DE LA RÉUNION Institut Universitaire de Technologie	**Projet de la SAE2.01**	**Objectif** Mise en place d'une infrastructure de réseau d'entreprise	**ALAMELOU BOUGHLEM CHEREAU**

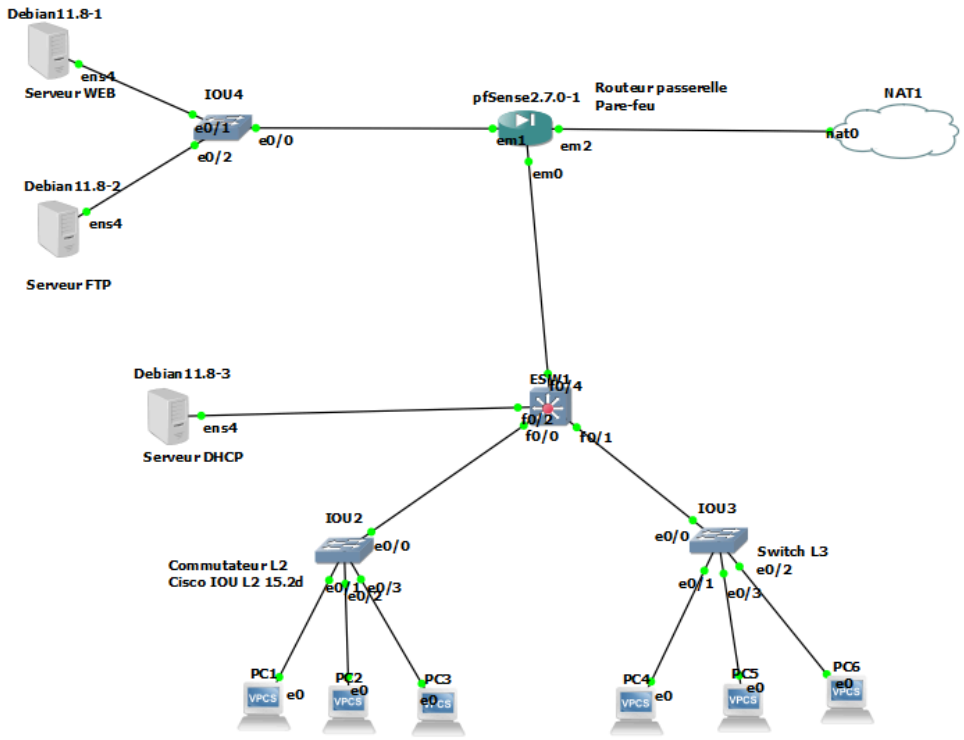
SAE 2.01 : Construire un réseau informatique pour une petite structure

Voici le rapport final comprenant les procédures et résultats pour la mise en place d'une infrastructure de réseau d'entreprise :

Une fois toutes les licences obtenues,

```
[license]
gns3vm = 73635fd3b0a13ad0;
```

On commence par créer la topologie de notre futur réseau :



Puis on décide du plan d'adressage de cette structure :

VLAN	NOM	Réseaux	PASSERELLE	Plage du service DHCP
10	DIRECTION	192.168.10.0	192.168.10.2	192.168.10.99 - 192.168.10.200
20	GESTION	192.168.20.0	192.168.20.2	192.168.20.99 - 192.168.20.200
30	COMTECH	192.168.30.0	192.168.30.2	192.168.30.99 - 192.168.30.200

Il faut ensuite créer un réseau segmenté comprenant la mise en place et la configuration des VLAN (Dans les deux commutateurs d'accès de couche 2) :

```
IOU2#show ip int brief
Interface              IP-Address      OK? Method Status                Protocol
Ethernet0/0            unassigned      YES unset  up                    up
Ethernet0/1            unassigned      YES unset  up                    up
Ethernet0/2            unassigned      YES unset  up                    up
Ethernet0/3            unassigned      YES unset  up                    up
Ethernet1/0            unassigned      YES unset  up                    up
Ethernet1/1            unassigned      YES unset  up                    up
Ethernet1/2            unassigned      YES unset  up                    up
Ethernet1/3            unassigned      YES unset  up                    up
Ethernet2/0            unassigned      YES unset  up                    up
Ethernet2/1            unassigned      YES unset  up                    up
Ethernet2/2            unassigned      YES unset  up                    up
Ethernet2/3            unassigned      YES unset  up                    up
Ethernet3/0            unassigned      YES unset  up                    up
Ethernet3/1            unassigned      YES unset  up                    up
Ethernet3/2            unassigned      YES unset  up                    up
Ethernet3/3            unassigned      YES unset  up                    up
Vlan1                  unassigned      YES unset  administratively down down
```

Voici donc l'affectation des différents ports aux différents VLAN et donc réseaux :

Port (pour les Switchs IOU2 et IOU3)	Affectation	Réseaux
Ethernet 0/0-3	VLAN 10	192.168.10.0
Ethernet 1/0-3	VLAN 20	192.168.20.0
Ethernet 2/0-3	VLAN 30	192.168.30.0

On obtient donc ce script de configuration de switch :

```
enable
configure terminal

vlan 10
name Direction
exit

vlan 20
name Gestion
exit

vlan 30
name COMTECH
exit
```

```
interface Gi0/0
no switchport
ip address 192.168.69.2 255.255.255.0

interface Gi0/1
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 10,20,30
exit

interface Gi0/2
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 10,20,30
exit

interface Gi0/3
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 10,20,30
exit

interface vlan 10
ip address 192.168.10.1 255.255.255.0
exit

interface vlan 20
ip address 192.168.20.1 255.255.255.0
exit

interface vlan 30
ip address 192.168.30.1 255.255.255.0
exit

ip routing

end
wr
```

Idem pour l'autre switch de niveau 2 :

```
enable
configure terminal

vlan 10
name DIRECTION
exit

vlan 20
name GESTION
exit

vlan 30
name COMTECH
exit

interface range eth0/0-3
switchport mode access
switchport access vlan 10
exit

interface range eth1/0-3
switchport mode access
```

```
switchport access vlan 20
exit

interface range eth2/0-3
switchport mode access
switchport access vlan 30
exit

interface eth3/3
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 10,20,30
exit
```

On peut ensuite mettre en place et configuration du routage inter-VLAN (nécessitant donc un commutateur de distribution de couche 3 expliquant le choix du Switch L3 :

```
enable
configure terminal

vlan 10
name Direction
exit

vlan 20
name Gestion
exit

vlan 30
name COMTECH
exit

interface Gi0/0
no switchport
ip address 192.168.69.2 255.255.255.0

interface Gi0/1
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 10,20,30
exit

interface Gi0/2
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 10,20,30
exit

interface Gi0/3
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 10,20,30
exit

interface vlan 10
ip address 192.168.10.1 255.255.255.0
exit

interface vlan 20
ip address 192.168.20.1 255.255.255.0
exit
```

```
interface vlan 30
ip address 192.168.30.1 255.255.255.0
exit
ip routing

end
wr
```

On peut alors installer et configurer des services **DHCP**, **FTP**, **HTTP** et **SSH** sur des serveurs (virtualisés) de l'infrastructure réseau (ce qui explique notre choix des Debian 11.8) :

Voici comment on configure et valide notre service DHCP opérationnel :

On commence par installer le paquet "isc-dhcp-server" puis modifier le fichier /etc/network/interfaces afin de permettre l'attribution des adresses statiques pour chaque VLAN sur le serveur DHCP :

```
# VLAN 10
auto ens4.10
iface ens4.10 inet static
    address 192.168.10.1
    netmask 255.255.255.0
    vlan-raw-device ens4

# VLAN 20
auto ens4.20
iface ens4.20 inet static
    address 192.168.20.1
    netmask 255.255.255.0
    vlan-raw-device ens4

# VLAN 30
auto ens4.30
iface ens4.30 inet static
    address 192.168.30.1
    netmask 255.255.255.0
    vlan-raw-device ens4
```

