Amine Boukherouba Lapierre BTS SIO SISR

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:

Prérequis :

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

Explication :

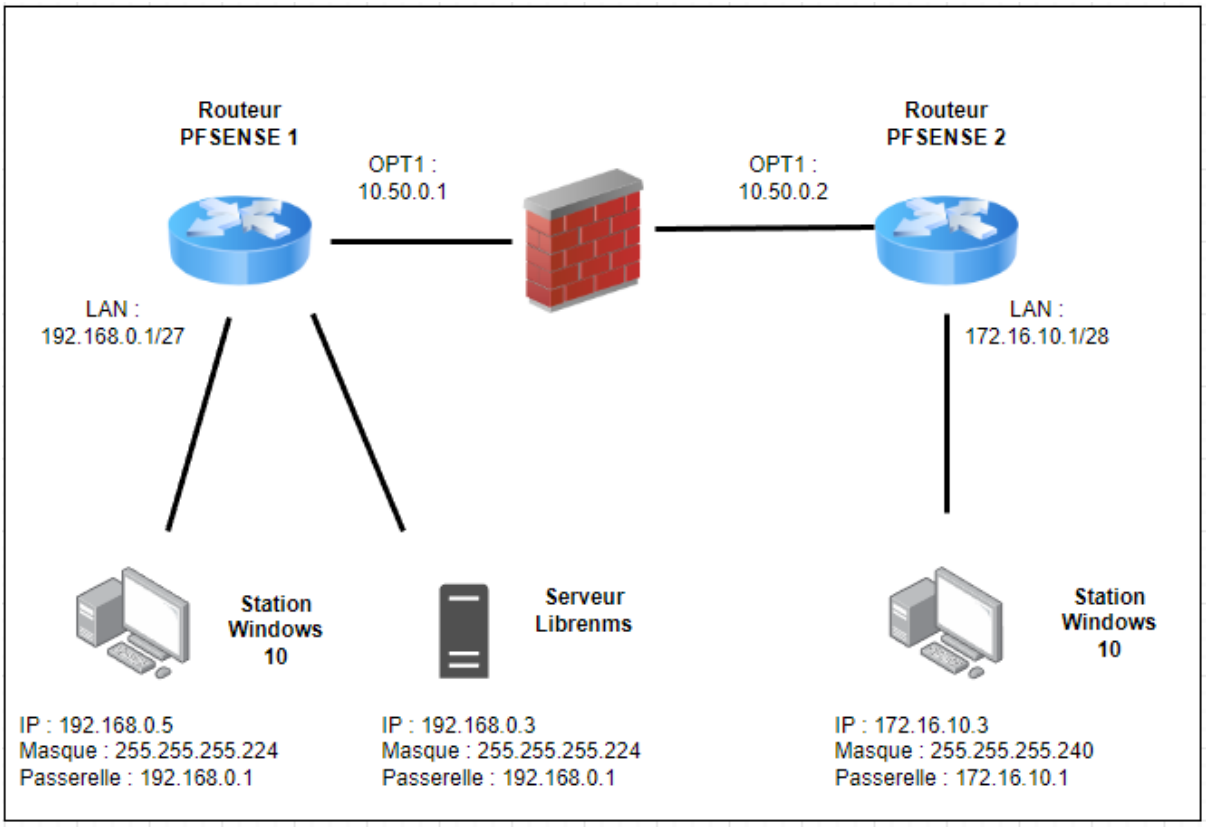
Tout d'abord, nous allons installer LibreNMS sur un serveur Ubuntu en version 22.04.

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: ~
root@serverlinux:~# root@serverlinux:~# apt install acl curl apache2 composer fping git graphviz imagemagick libapache2-mod-fcgid mariadb-client mariadb-s
erver 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 rrdt
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
/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
v composer-stable.phar /usr/bin/composer
chmod +x /usr/bin/composer
```

On fait ça aussi pour installer composer.

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

On fait cela pour le système.

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

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

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

```
root@serverlinux: /opt
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 OK, 1 row affected (0.001 sec)

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

;
FLUSH PRMariaDB [(none)]> GRANT ALL PRIVILEGES ON librenms.* TO 'librenms'@'localhost';
IVILEGES;
exit
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 privilèges.

```
root@serverlinux: /opt
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
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/sit
<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
od proxy_fcgi setenvif rewrite
a2ensite librenms.conf
systemctl restart apache2
systemctl restart php7.4-fpmSite 000-default disabled.
To activate the new configuration, you need to run:
  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
/opt/librenms/misc/lnms-completion.bash /etc/bash_completion.d/
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# cp /opt/librenms/snmpd.conf.example /etc/snmp/snmpd.conf
```

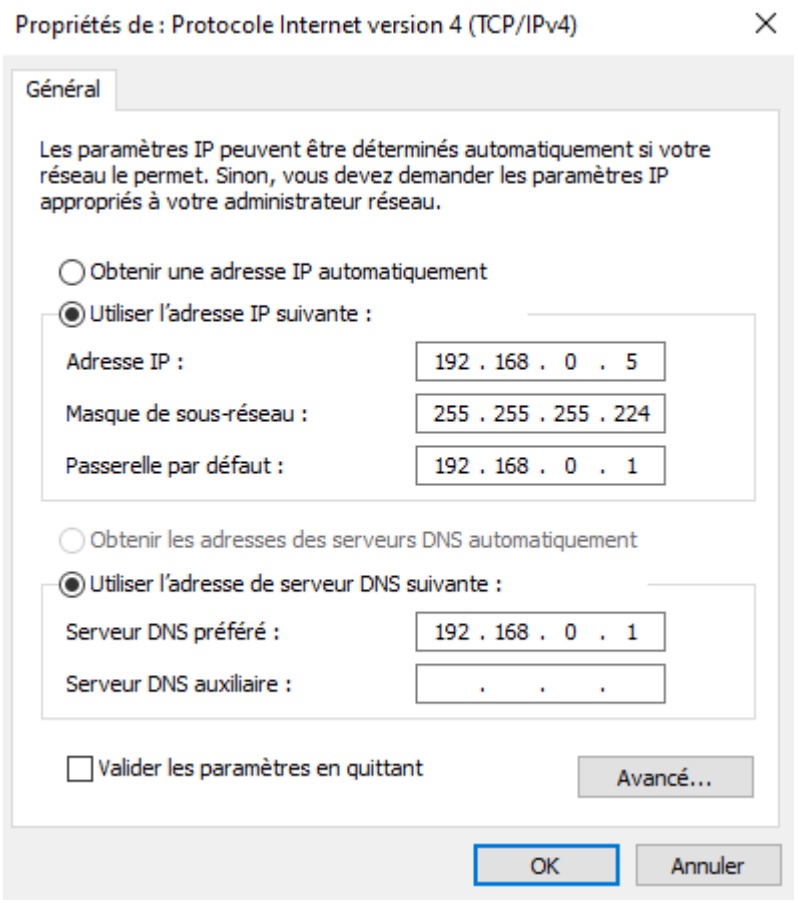
```
root@serverlinux: /opt
  GNU nano 6.2                                                  /etc/snmp/snmpd.conf *
# Change RANDOMSTRINGGOESHERE to your preferred SNMP community string
com2sec readonly  default         librenms

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.

