses deux lignes .

```
root@serverlinux:/opt# systemctl enable mariadb
root@serverlinux:/opt# systemctl restart mariadb
```

On active et redémarre mariadb pour prendre en compte ce qu'on a ajouté.

```
root@serverlinux:/opt# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or '\g'.
Your MariaDB connection id is 31
Server version: 10.6.7-MariaDB-2ubuntu1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE librenms CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER 'lQuery'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON librenms.* TO 'lQuery'@'localhost'Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER 'librenms'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON librenms.* TO 'librenms'@'localhost'Query OK, 0 rows affected (0.002 sec)

;
FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit
Bye
```

On se connecte à la base de données pour ajouter l'utilisateur libreNMS en lui accordant tous les priviléges.

```
root@serverlinux:/opt# cp /etc/php/7.4/fpm/pool.d/www.conf /etc/php/7.4/fpm/pool.d/librenms.conf
```

On copie le fichier de configuration en le collant sous un nom qu'on reconnaîtra qui est libreNMS.

```
root@serverlinux:/opt# sudo nano /etc/php/7.4/fpm/pool.d/librenms.conf
```

```
[librenms]

; Per pool prefix
; It only applies on the following directives:
; - 'access.log'
; - 'slowlog'
; - 'listen' (unixsocket)
; - 'chroot'
; - 'chdir'
; - 'php_values'
; - 'php_admin_values'
; When not set, the global prefix (or /usr) applies instead.
; Note: This directive can also be relative to the global prefix.
; Default Value: none
;prefix = /path/to/pools/$pool

; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the default user's group
;       will be used.
user = librenms
group = librenms

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
;   'ip.add.re.ss:port'      - to listen on a TCP socket to a specific IPv4 address on
;                             a specific port;
;   '[ip:6:addr:ess]:port'  - to listen on a TCP socket to a specific IPv6 address on
;                             a specific port;
;   'port'                  - to listen on a TCP socket to all addresses
;                             (IPv6 and IPv4-mapped) on a specific port;
;   '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
listen = /run/php/php7.4-fpm.sock
listen = /run/php-fpm-librenms.sock
```

On modifie ce fichier qu'on vient de collé et on ajoute l'utilisateur qu'on vient de créer.
On remplace les trois www par libreNMS et on ajoute la ligne listen.

```
root@serverlinux: /opt
root@serverlinux:/opt# sudo nano /etc/apache2/sites-available/librenms.conf
```

```
GNU nano 6.2                                         /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
  DocumentRoot /opt/librenms/html/
  ServerName librenms.example.com

  AllowEncodedSlashes NoDecode
  <Directory "/opt/librenms/html/">
    Require all granted
    AllowOverride All
    Options FollowSymLinks MultiViews
  </Directory>

  # Enable http authorization headers
  <IfModule setenvif_module>
    SetEnvIfNoCase ^Authorization$ "(.*)" HTTP_AUTHORIZATION=$1
  </IfModule>

  <FilesMatch ".+\.\php$">
    SetHandler "proxy:unix:/run/php-fpm-librenms.sock|fcgi://localhost"
  </FilesMatch>
</VirtualHost>
```

On configure le serveur web en ajoutant les lignes pour le fichier de libreNMS.

```
root@serverlinux:/opt
root@serverlinux:/opt# root@serverlinux:/opt# a2dissite 000-default
root@serverlinux:/opt# root@serverlinux:/opt# a2ensite librenms
root@serverlinux:/opt# root@serverlinux:/opt# a2ensite librenms
root@serverlinux:/opt# root@serverlinux:/opt# systemctl restart apache2
root@serverlinux:/opt# root@serverlinux:/opt# systemctl restart php7.4-fpm
root@serverlinux:/opt# root@serverlinux:/opt# To activate the new configuration, you need to run:
root@serverlinux:/opt# root@serverlinux:/opt# systemctl reload apache2
```

Puis on redémarre apache2 pour prendre en compte le fichier configuré.

```
root@serverlinux:/opt
root@serverlinux:/opt# root@serverlinux:/opt# ln -s /opt/librenms/lnms /usr/bin/lnms
root@serverlinux:/opt# root@serverlinux:/opt# ln -s /opt/librenms/misc/lnms-completion.bash /etc/bash_completion.d/
root@serverlinux:/opt# root@serverlinux:/opt# cp /opt/librenms/misc/lnms-completion.bash /etc/bash_completion.d/
```

C'est pour compléter les commandes lnms comme les commandes linux.

```
root@serverlinux:/opt
root@serverlinux:/opt# root@serverlinux:/opt# cp /opt/librenms/snmpd.conf.example /etc/snmp/snmpd.conf
root@serverlinux:/opt#
```

```
root@serverlinux:/opt
GNU nano 6.2
/etc/snmp/snmpd.conf *
# Change RANDOMSTRINGGOESHERE to your preferred SNMP community string
com2sec readonly default librenms

group MyROGroup v2c      readonly
view all included .1          80
access MyROGroup ""      any      noauth      exact  all      none      none

syslocation Rack, Room, Building, City, Country [Lat, Lon]
syscontact Your Name <your@email.address>

#OS Distribution Detection
extend distro /usr/bin/distro

#Hardware Detection
# (uncomment for x86 platforms)
#extend manufacturer '/bin/cat /sys/devices/virtual/dmi/id/sys_vendor'
#extend hardware '/bin/cat /sys/devices/virtual/dmi/id/product_name'
#extend serial '/bin/cat /sys/devices/virtual/dmi/id/product_serial'

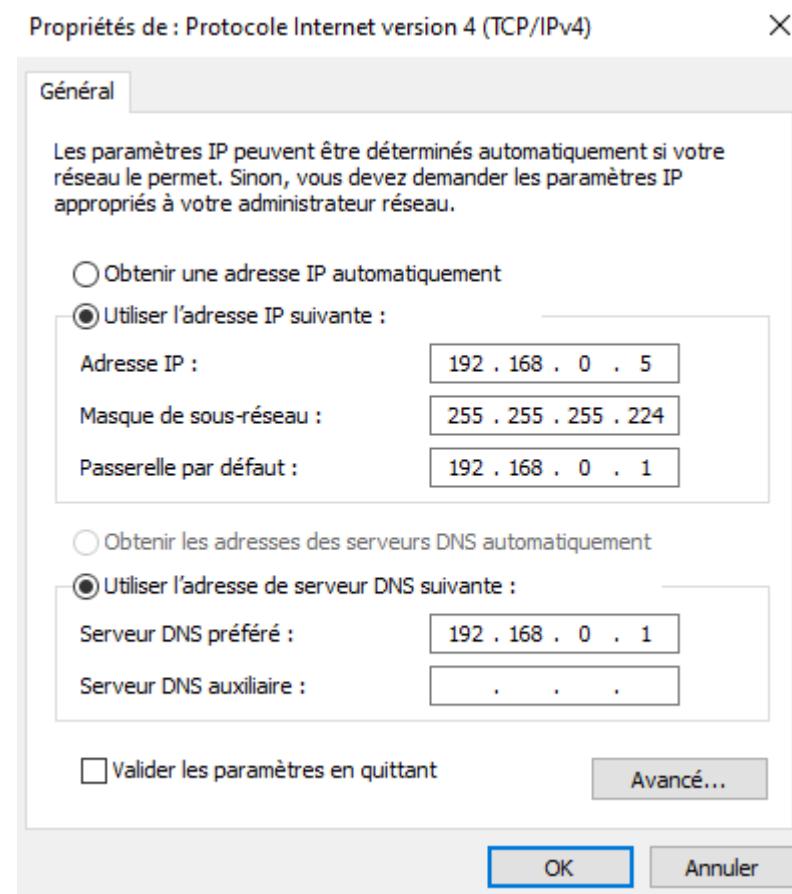
# (uncomment for ARM platforms)
#extend hardware '/bin/cat /sys/firmware/devicetree/base/model'
#extend serial '/bin/cat /sys/firmware/devicetree/base/serial-number'
```