Puis on modifie le fichier /etc/dhcp/dhcpd.conf pour créer le pool DHCP pour chaque VLAN :

```
  GNU nano 5.4                  /etc/dhcp/dhcpd.conf *
# VLAN 10
subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.100 192.168.10.200;
    option routers 192.168.10.2;
    option subnet-mask 255.255.255.0;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}

# VLAN 20
subnet 192.168.20.0 netmask 255.255.255.0 {
    range 192.168.20.100 192.168.20.200;
    option routers 192.168.20.2;
    option subnet-mask 255.255.255.0;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}

# VLAN 30
subnet 192.168.30.0 netmask 255.255.255.0 {
    range 192.168.30.100 192.168.30.200;
    option routers 192.168.30.2;
```

Puis on configure le fichier /etc/default/isc-dhcp-server pour activer notre service DHCP sur les bonnes interfaces :

```
  GNU nano 5.4              /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="ens4 ens4.10 ens4.20 ens4.30"
INTERFACESv6=""
```

Et on peut vérifier que notre service DHCP fonctionne correctement :

```
debian@debian:~$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - LSB: DHCP server
     Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
     Active: active (running) since Fri 2024-06-14 07:05:05 UTC; 1min 13s ago
       Docs: man:systemd-sysv-generator(8)
```

Voici donc les sous-interfaces prises en compte par le serveur DHCP :

```
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 0c:b9:55:a5:00:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet6 fe80::eb9:55ff:fea5:0/64 scope link
       valid_lft forever preferred_lft forever
3: ens4.10@ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 0c:b9:55:a5:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.1/24 brd 192.168.10.255 scope global ens4.10
       valid_lft forever preferred_lft forever
    inet6 fe80::eb9:55ff:fea5:0/64 scope link
       valid_lft forever preferred_lft forever
4: ens4.20@ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 0c:b9:55:a5:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.1/24 brd 192.168.20.255 scope global ens4.20
       valid_lft forever preferred_lft forever
    inet6 fe80::eb9:55ff:fea5:0/64 scope link
       valid_lft forever preferred_lft forever
5: ens4.30@ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 0c:b9:55:a5:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.30.1/24 brd 192.168.30.255 scope global ens4.30
       valid_lft forever preferred_lft forever
    inet6 fe80::eb9:55ff:fea5:0/64 scope link
       valid_lft forever preferred_lft forever
debian@debian:~$
```

Et plus rien ne devrait empêcher notre serveur DHCP d'offrir des adresses IP aux PCs (On effectuer un test sur le VLAN 10) :

Une requête sur le PC1 suffit pour que le serveur DHCP lui offre l'adresse IP 192.168.10.100/24 :

```
PC1> ip dhcp
DORA IP 192.168.10.100/24 GW 192.168.10.2

PC1> show ip

NAME        : PC1[1]
IP/MASK     : 192.168.10.100/24
GATEWAY     : 192.168.10.2
DNS         : 8.8.8.8  8.8.4.4
DHCP SERVER : 192.168.10.1
DHCP LEASE  : 447, 451/225/394
DOMAIN NAME : example.org
MAC         : 00:50:79:66:68:00
LPORT       : 10006
RHOST:PORT  : 127.0.0.1:10007
MTU:        : 1500
```

Idem pour le PC4 qui obtient l'adresse IP 192.168.10.101/24 :

```
PC4> ip dhcp
DORA IP 192.168.10.101/24 GW 192.168.10.2

PC4> show ip

NAME        : PC4[1]
IP/MASK     : 192.168.10.101/24
GATEWAY     : 192.168.10.2
DNS         : 8.8.8.8  8.8.4.4
DHCP SERVER : 192.168.10.1
DHCP LEASE  : 594, 600/300/525
DOMAIN NAME : example.org
MAC         : 00:50:79:66:68:03
LPORT       : 10012
RHOST:PORT  : 127.0.0.1:10013
MTU:        : 1500
```

Et la connectivité se fait bel et bien entre les machines du même réseau :

```
PC4> ping 192.168.10.100
84 bytes from 192.168.10.100 icmp_seq=1 ttl=64 time=5.414 ms
84 bytes from 192.168.10.100 icmp_seq=2 ttl=64 time=5.306 ms
84 bytes from 192.168.10.100 icmp_seq=3 ttl=64 time=5.859 ms
84 bytes from 192.168.10.100 icmp_seq=4 ttl=64 time=10.412 ms
84 bytes from 192.168.10.100 icmp_seq=5 ttl=64 time=8.654 ms
```

Le précédent test était un test de connection intra-VLAN (puisque le PC1 et le PC4 font parti du même VLAN) alors effectuons à présent un test de connection inter-VLAN :

Une requête sur le PC2 suffit pour que le serveur DHCP lui offre l'adresse IP 192.168.20.101/24 :

```
PC2> ip dhcp
DORA IP 192.168.20.101/24 GW 192.168.20.2

PC2> ping 192.168.10.100

192.168.10.100 icmp_seq=1 timeout
192.168.10.100 icmp_seq=2 timeout
84 bytes from 192.168.10.100 icmp_seq=3 ttl=63 time=5.889 ms
84 bytes from 192.168.10.100 icmp_seq=4 ttl=63 time=4.730 ms
84 bytes from 192.168.10.100 icmp_seq=5 ttl=63 time=6.881 ms
```

Au passage, voici des informations supplémentaires sur le PC2 :

```
PC2> show ip

NAME        : PC2[1]
IP/MASK     : 192.168.20.101/24
GATEWAY     : 192.168.20.2
DNS         : 8.8.8.8  8.8.4.4
DHCP SERVER : 192.168.20.1
DHCP LEASE  : 555, 600/300/525
DOMAIN NAME : example.org
MAC         : 00:50:79:66:68:01
LPORT       : 10008
RHOST:PORT  : 127.0.0.1:10009
MTU:        : 1500
```

On a donc un DHCP opérationnel, attaquons la suite…

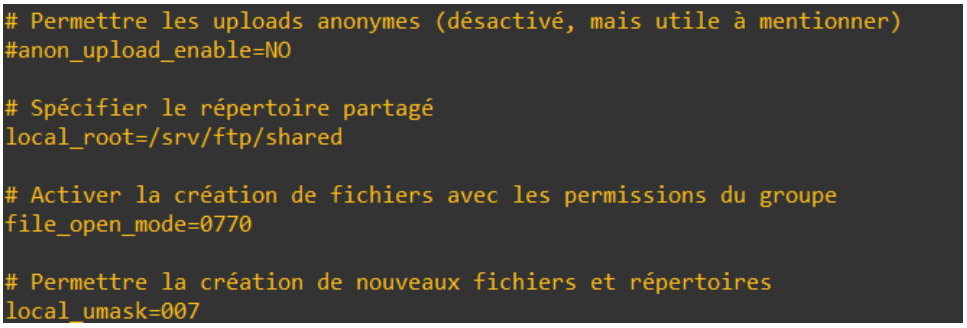
Voici comment on configure et valide notre service FTP opérationnel :

On commence par installer le paquet "vsftpd" qui, parmi tout les choix de serveurs FTP, est un des plus populaires car réputé pour sa sécurité (d'où le "vs" dans son nom pour "Very Secure") :

```
debian@debian:~$ sudo apt install vsftpd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 31 not upgraded.
Need to get 153 kB of archives.
After this operation, 358 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 vsftpd amd64 3.0.3-12+b1 [153 kB]
Fetched 153 kB in 1s (140 kB/s)
Preconfiguring packages ...
```