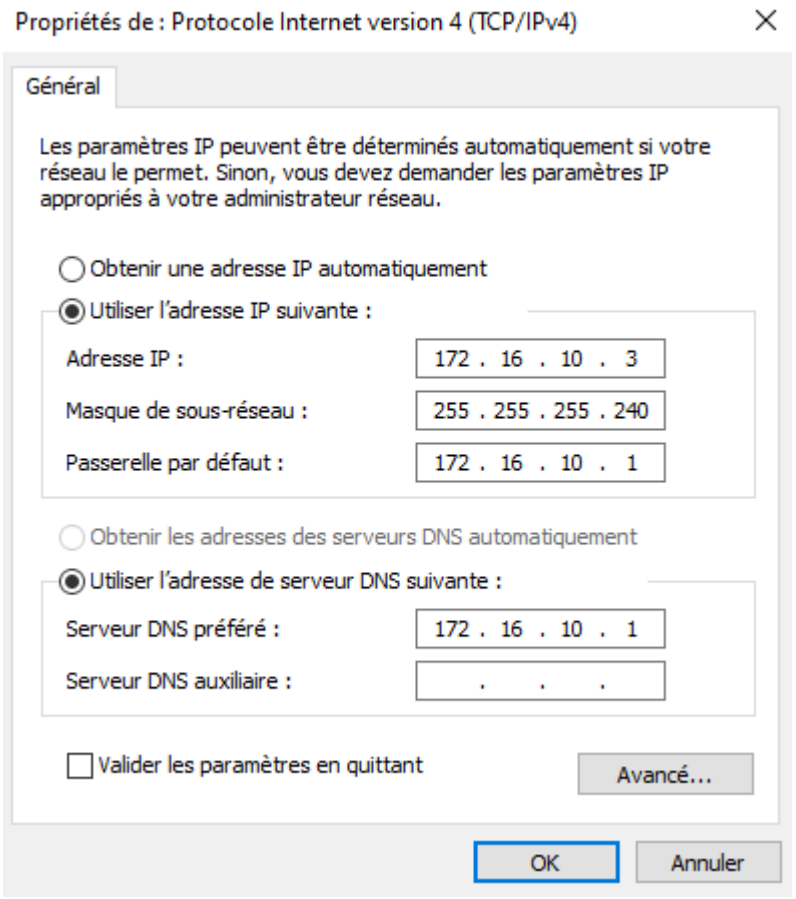
```
root@serverlinux: /opt
root@serverlinux:/opt# curl -o /usr/bin/distro https://raw.githubusercontent.com/librenms/librenms-agent/master/snmp/distro
x /usr/bin/distro
systemctl enable snmpd
systemctl restart snmpd
```

On active et redémarre snmpd une fois snmpd installé.

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

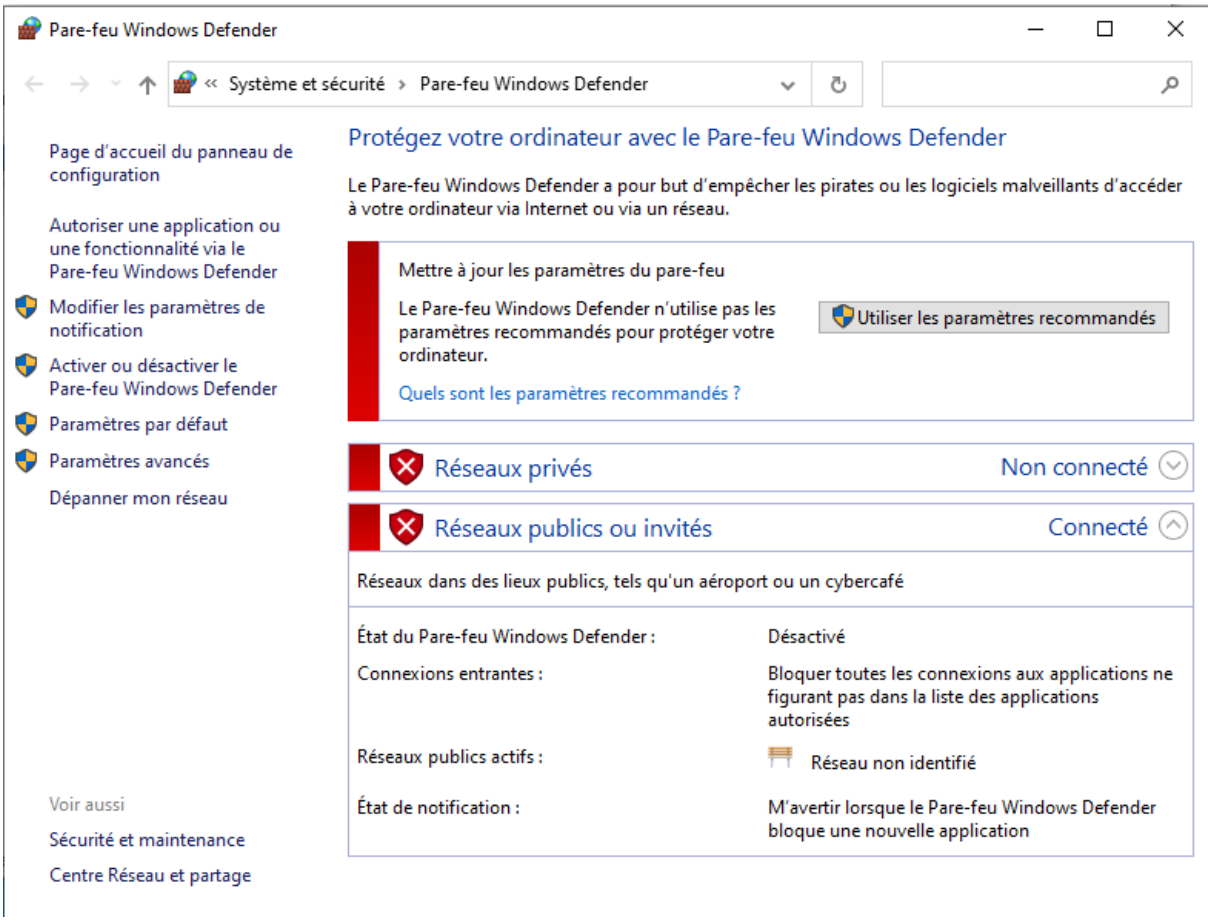