On modifie **RANDOMSTRINGGOESHERE** par librenms.

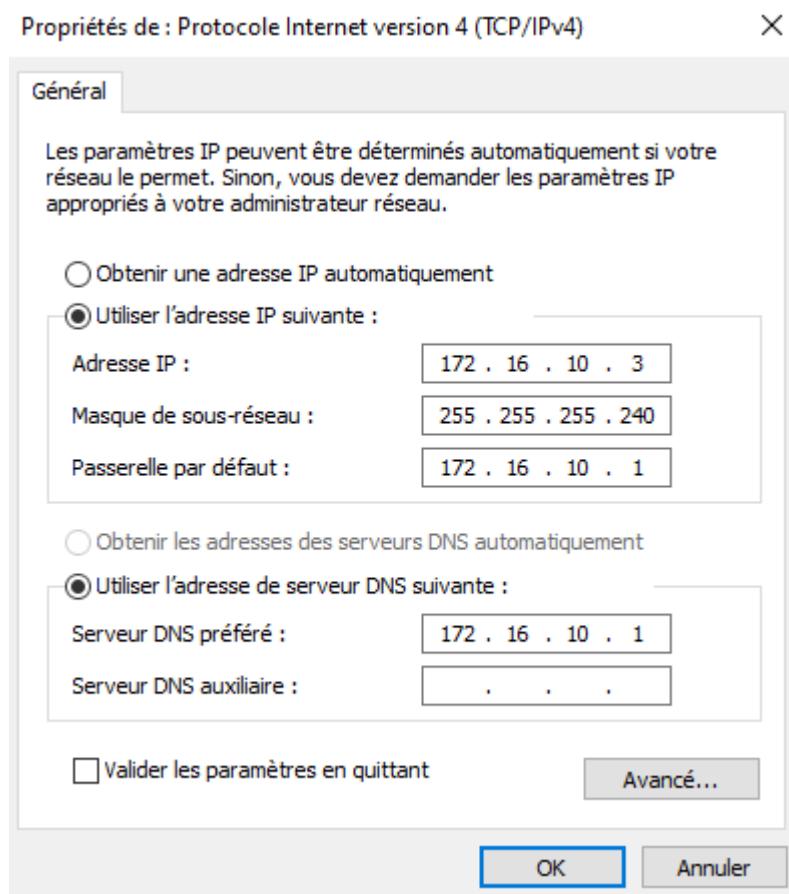
```
root@serverlinux:/opt#
root@serverlinux:/opt# curl -o /usr/bin/distro https://raw.githubusercontent.com/librenms/librenms-agent/master/snmp/distro
x /usr/bin/distro
systemctl enable snmpd
systemctl restart snmpd
```

On active et redémarre snmpd une fois snmpd installé.

On met en place deux stations avec Windows 10 pro et on configure le réseau :



On met une ip qui est dans la plage de notre réseau du routeur 192.168.0.1 et on met aussi cet ip dans la passerelle pour ensuite se relier au deuxième routeur. (PC 1)



Idem pour ce pc qui sera dans un autre réseau lan.

Pare-feu Windows Defender

Page d'accueil du panneau de configuration

Autoriser une application ou une fonctionnalité via le Pare-feu Windows Defender

Modifier les paramètres de notification

Activer ou désactiver le Pare-feu Windows Defender

Paramètres par défaut

Paramètres avancés

Dépanner mon réseau

Voir aussi

Sécurité et maintenance

Centre Réseau et partage

Protégez votre ordinateur avec le Pare-feu Windows Defender

Le Pare-feu Windows Defender a pour but d'empêcher les pirates ou les logiciels malveillants d'accéder à votre ordinateur via Internet ou via un réseau.

Mettre à jour les paramètres du pare-feu

Le Pare-feu Windows Defender n'utilise pas les paramètres recommandés pour protéger votre ordinateur.

Quels sont les paramètres recommandés ?