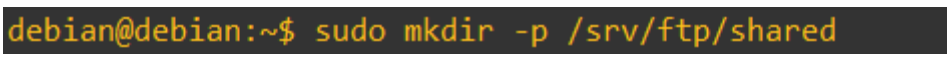
On peut alors configurer le fichier de configuration principal de vsftpd qui est vsftpd.conf :

```
  GNU nano 5.4                    /etc/vsftpd.conf
listen=YES
anonymous_enable=NO
local_enable=YES
ftpd_banner=Bienvenue !

# Permettre aux utilisateurs d'écrire dans le répertoire
write_enable=YES

chroot_local_user=NO
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list

# Limiter les utilisateurs à leur répertoire personnel
chroot_local_user=YES
allow_writeable_chroot=YES

# Permettre les uploads anonymes (désactivé, mais utile à mentionner)
#anon_upload_enable=NO

# Spécifier le répertoire partagé
```



```
# Permettre les uploads anonymes (désactivé, mais utile à mentionner)
#anon_upload_enable=NO

# Spécifier le répertoire partagé
local_root=/srv/ftp/shared

# Activer la création de fichiers avec les permissions du groupe
file_open_mode=0770

# Permettre la création de nouveaux fichiers et répertoires
local_umask=007
```

Puis créer le dossier de partage :



```
debian@debian:~$ sudo mkdir -p /srv/ftp/shared
```

On peut alors créer les utilisateurs de notre service FTP avec ici un exemple de création de l'utilisateur ftpuser2 (la même chose est faite pour ftpuser1) :

```
debian@debian:~$ sudo adduser ftpuser2
Adding user `ftpuser2' ...
Adding new group `ftpuser2' (1003) ...
Adding new user `ftpuser2' (1003) with group `ftpuser2'
Creating home directory `/home/ftpuser2' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ftpuser2
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
debian@debian:~$
```

On peut alors créer le groupe d'utilisateurs du service FTP nommé ftpaccess :

```
debian@debian:~$ sudo groupadd ftpaccess
```

Avant d'y insérer les deux utilisateurs préalablement créés :

```
debian@debian:~$ sudo usermod -aG ftpaccess ftpuser1
debian@debian:~$ sudo usermod -aG ftpaccess ftpuser2
```

Il faut également que le dossier de partage appartienne au groupe des utilisateurs du service FTP :

```
debian@debian:~$ sudo chgrp ftpaccess /srv/ftp/shared
debian@debian:~$ sudo chmod 770 /srv/ftp/shared
```

Il n'y a donc plus aucune raison que les différents utilisateurs du service FTP ne puisse l'utiliser :

Ici l'utilisateur ftpuser1 partage le fichier test4 :

```
ftp> put test4
local: test4 remote: test4
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
ftp> ls -l
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rwxrwx---    1 1002     1002            0 Jun 14 10:17 test
-rwxrwx---    1 1003     1003            0 Jun 14 10:25 test2
-rwxrwx---    1 1002     1002            0 Jun 14 10:41 test3
-rwxrwx---    1 1002     1002            0 Jun 14 10:48 test4
226 Directory send OK.
```

Et l'utilisateur ftpuser2 le récupère correctement :

```
ftp> get test4
local: test4 remote: test4
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for test4 (0 bytes).
226 Transfer complete.
ftp>
```

On peut donc dire que le fichier test4 a été transféré avec succès.

On a donc un FTP opérationnel, attaquons la suite…

Voici comment on configure et valide notre service HTTP opérationnel :

On commence par installer le paquet "apache2" puis démarrer le service :

```
debian@debian:/var/www$ sudo systemctl start apache2
```

Un service fonctionnant correctement :

```
debian@debian:/var/www$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
     Active: active (running) since Fri 2024-06-14 06:20:08 UTC; 1h 46min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 351 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC>
   Main PID: 371 (apache2)
      Tasks: 55 (limit: 546)
     Memory: 7.4M
        CPU: 872ms
     CGroup: /system.slice/apache2.service
             ├─371 /usr/sbin/apache2 -k start
             ├─374 /usr/sbin/apache2 -k start
             └─375 /usr/sbin/apache2 -k start

Jun 14 06:20:06 debian systemd[1]: Starting The Apache HTTP Server...
Jun 14 06:20:08 debian apachectl[365]: AH00558: apache2: Could not reliably det>
Jun 14 06:20:08 debian systemd[1]: Started The Apache HTTP Server.
lines 1-17/17 (END)
```

On édite alors le fichier 000-default.conf afin de modifier la racine du serveur web qui sera donc sae201.com. Le répertoire racine du site web, est celui où le serveur Apache cherchera les fichiers lorsqu'il recevra une requête HTTP :

```
  GNU nano 5.4                    000-default.conf *
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port t>
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/sae201.com

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Il faut également changer les droits du répertoire sae201.com pour que le serveur Web ait les autorisations d'accès au répertoire et ce qu'il contient, c'est faisable de cette manière :

```
debian@debian:/etc/apache2/sites-available$ sudo chown -R www-data:www-data /var/www/sae201.com
debian@debian:/etc/apache2/sites-available$ sudo chmod -R 755 /var/www/sae201.com
```

On peut alors créer un fichier index. html dans le répertoire sae201.com :

```
  GNU nano 5.4                          index.html *
<!DOCTYPE html>
<html>
<head>
    <title>SAE201</title>
</head>
<body>
    <h1>Serveur Web de la SAE201</h1>
    <p>Bienvenue sur notre site !</p>
</body>
</html>
```

Les tests du services HTTP sont effectués une fois pfsense appliqué à toute l'infrastructure.

On a donc un HTTP opérationnel, attaquons la suite…

Voici comment on configure et valide notre service SSH opérationnel :

On commence par installer les paquets pour faire fonctionner notre serveur SSH :

```
debian@debian:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libcbor0
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libcbor0.8 libfido2-1 libssl3 openssh-client openssh-sftp-server
  runit-helper
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard ufw
Recommended packages:
  xauth ncurses-term
The following NEW packages will be installed:
  libcbor0.8 libssl3
The following packages will be upgraded:
  libfido2-1 openssh-client openssh-server openssh-sftp-server runit-helper
5 upgraded, 2 newly installed, 0 to remove and 272 not upgraded.
Need to get 3641 kB of archives.
After this operation, 8278 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Puis les paquets pour faire fonctionner nos clients ssh :

```
debian@debian:~$ sudo apt install openssh-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-client is already the newest version (1:8.4p1-5+deb11u3).
openssh-client set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
```

Après cela on active la fonction Permitrootlogin pour que le compte root puisse connecter directement via SSH :