On met une ip qui est dans la plage de notre réseau du routeur 192.168.0.1 et on met aussi cet ip dans la passerelle pour ensuite se relier au deuxième routeur. (PC 1)



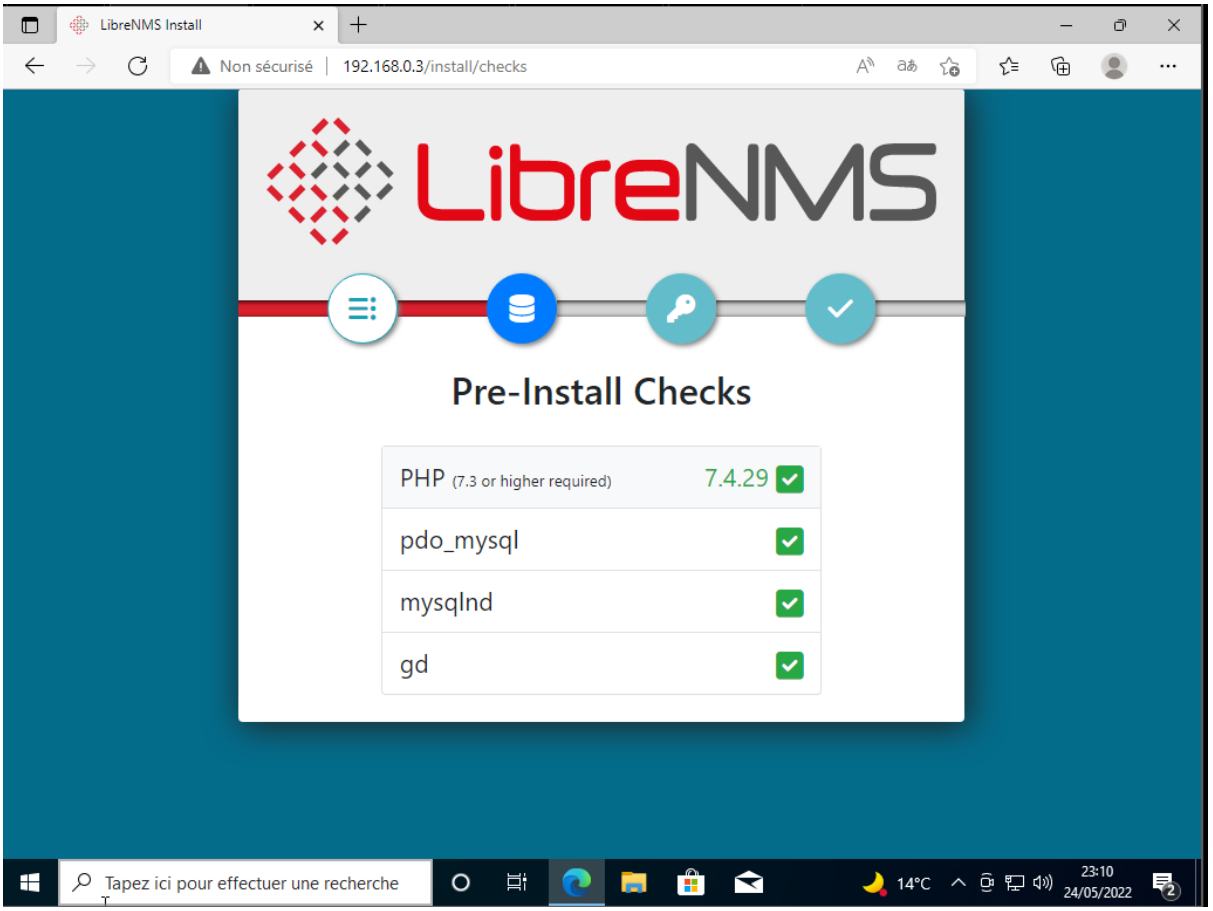
Idem pour ce pc qui sera dans un autre réseau lan.

Amine Boukherouba Lapierre BTS SIO SISR

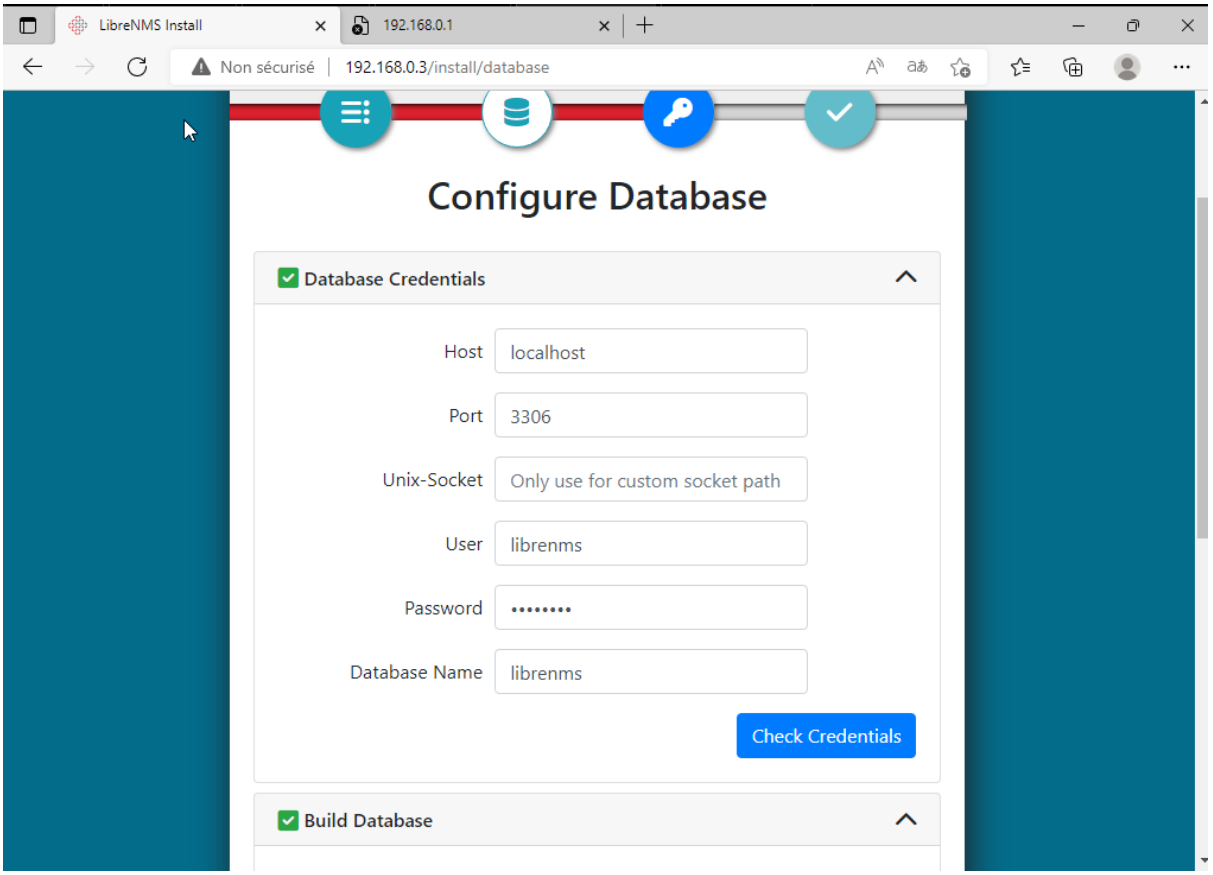


On désactive le pare-feu windows pour les deux pc pour qu'il ne bloque pas les paquets venant d'autrui.