Réseaux privés Non connecté

Réseaux publics ou invités Connecté

Réseaux dans des lieux publics, tels qu'un aéroport ou un cybercafé

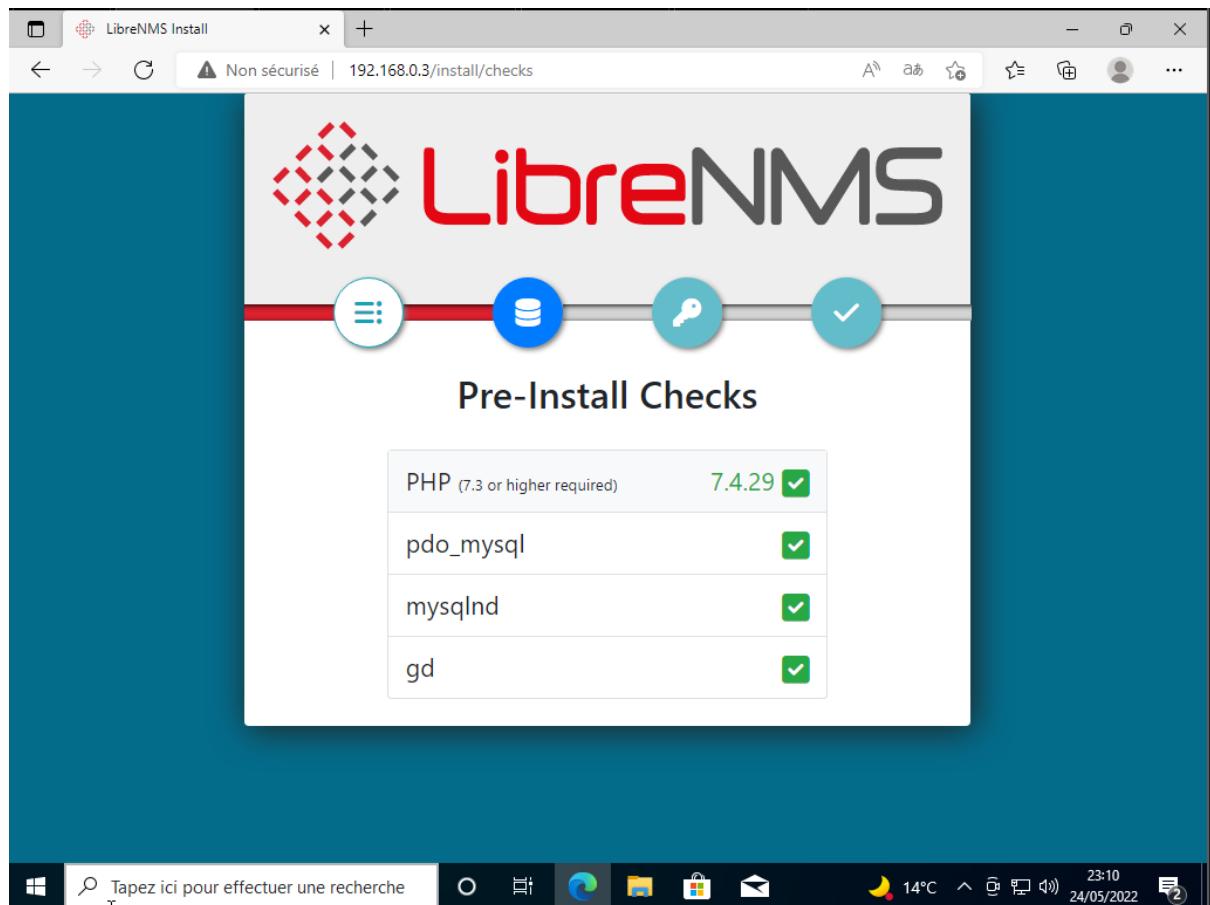
État du Pare-feu Windows Defender : Désactivé

Connexions entrantes : Bloquer toutes les connexions aux applications ne figurant pas dans la liste des applications autorisées

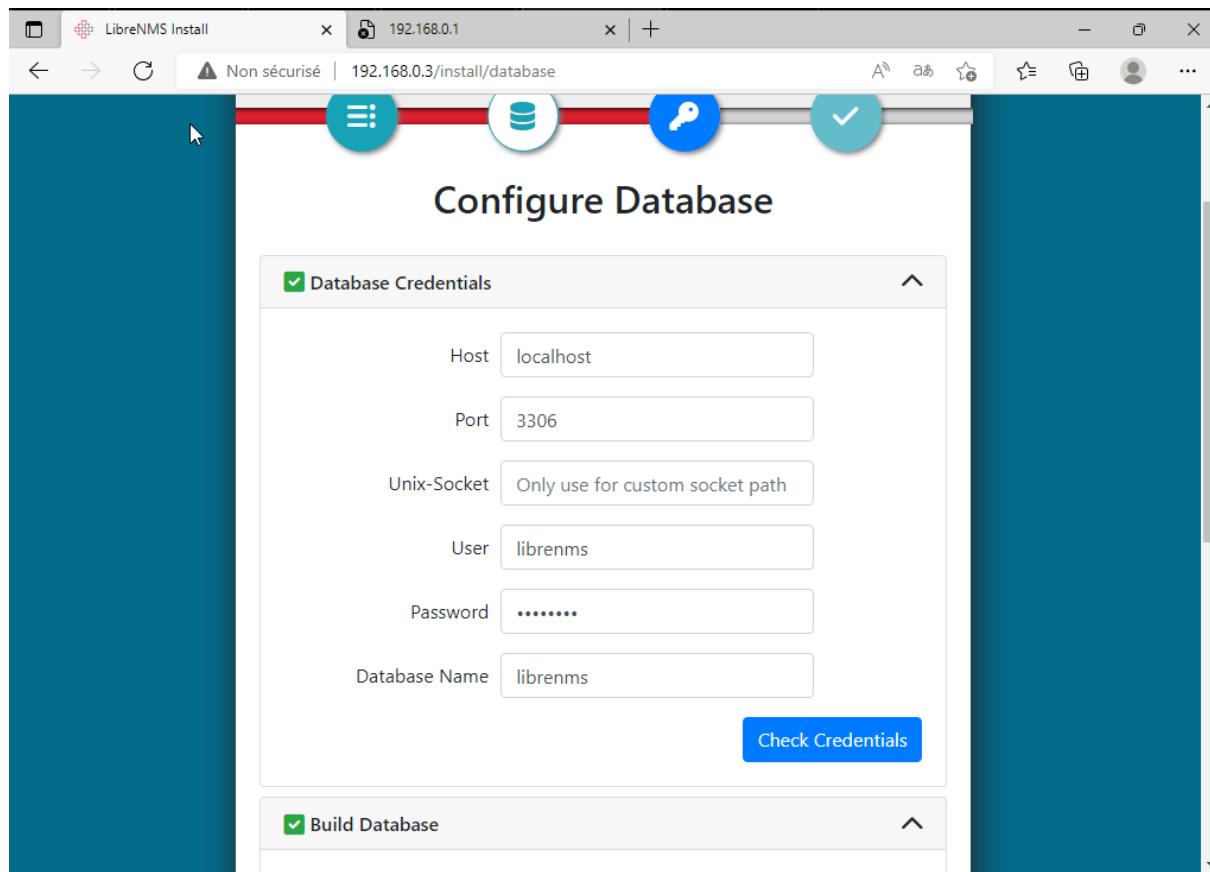
Réseaux publics actifs : Réseau non identifié

État de notification : M'avertir lorsque le Pare-feu Windows Defender bloque une nouvelle application