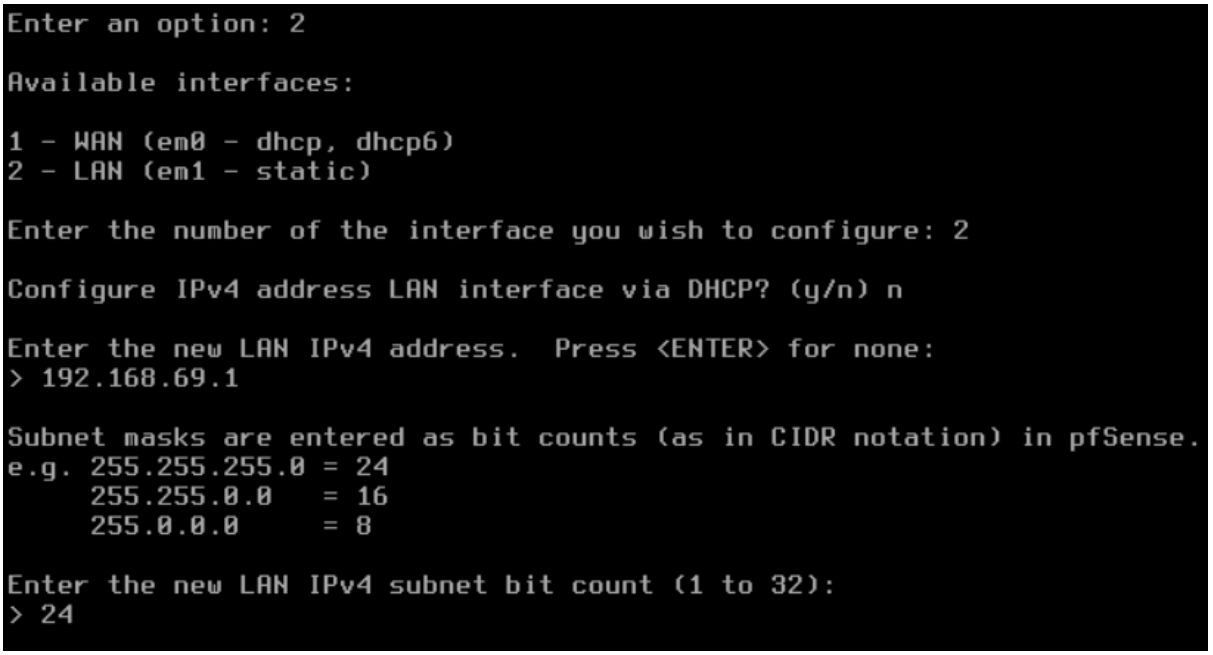
```
  GNU nano 5.4                  /etc/ssh/sshd_config
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Les tests du services SSH sont effectués une fois pfsense appliqué à toute l'infrastructure.

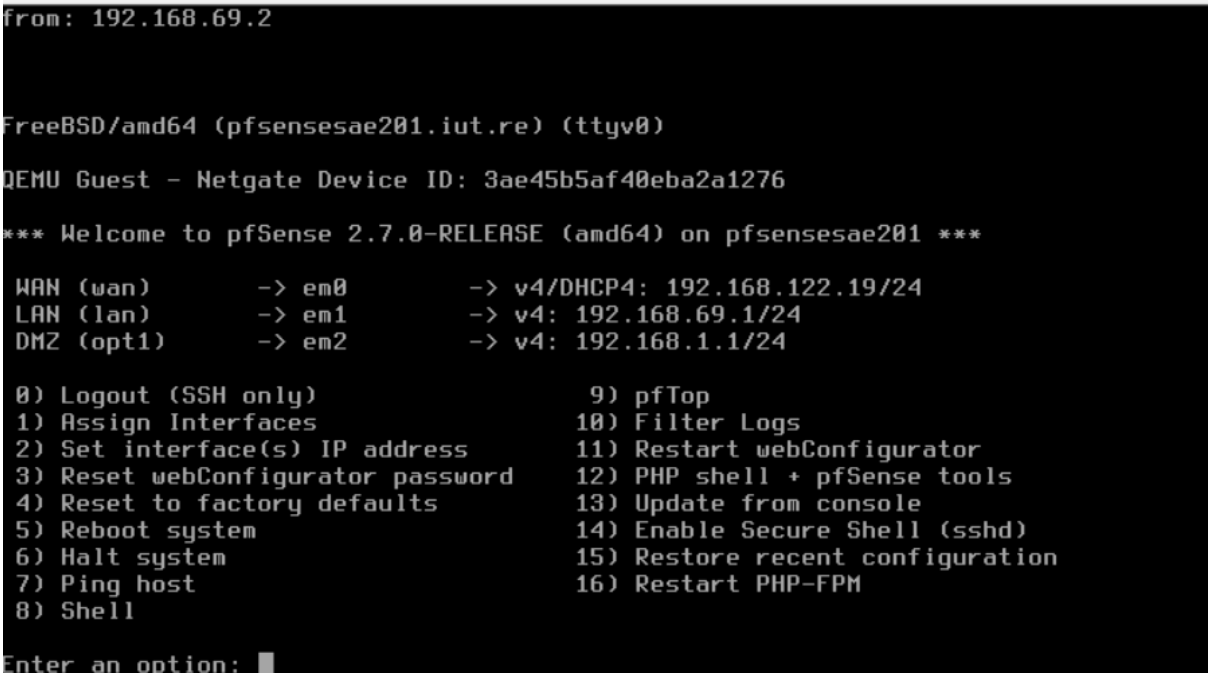
On a donc un SSH opérationnel, attaquons la suite…

Configuration de pfsense

On commence par configurer les adresses ip des interfaces du routeur pfsense :

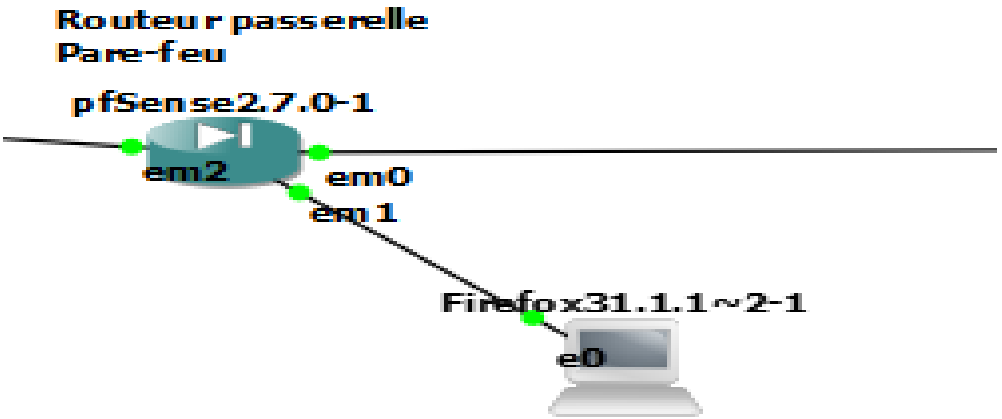