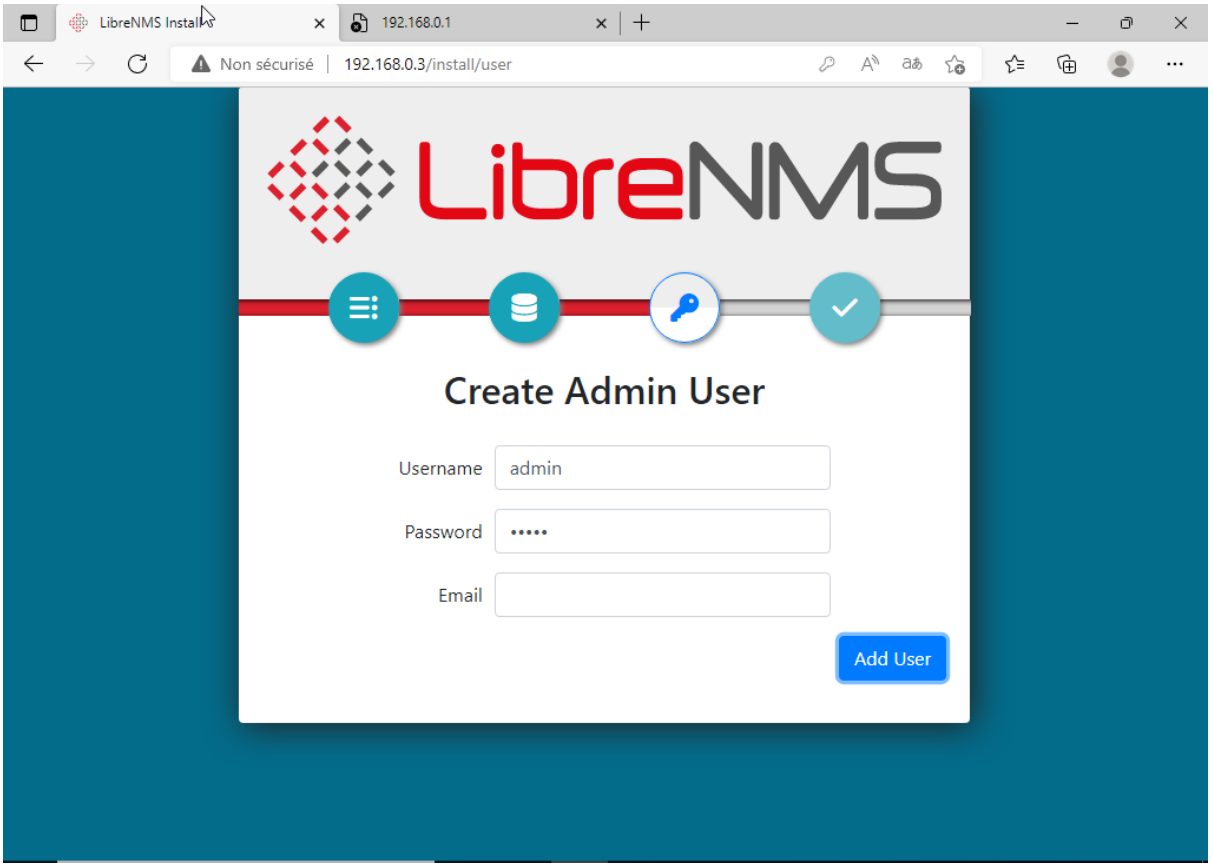
Amine Boukherouba Lapierre BTS SIO SISR



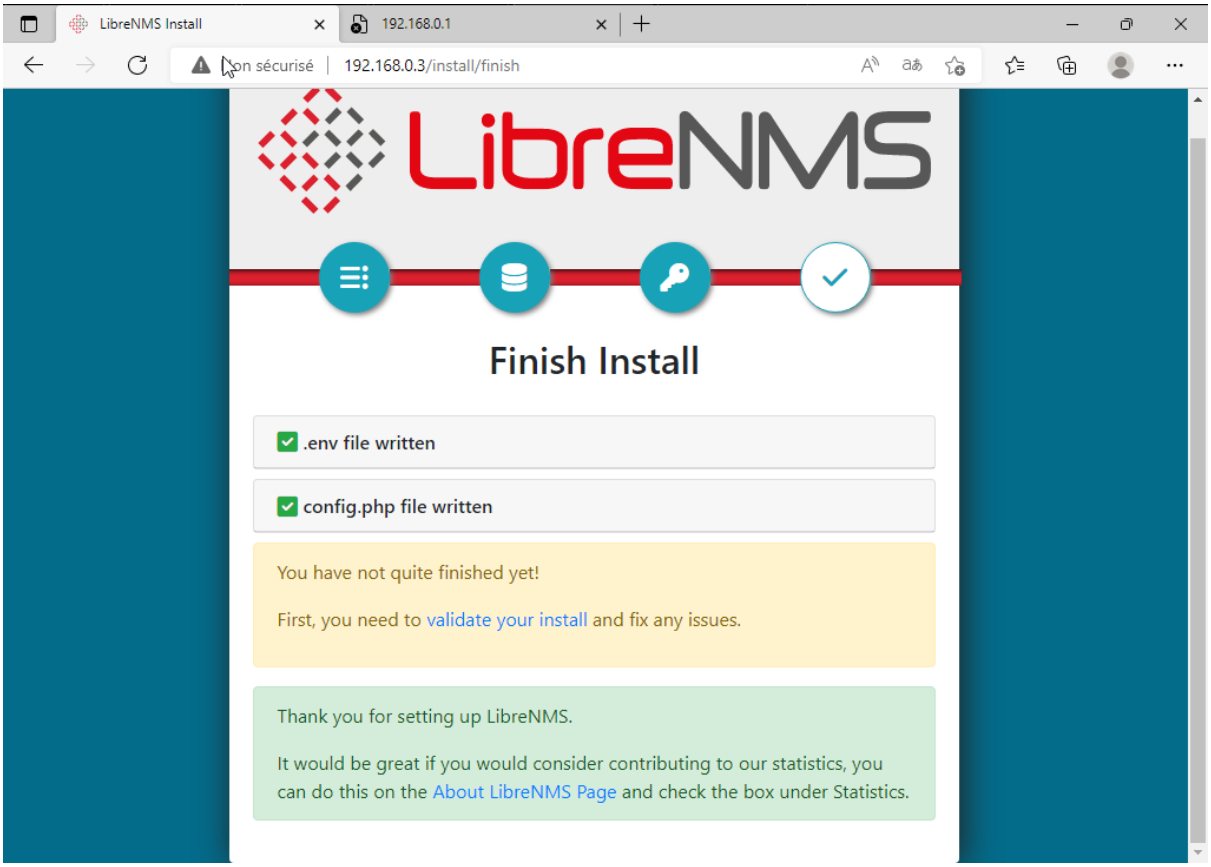
On fait une vérification des pré installations des différents packages.