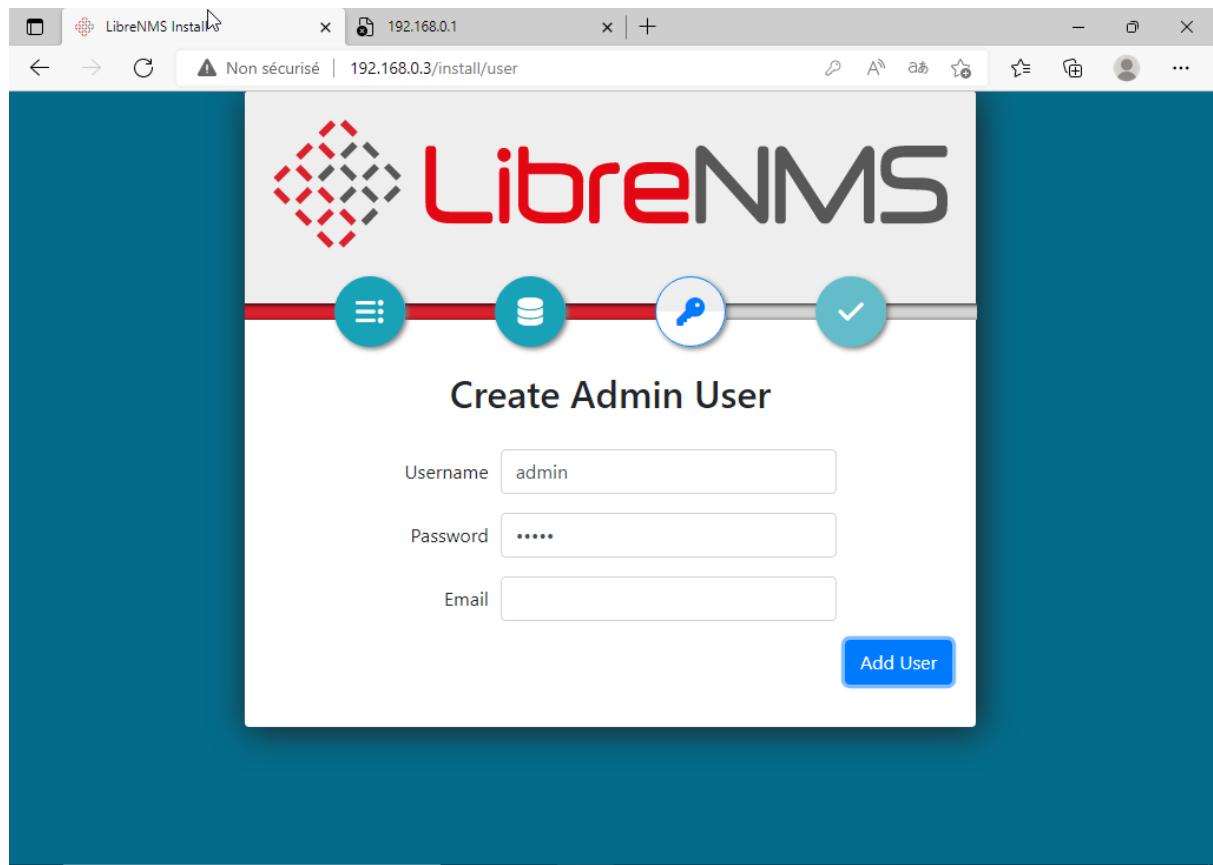
On désactive le pare-feu windows pour les deux pc pour qu'il ne bloque pas les paquets venant d'autrui.



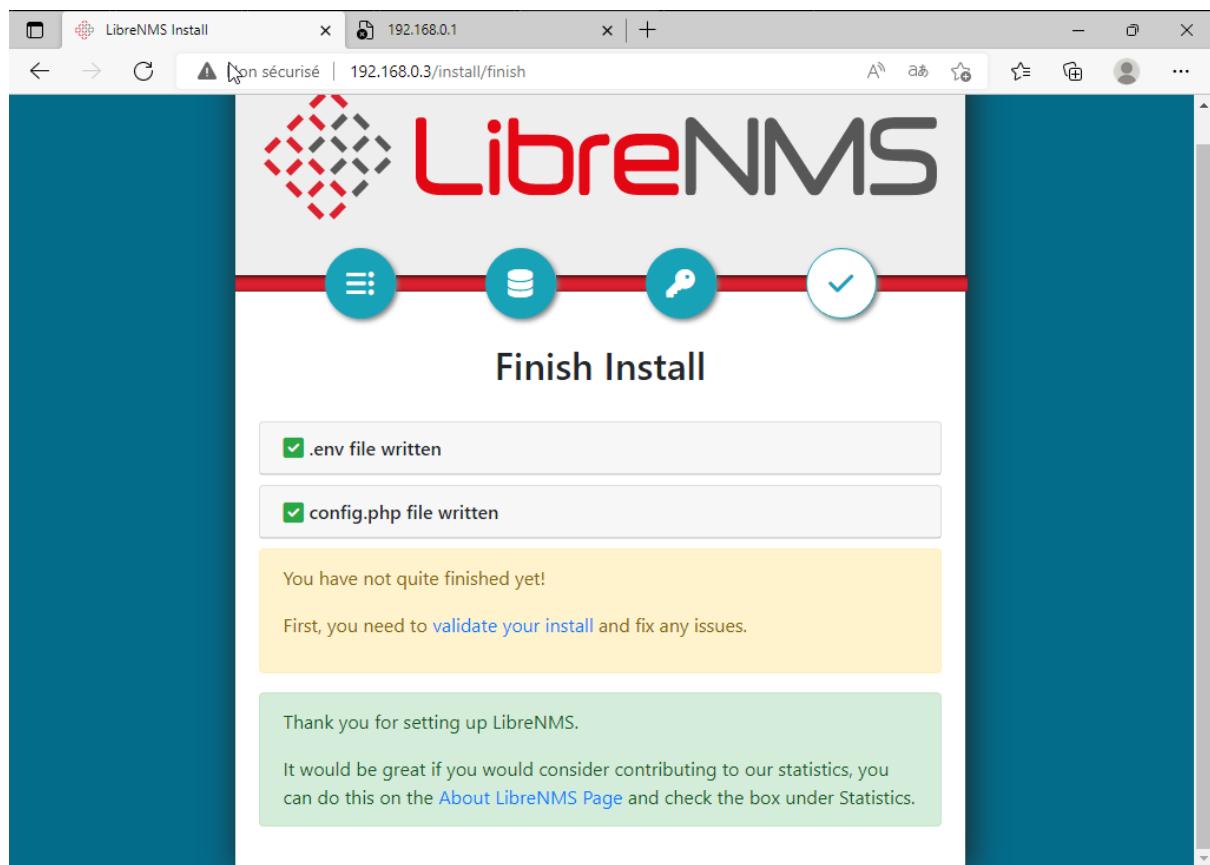
On fait une vérification des pré installations des différents packages.



Ensuite on se connecte avec le compte qu'on a créé dans la base de donné mysql et on crée la database.