Et voilà donc nos interfaces configurées :

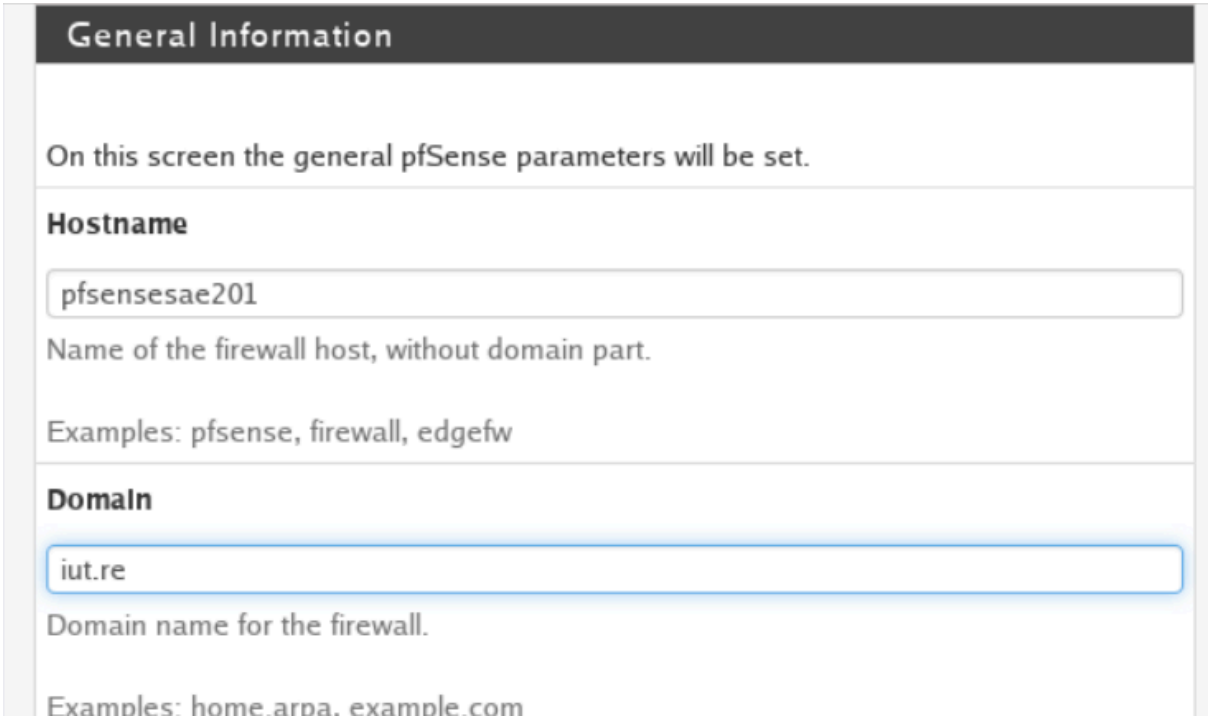


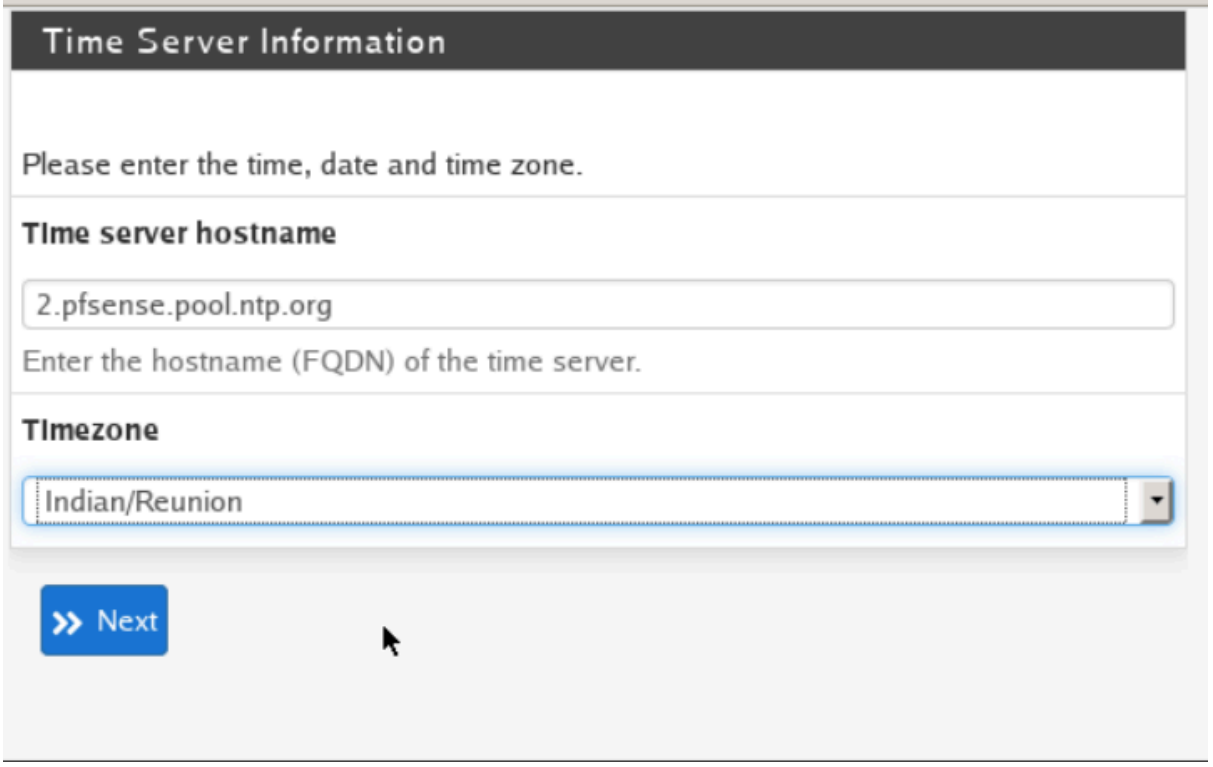
On peut à présent configurer pfsense graphiquement :

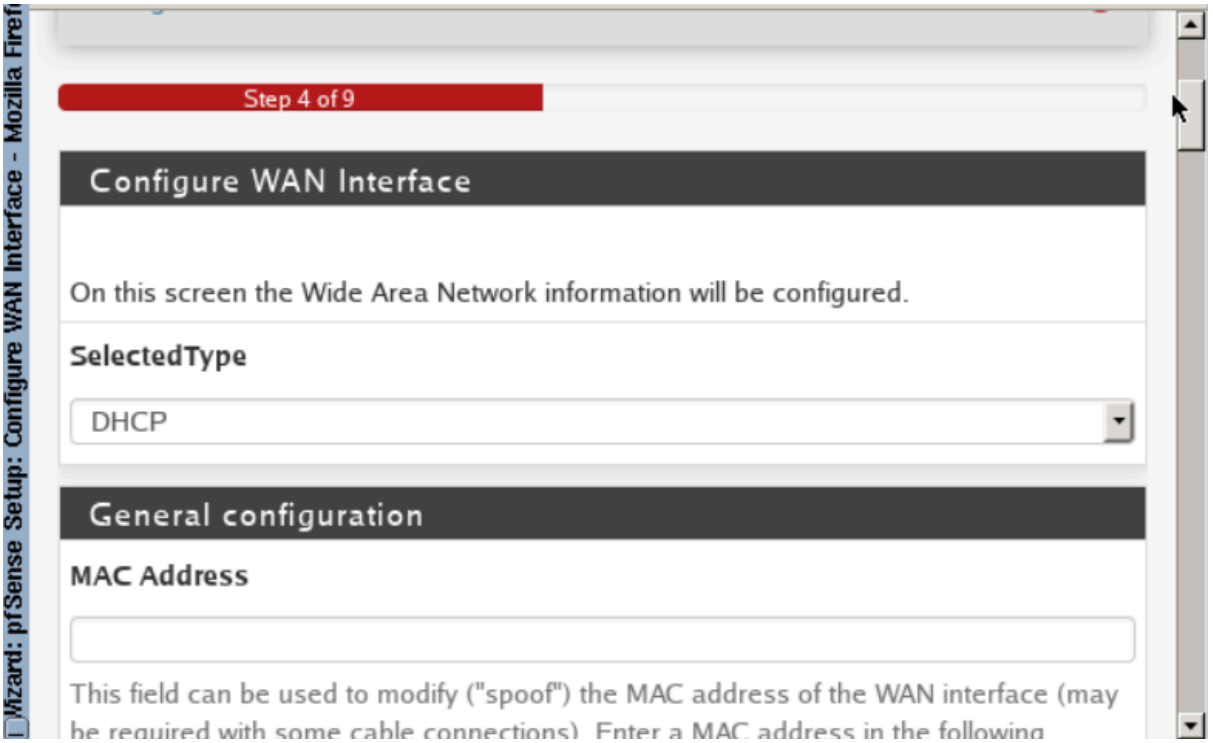
On décide de configurer graphiquement pfsense depuis l'interface LAN afin de faciliter cette tâche :

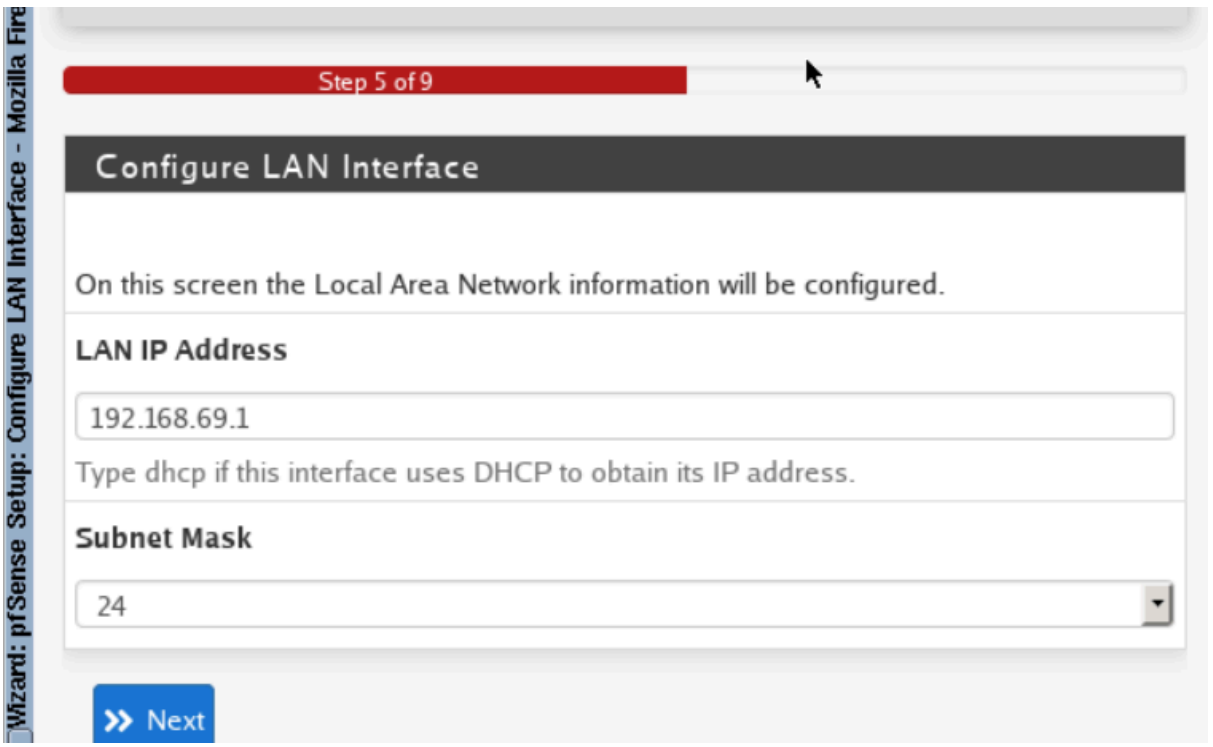


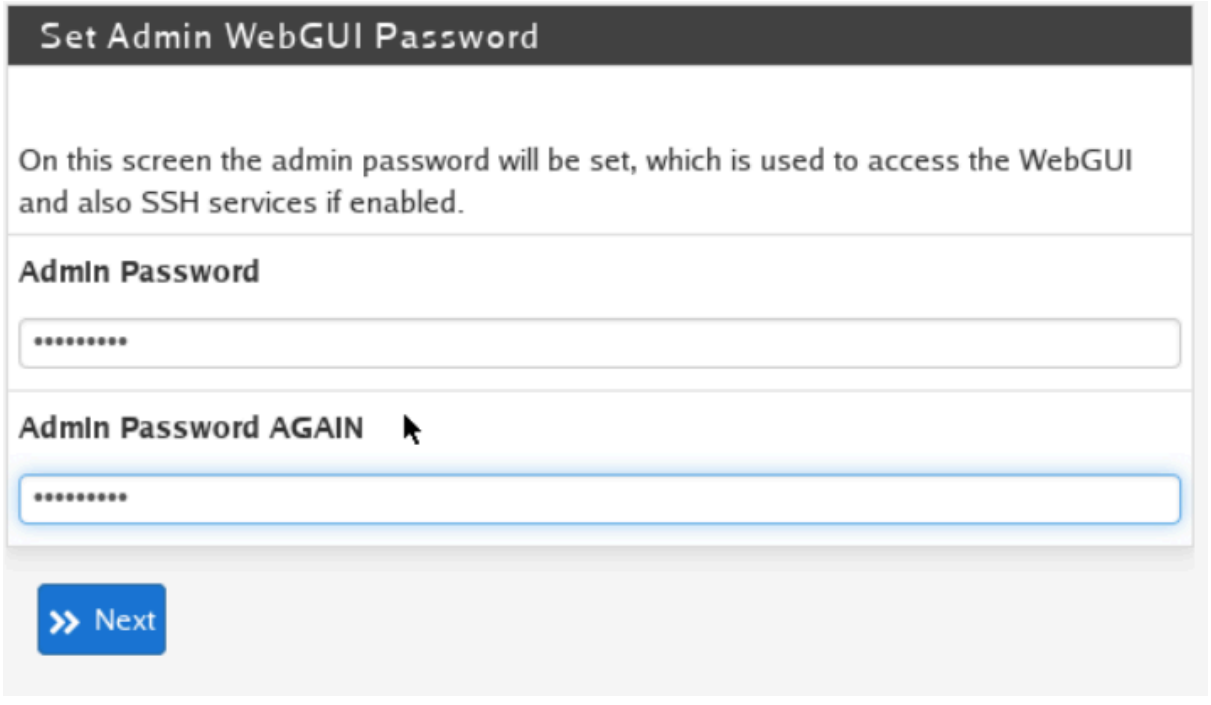
Voici donc le commencement de la configuration de pfsense où l'on définit les différents paramètres de notre pare feu et autres :