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

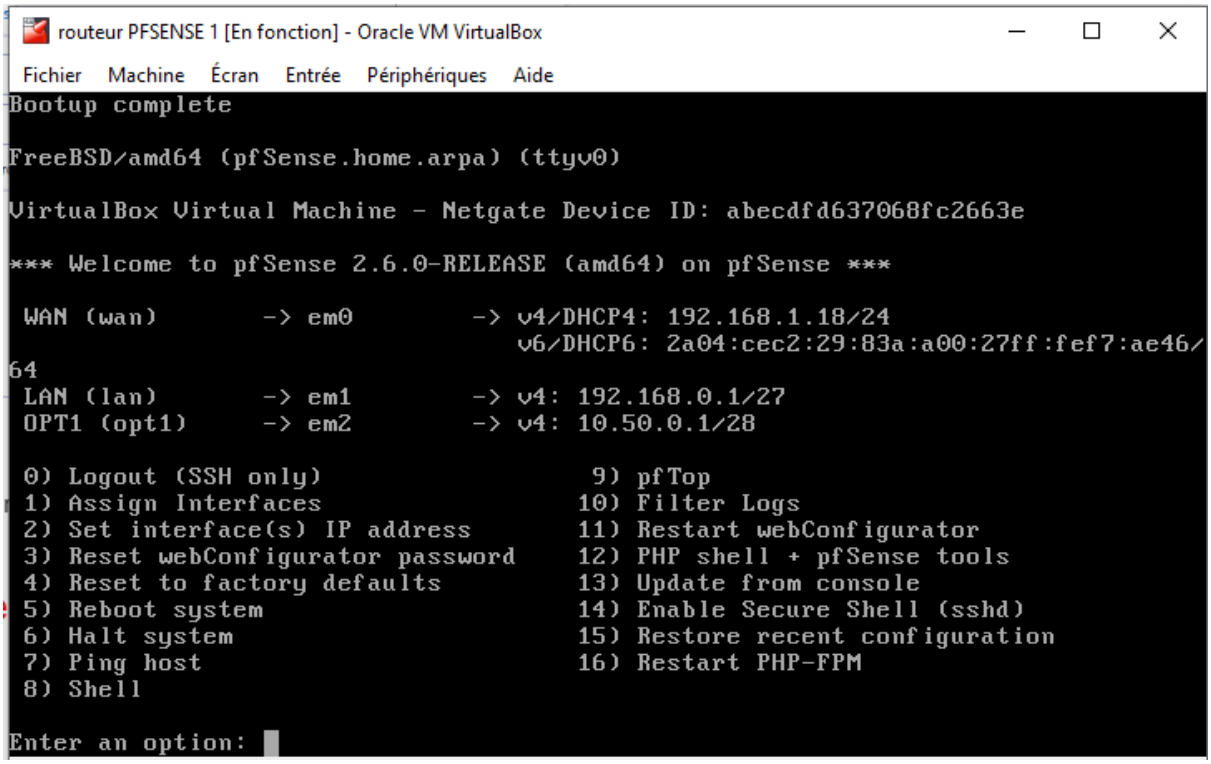


On crée le compte admin pour se connecter dessus.

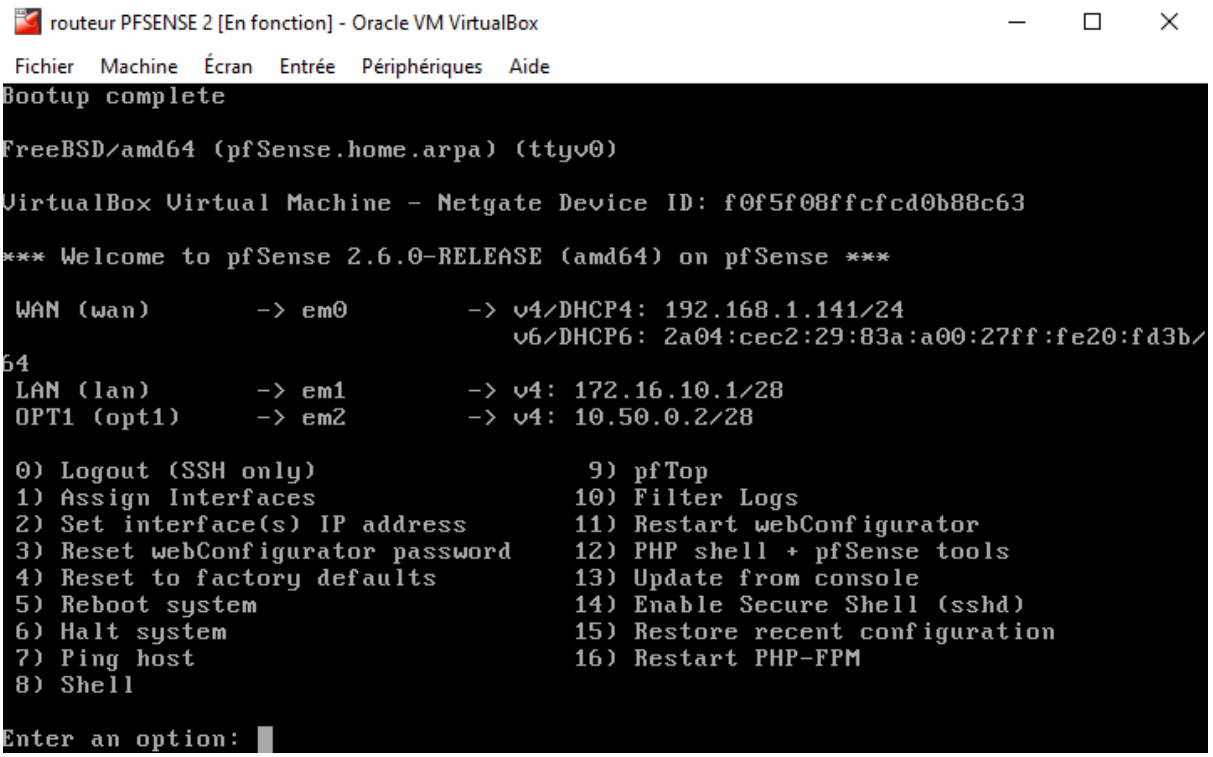


Et on vient de finir l'installation.