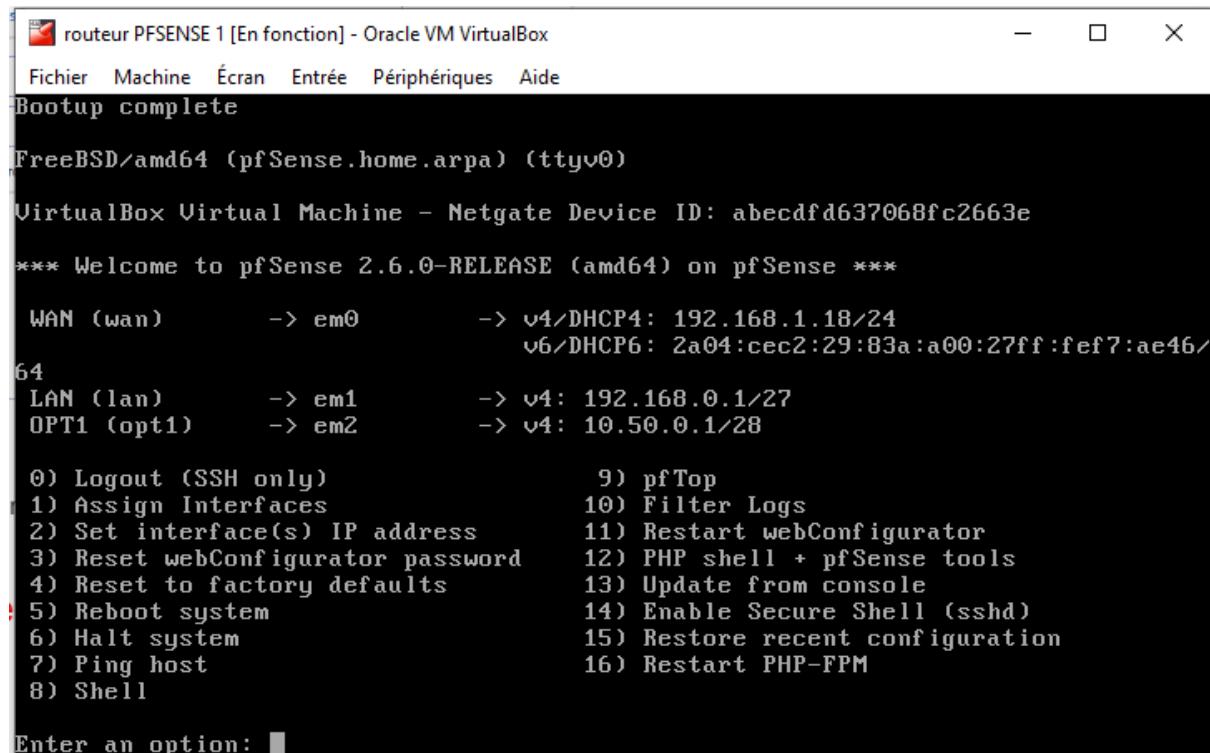


On crée le compte admin pour se connecter dessus.

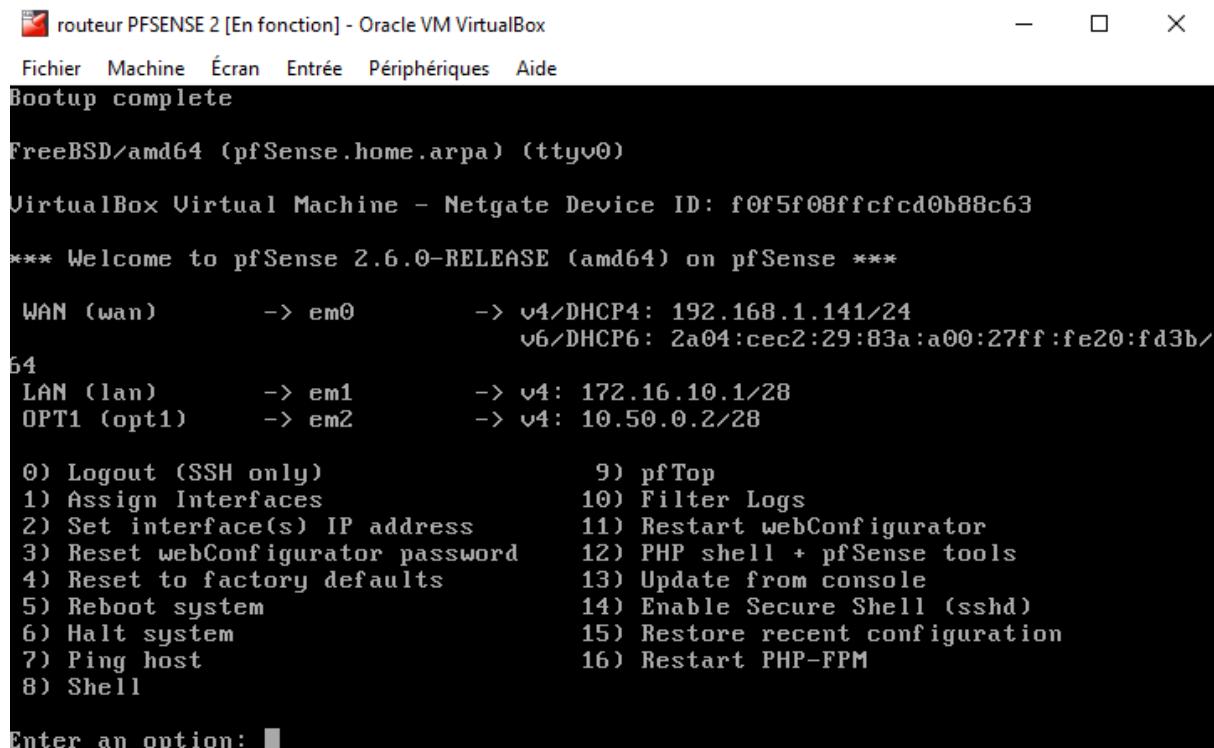


Et on vient de finir l'installation.