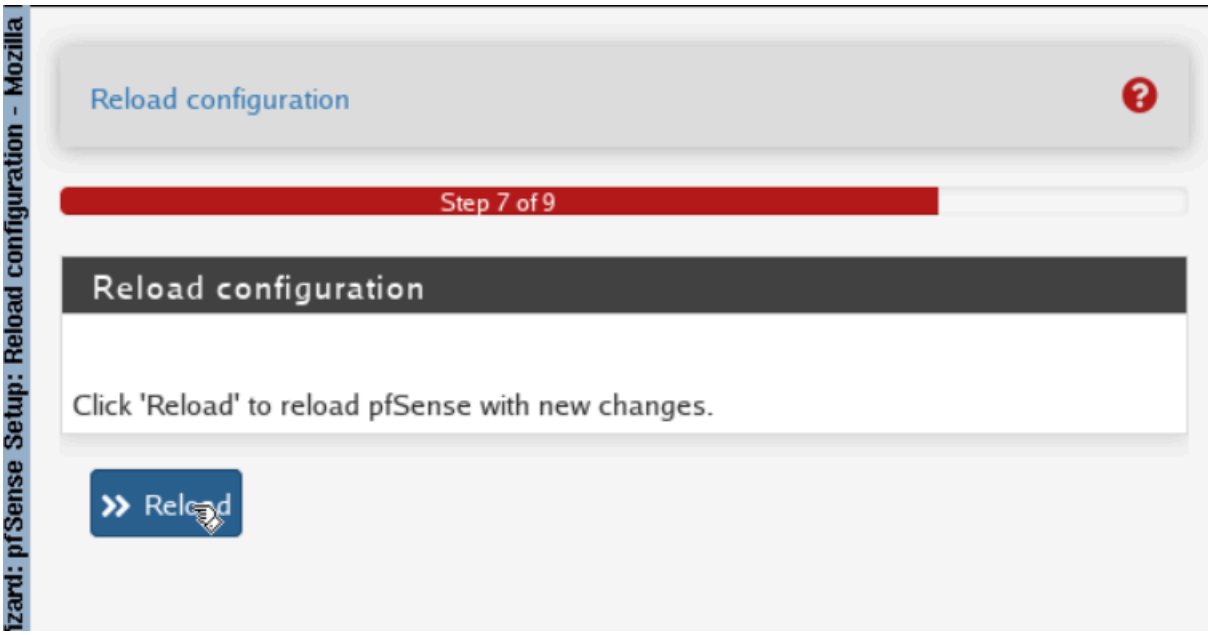




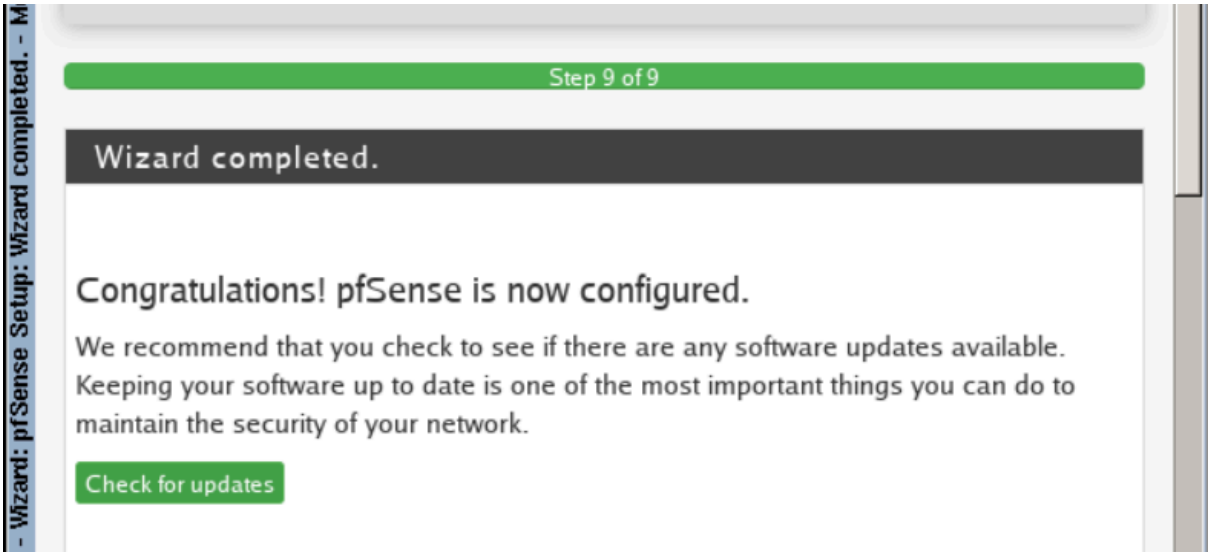

Wizard: pfSense Setup: Configure WAN Interface - Mozilla Firef
Step 4 of 9
Configure WAN Interface
On this screen the Wide Area Network information will be configured.
SelectedType
DHCP
General configuration
MAC Address
This field can be used to modify ("spoof") the MAC address of the WAN interface (may