2- Installation de deux routeurs Pfsense

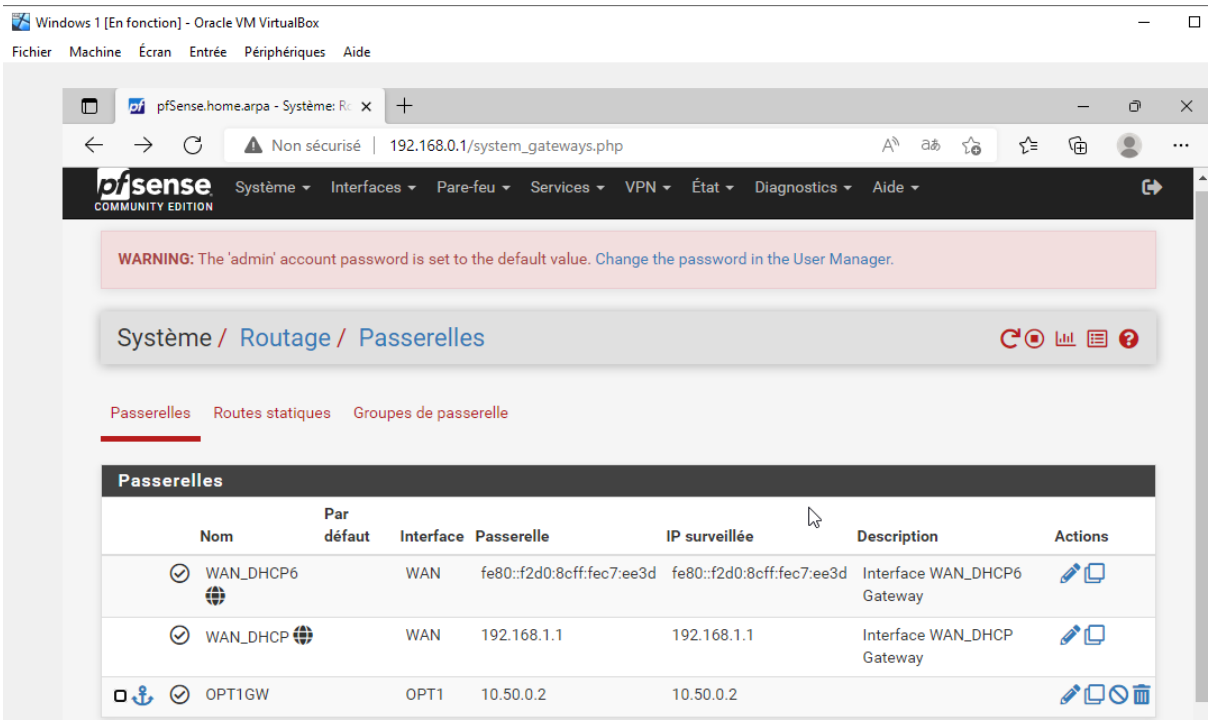


Activer trois interfaces réseau en les ajoutant avec 2 LAN (LAN et OPT1).
OPT1 permettra de faire la passerelle entre le premier routeur et le deuxième.

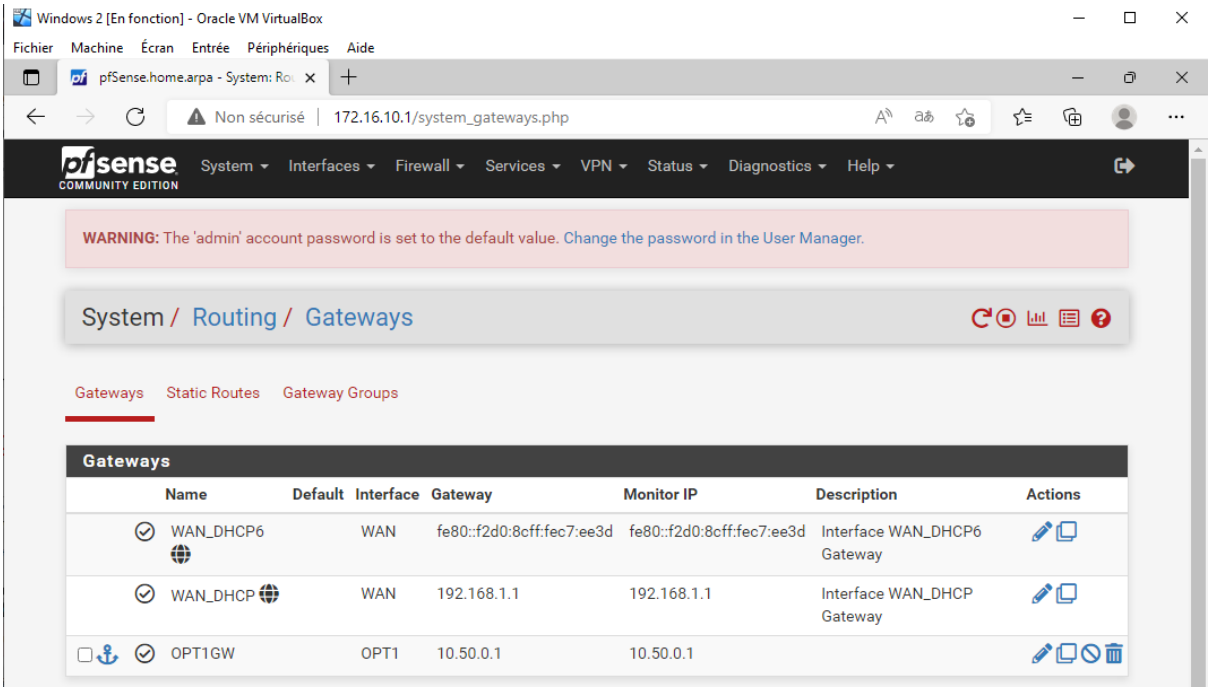


On fait de même avec le deuxième routeur.