2- Installation de deux routeurs Pfsense



Activer trois interfaces réseau en les ajoutant avec 2 LAN (LAN et OPT1).
OPT1 permettra de faire la passerelle entre le premier routeur et le deuxième.



```

routeur PFSENSE 2 [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
Bootup complete

FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: f0f5f08ffcfcd0b88c63

*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 192.168.1.141/24
                      v6/DHCP6: 2a04:cec2:29:83a:a00:27ff:fe20:fd3b/64
LAN (lan)      -> em1      -> v4: 172.16.10.1/28
OPT1 (opt1)    -> em2      -> v4: 10.50.0.2/28

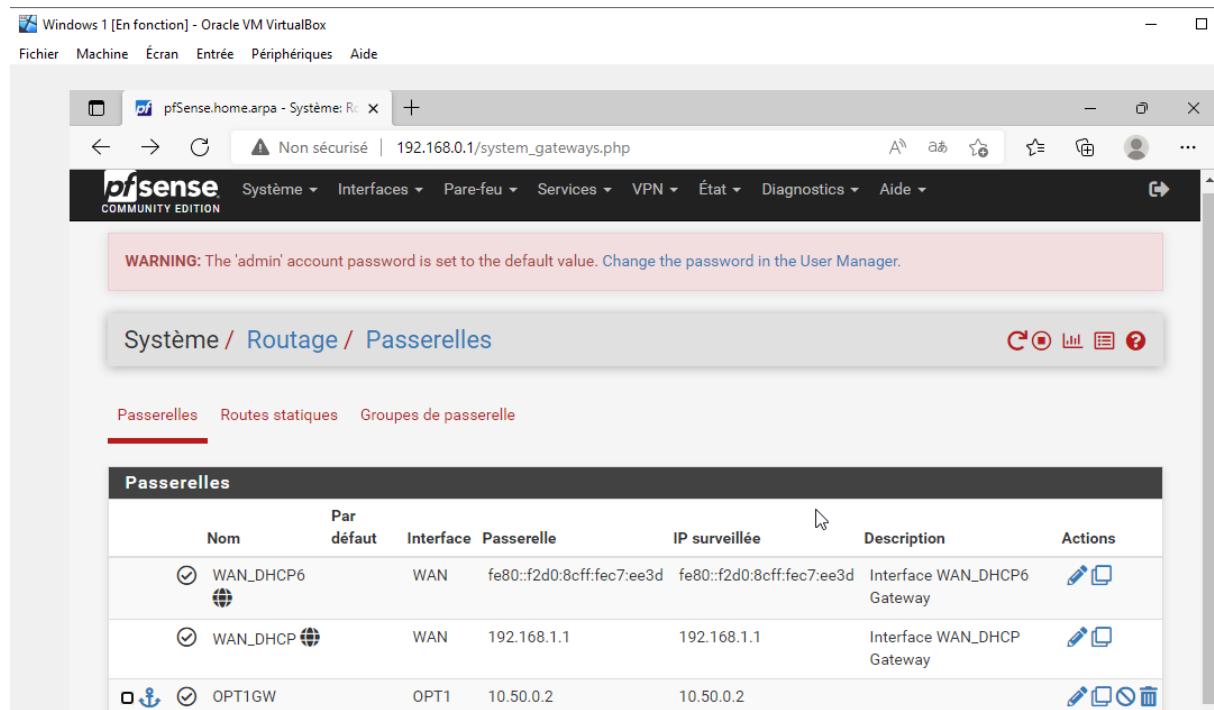
0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults 13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                15) Restore recent configuration
7) Ping host                  16) Restart PHP-FPM

Enter an option: 

```

On fait de même avec le deuxième routeur.

3- Configuration des routeurs PfSense



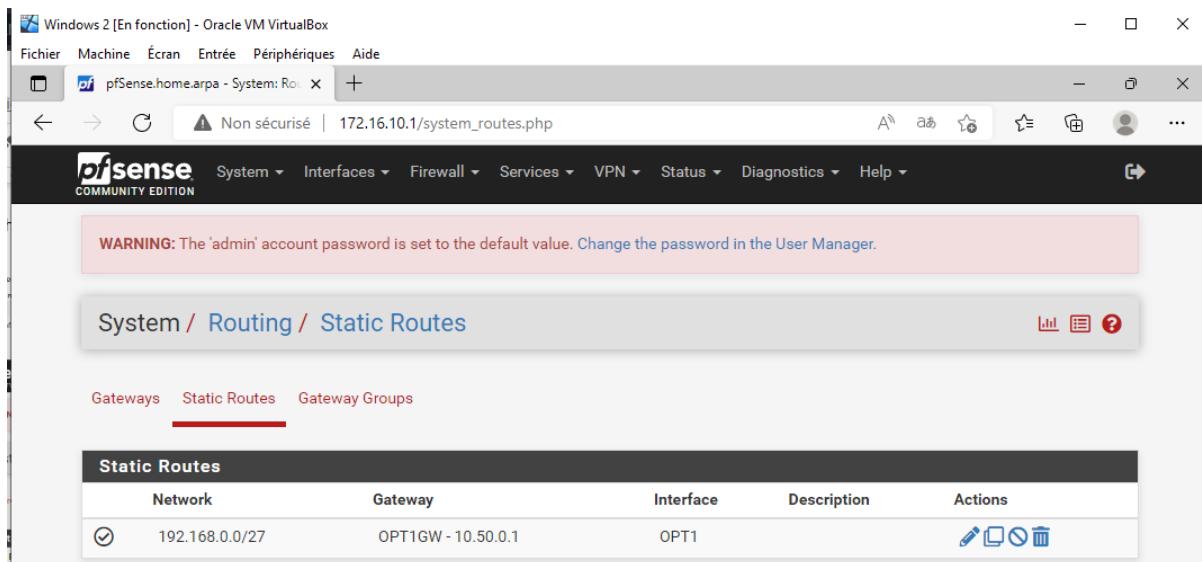
Passerelles						
Nom	Par défaut	Interface	Passerelle	IP surveillée	Description	Actions
WAN_DHCP6	<input checked="" type="checkbox"/>	WAN	fe80::f2d0:8cff:fe7:ee3d	fe80::f2d0:8cff:fe7:ee3d	Interface WAN_DHCP6 Gateway	
WAN_DHCP	<input checked="" type="checkbox"/>	WAN	192.168.1.1	192.168.1.1	Interface WAN_DHCP Gateway	
OPT1GW	<input checked="" type="checkbox"/>	OPT1	10.50.0.2	10.50.0.2		

On configure la passerelle pour accéder au deuxième routeur en la connectant à l'interface de l'OPT1 du premier routeur.

Name	Default	Interface	Gateway	Monitor IP	Description	Actions
WAN_DHCP6		WAN	fe80::f2d0:8cff:fe7:ee3d	fe80::f2d0:8cff:fe7:ee3d	Interface WAN_DHCP6 Gateway	
WAN_DHCP		WAN	192.168.1.1	192.168.1.1	Interface WAN_DHCP Gateway	
OPT1GW		OPT1	10.50.0.1	10.50.0.1		

De même avec le deuxième routeur.