Wizard: pfSense Setup: Configure LAN Interface - Mozilla Fire
Step 5 of 9
Configure LAN Interface
On this screen the Local Area Network information will be configured.
LAN IP Address
192.168.69.1
Type dhcp if this interface uses DHCP to obtain its IP address.
Subnet Mask
24
Next

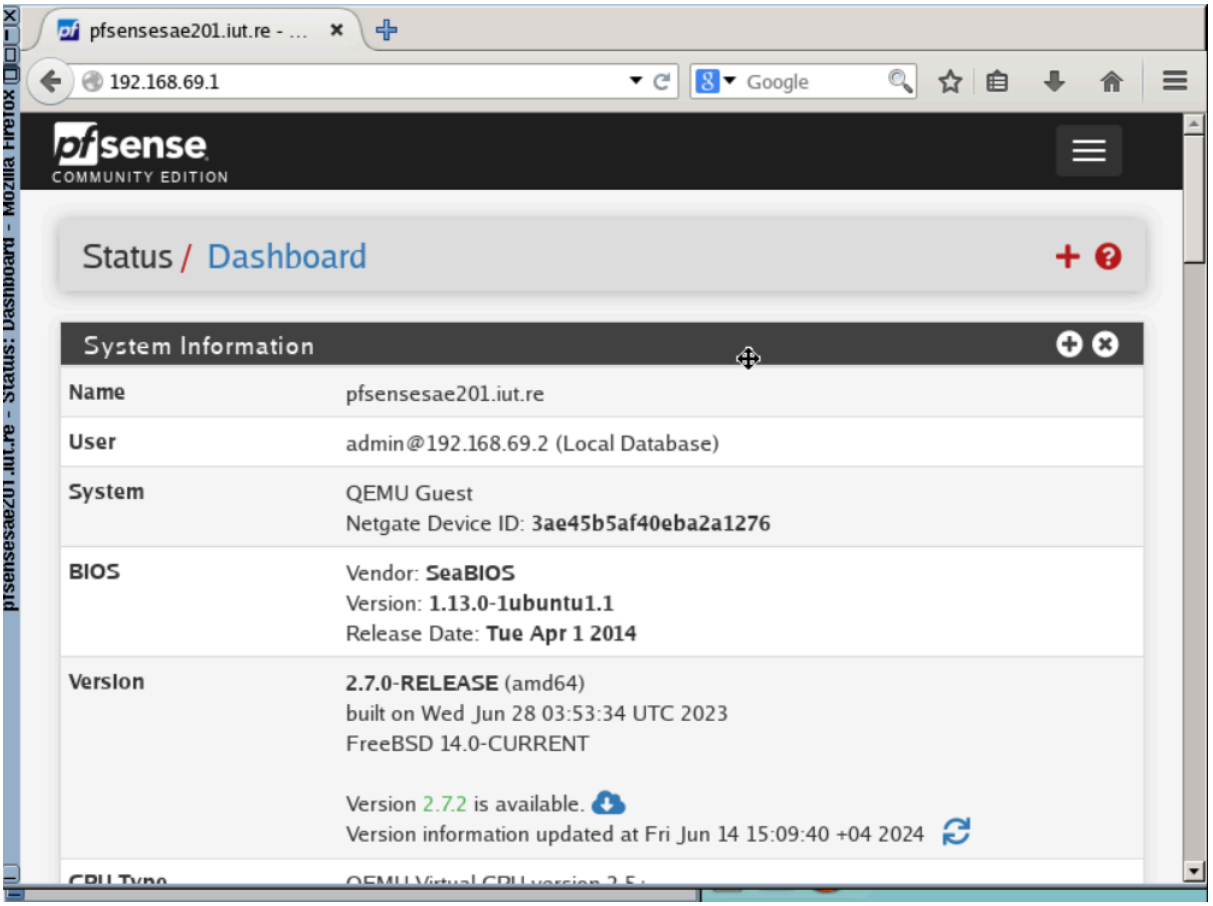

Set Admin WebGUI Password
On this screen the admin password will be set, which is used to access the WebGUI and also SSH services if enabled.
Admin Password
Admin Password AGAIN
Next