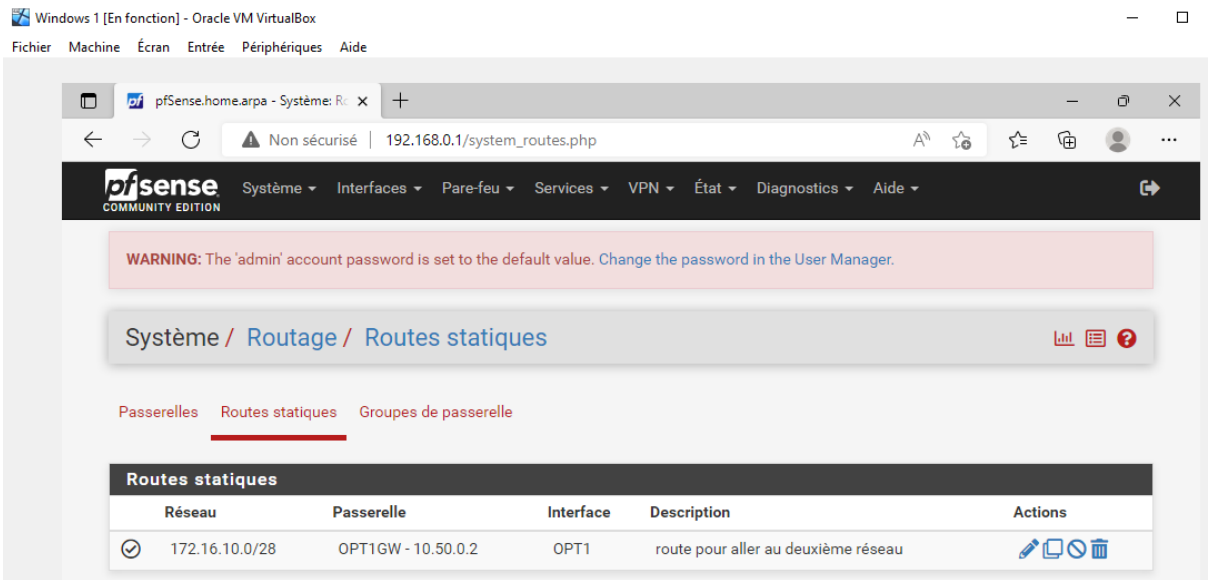
3- Configuration des routeurs Pfsense



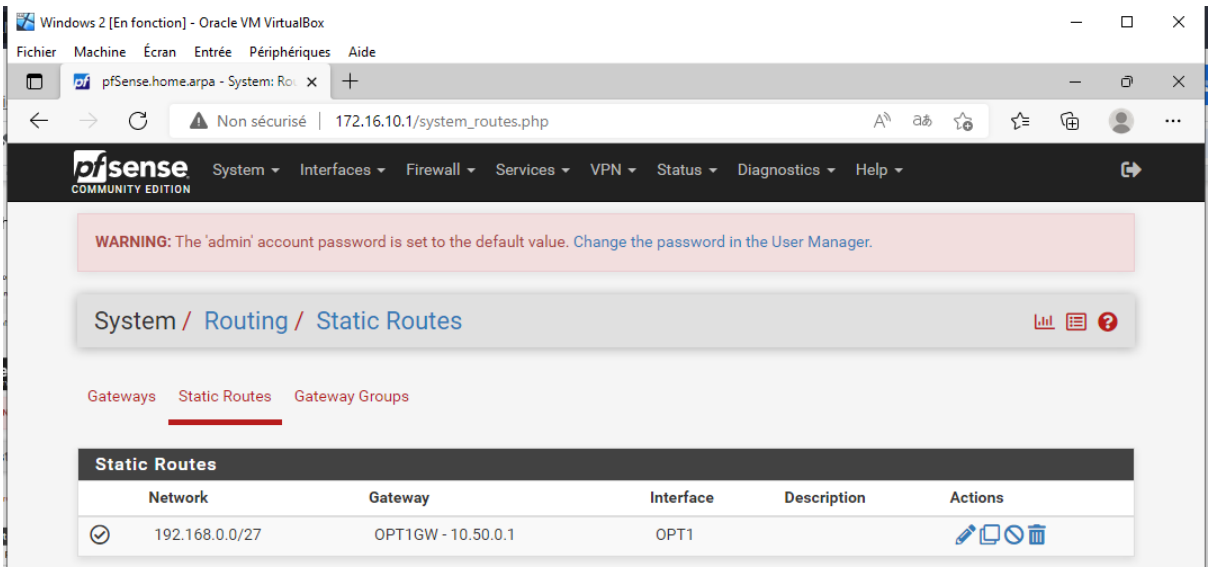
On configure la passerelle pour accéder au deuxième routeur en la connectant à l'interface de l'OPT1 du premier routeur.



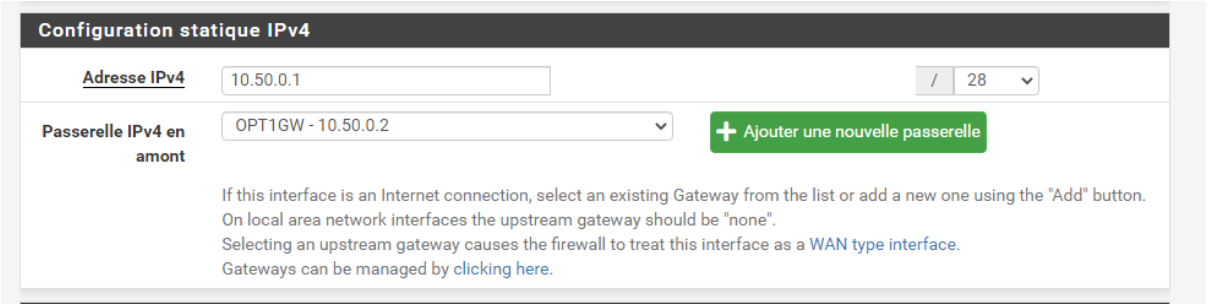
De même avec le deuxième routeur.

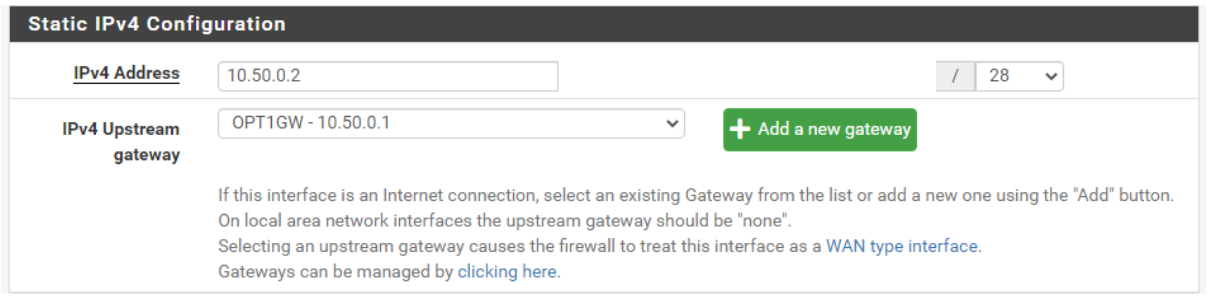


Amine Boukherouba Lapierre BTS SIO SISR

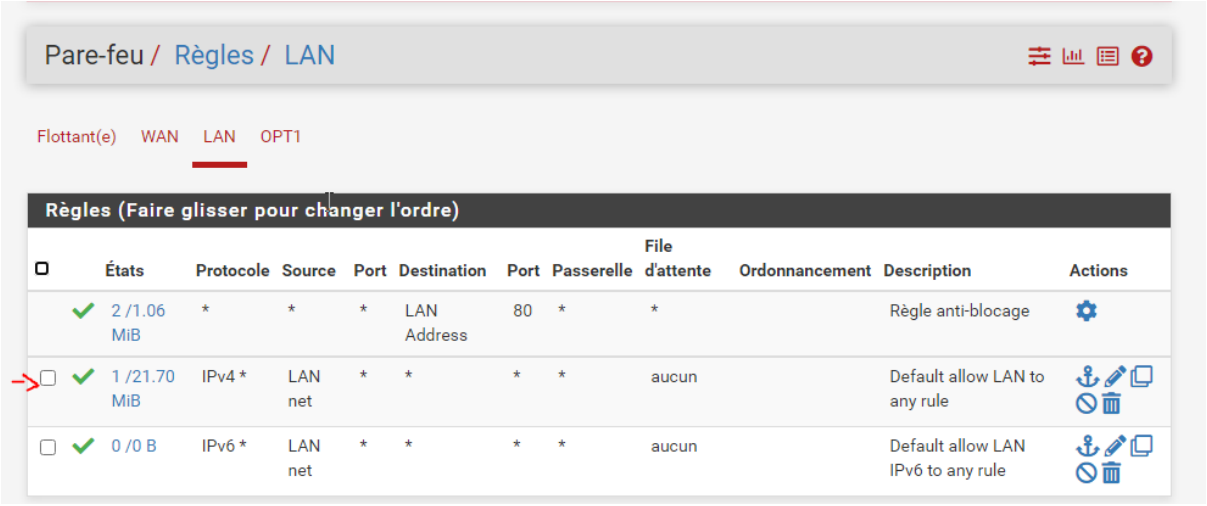


On crée ensuite les routes pour accéder à chacun des réseaux (cela va permettre de cibler toutes les stations se trouvant sur le réseau)