Réseau	Passerelle	Interface	Description	Actions
172.16.10.0/28	OPT1GW - 10.50.0.2	OPT1	route pour aller au deuxième réseau	



Windows 2 [En fonction] - Oracle VM VirtualBox

Fichier Machine Écran Entrée Périphériques Aide

pfSense.home.arpa - System: Root

Non sécurisé | 172.16.10.1/system_routes.php

pfSense COMMUNITY EDITION

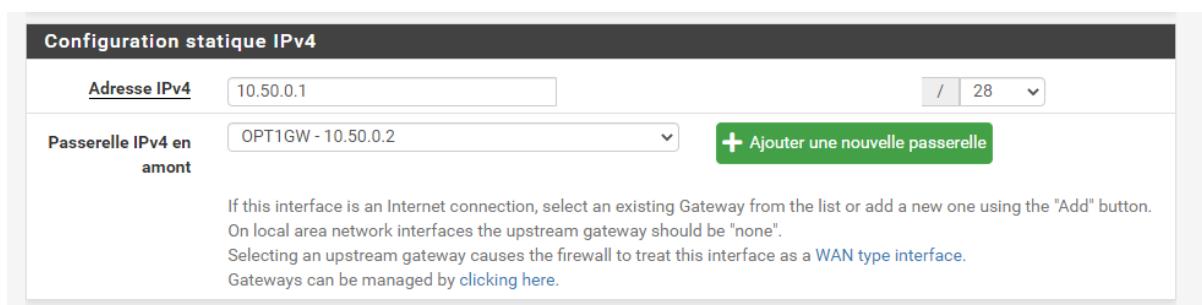
System / Routing / Static Routes

Gateways Static Routes Gateway Groups

Static Routes

Network	Gateway	Interface	Description	Actions
192.168.0.0/27	OPT1GW - 10.50.0.1	OPT1		

On crée ensuite les routes pour accéder à chacun des réseaux (cela va permettre de cibler toutes les stations se trouvant sur le réseau)



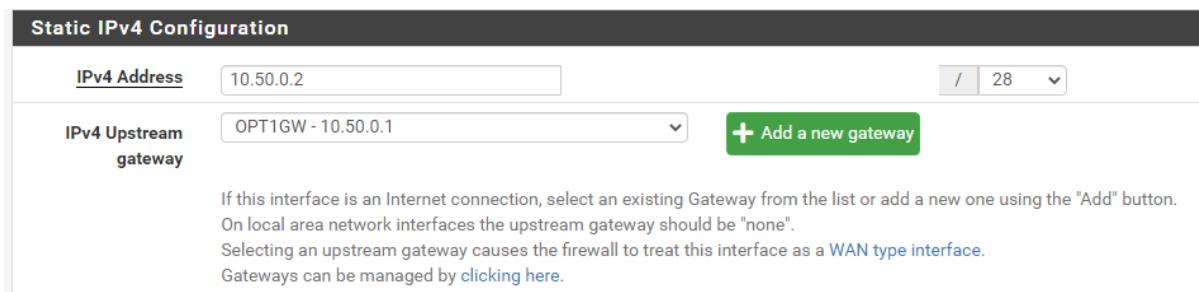
Configuration statique IPv4

Adresse IPv4: 10.50.0.1 / 28

Passerelle IPv4 en amont: OPT1GW - 10.50.0.2

+ Ajouter une nouvelle passerelle

If this interface is an Internet connection, select an existing Gateway from the list or add a new one using the "Add" button. On local area network interfaces the upstream gateway should be "none". Selecting an upstream gateway causes the firewall to treat this interface as a WAN type interface. Gateways can be managed by clicking here.



Static IPv4 Configuration

IPv4 Address: 10.50.0.1 / 28

IPv4 Upstream gateway: OPT1GW - 10.50.0.1

+ Add a new gateway

If this interface is an Internet connection, select an existing Gateway from the list or add a new one using the "Add" button. On local area network interfaces the upstream gateway should be "none". Selecting an upstream gateway causes the firewall to treat this interface as a WAN type interface. Gateways can be managed by clicking here.

On ajoute la passerelle directement sur l'interface (OPT1)

Pare-feu / Règles / LAN

Flottant(e) WAN LAN OPT1

Règles (Faire glisser pour changer l'ordre)

Etat	Protocole	Source	Port	Destination	Port	Passerelle	File d'attente	Ordonnancement	Description	Actions
<input checked="" type="checkbox"/>	2 /1.06 MiB	*	*	*	LAN Address	80	*	*	Règle anti-blocage	
<input checked="" type="checkbox"/>	1 /21.70 MiB	IPv4	*	LAN net	*	*	*	*	Default allow LAN to any rule	
<input type="checkbox"/>	0 /0 B	IPv6	*	LAN net	*	*	*	*	Default allow LAN IPv6 to any rule	

Par défaut les règles pour le LAN sont déjà créées pour pouvoir ping du routeur jusqu'au station et l'inverse.

Cela paraît logique.

Pare-feu / Règles / OPT1

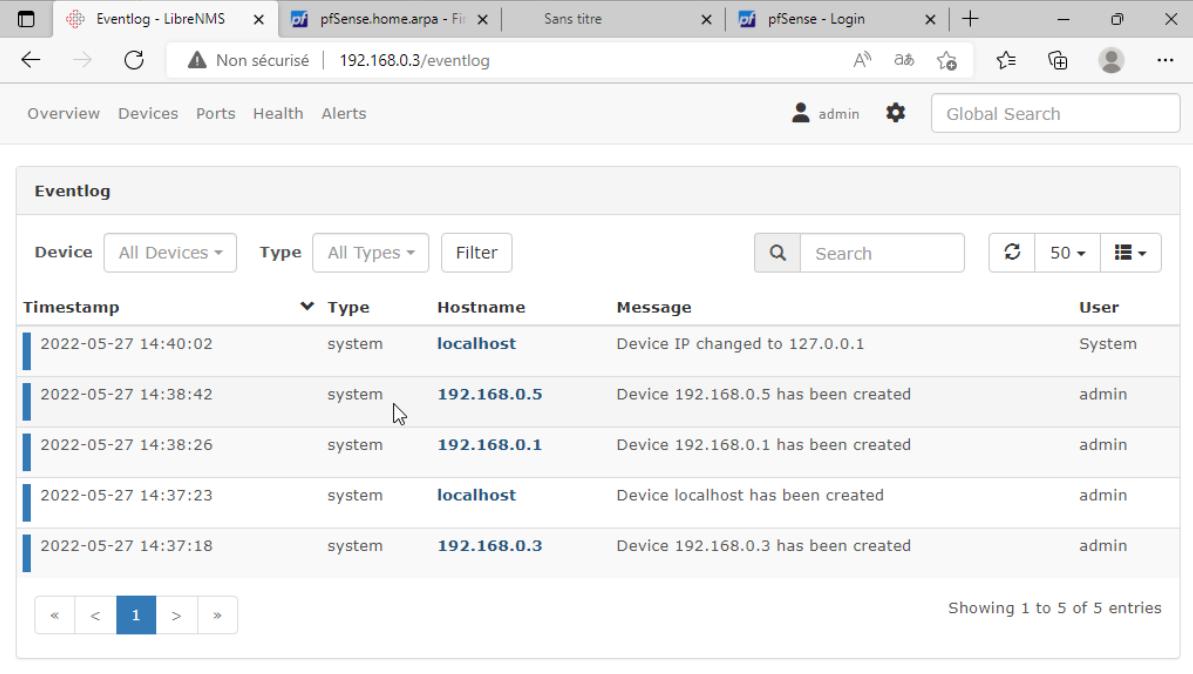
Flottant(e) WAN LAN OPT1

Règles (Faire glisser pour changer l'ordre)