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

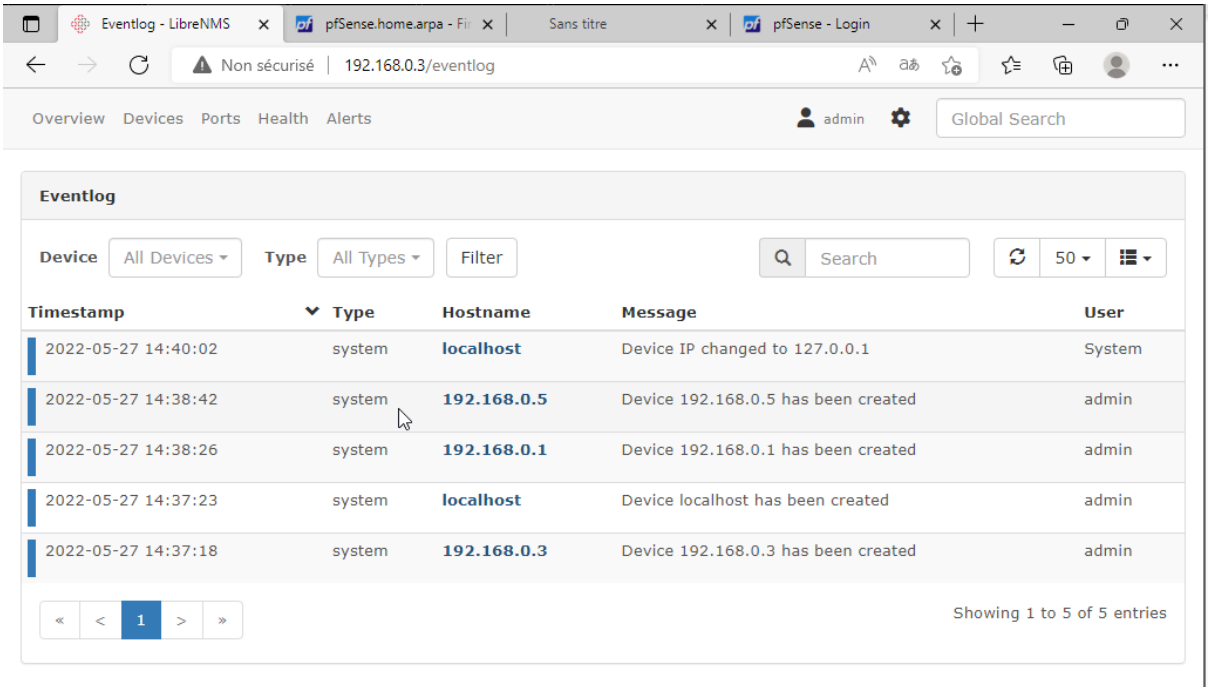


Par défaut les règles pour le LAN sont déjà créées pour pouvoir ping du routeur jusqu'au station et l'inverse.
Cela parait logique.

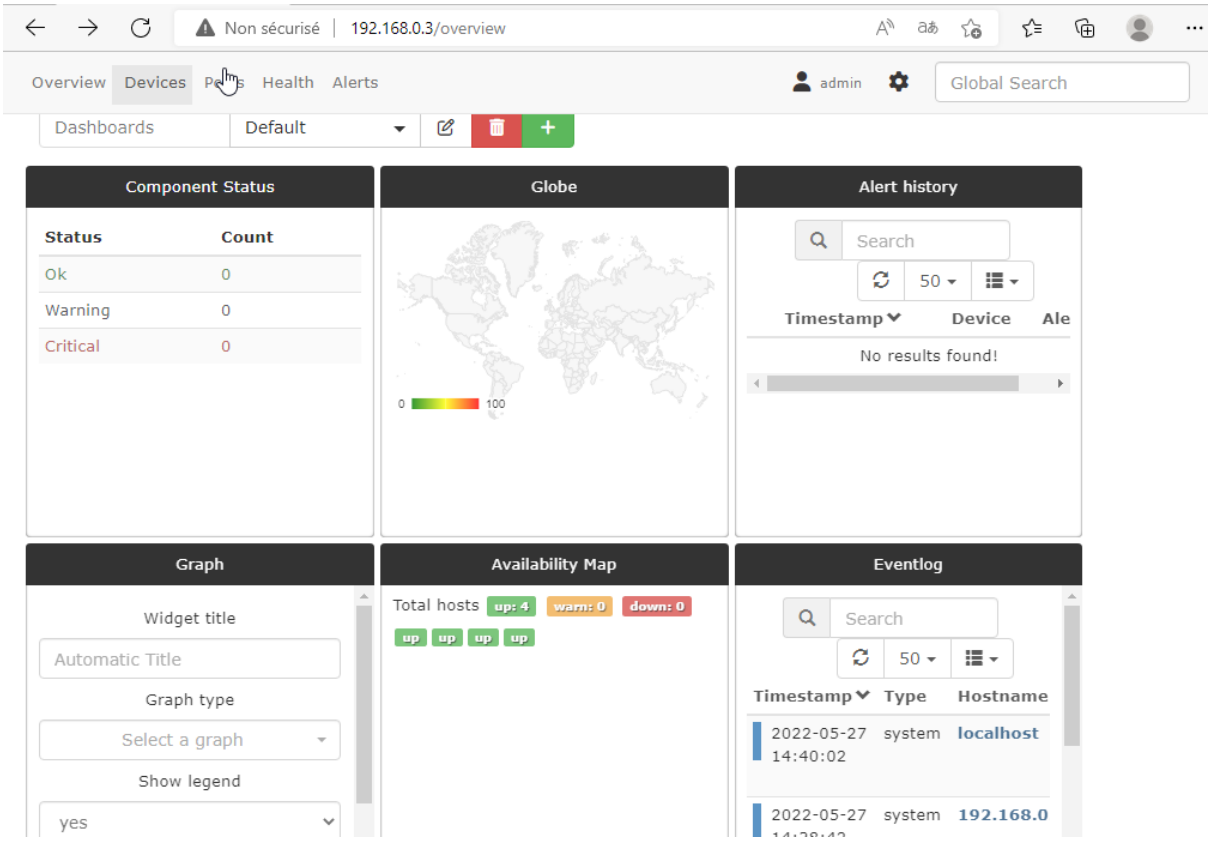
Enfin on ajoute la règle qui permet de faire passer le ping par la passerelle entre les deux routeurs.

4- Configuration de Librenms

Amine Boukherouba Lapierre BTS SIO SISR



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



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.