Etat	Protocole	Source	Port	Destination	Port	Passerelle	File d'attente	Ordonnancement	Description	Actions
<input type="checkbox"/>	1 /36 KiB	IPv4 ICMP	*	*	*	*	*	*	aucun	

Enfin on ajoute la règle qui permet de faire passer le ping par la passerelle entre les deux routeurs.

4- Configuration de Librenms

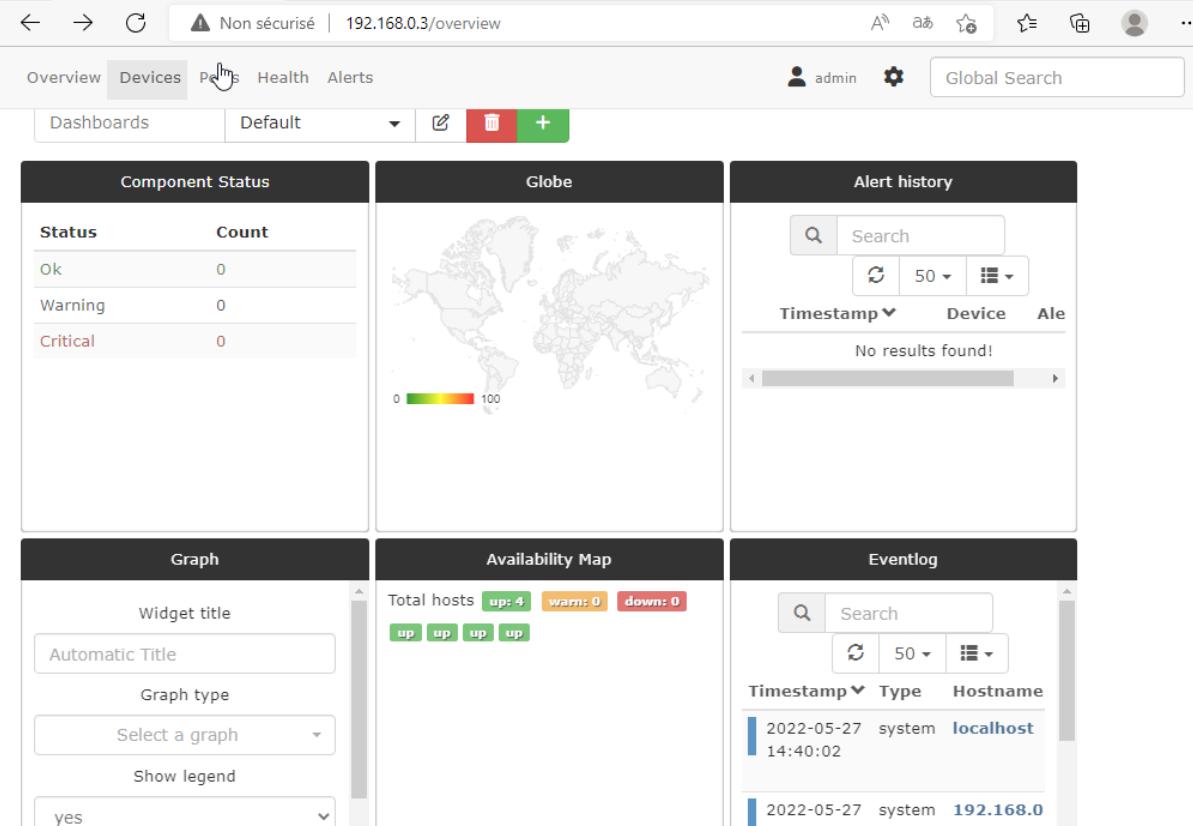


The screenshot shows the LibreNMS Eventlog interface. The top navigation bar includes tabs for Eventlog - LibreNMS, pfSense.home.arp - Fire, Sans titre, and pfSense - Login. The main content area is titled "Eventlog" and displays a table of log entries. The columns are: Timestamp, Type, Hostname, Message, and User. The logs are as follows:

Timestamp	Type	Hostname	Message	User
2022-05-27 14:40:02	system	localhost	Device IP changed to 127.0.0.1	System
2022-05-27 14:38:42	system	192.168.0.5	Device 192.168.0.5 has been created	admin
2022-05-27 14:38:26	system	192.168.0.1	Device 192.168.0.1 has been created	admin
2022-05-27 14:37:23	system	localhost	Device localhost has been created	admin
2022-05-27 14:37:18	system	192.168.0.3	Device 192.168.0.3 has been created	admin

At the bottom, there are navigation buttons («, <, 1, >, ») and a message: "Showing 1 to 5 of 5 entries".

On ajoute les appareils qu'on connaît donc le routeur, le pc et le serveur linux qui contient le service libreNMS.



The screenshot shows the LibreNMS Overview interface. The top navigation bar includes tabs for Overview, Devices, Pe, Health, and Alerts. The main content area is divided into several gadgets:

- Component Status:** Shows a table with Status (Ok, Warning, Critical) and Count (0, 0, 0).
- Globe:** A world map showing network topology with a color scale from 0 to 100.
- Alert history:** An empty search interface with a message: "No results found!"
- Graph:** A configuration panel for a graph, including fields for Widget title, Automatic Title, Graph type, and Show legend (set to yes).
- Availability Map:** Shows 4 total hosts, all up (green).
- Eventlog:** A table showing the same log entries as the previous screenshot, with the first entry: "2022-05-27 system localhost 14:40:02" and the second entry: "2022-05-27 system 192.168.0 14:38:42".

Puis, on ajoute les gadgets utiles comme les alertes si jamais il y a un problème sur un appareil, des logs pour chaque appareil et des graphes.

Conclusion :

On a bien rempli la mission qui est de mettre en place deux sous-réseaux avec deux routeurs et qui sont séparés par un pare-feu. Le Lan dont l'adresse de réseau est 192.168.0.1 contient un serveur Librenms qui permet de surveiller le réseau avec un système d'alerte. On peut vérifier les logs pour voir si un problème se pose sur un appareil du sous-réseau.