Wizard: pfSense Setup: Reload configuration - Mozilla
Reload configuration
Step 7 of 9
Reload configuration
Click 'Reload' to reload pfSense with new changes.
Reload

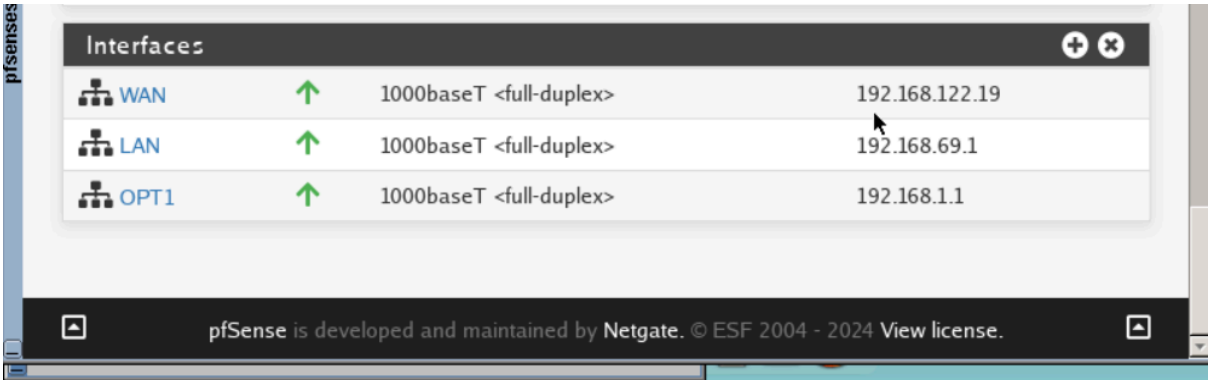
Et voici donc notre pare feu configuré :



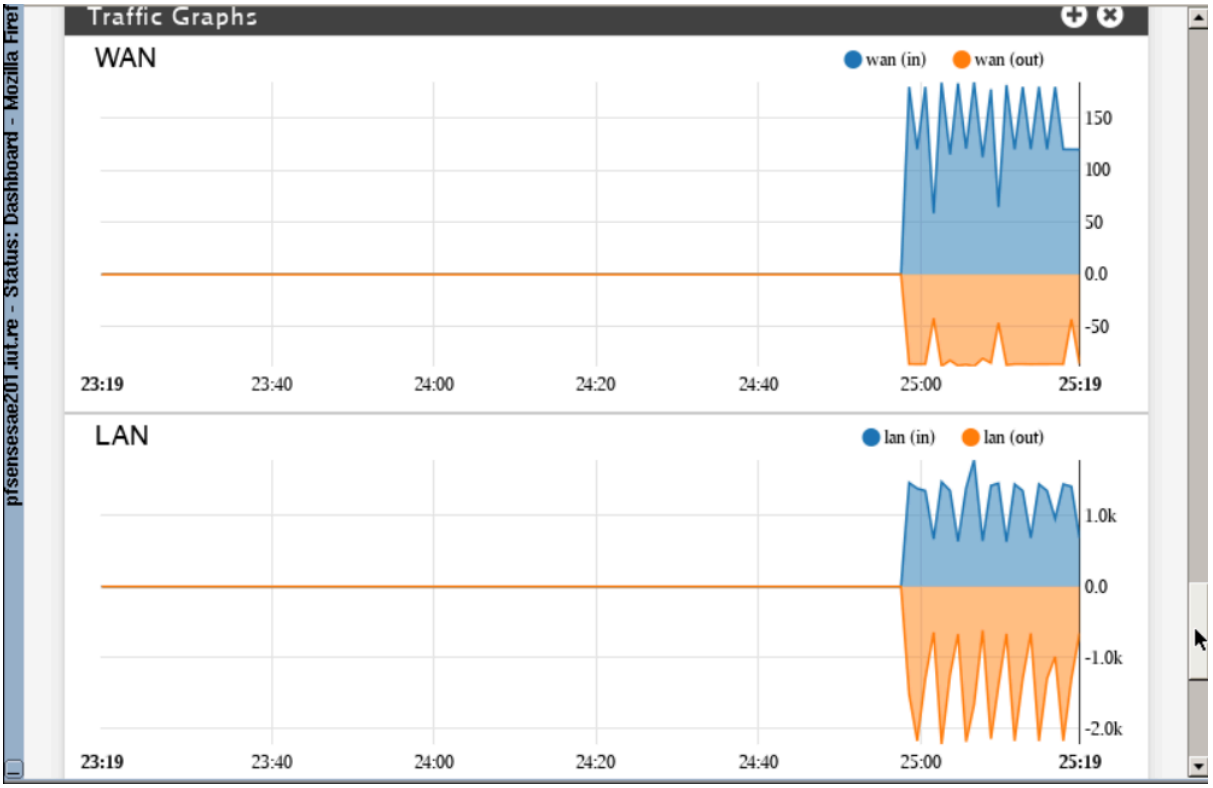
On arrive donc sur cette page affichant des informations sur notre configuration :



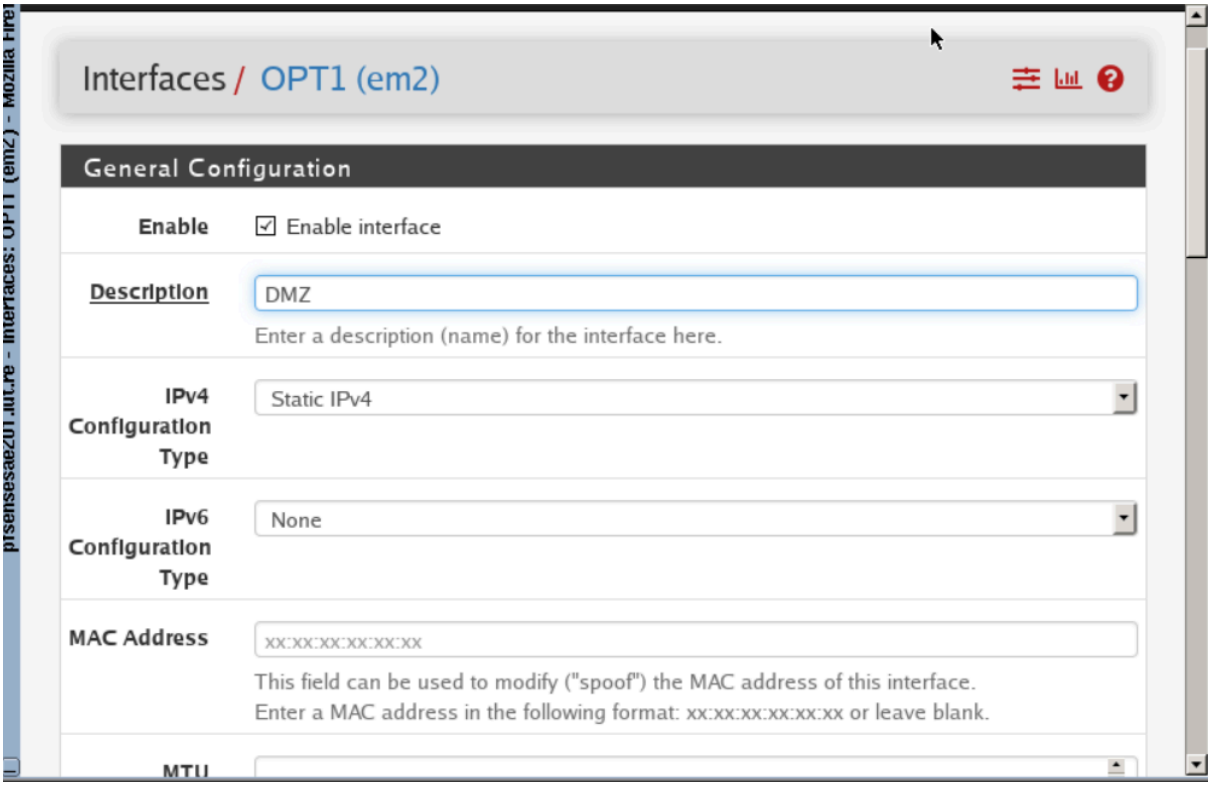
On peut également accéder à des informations sur nos interfaces :

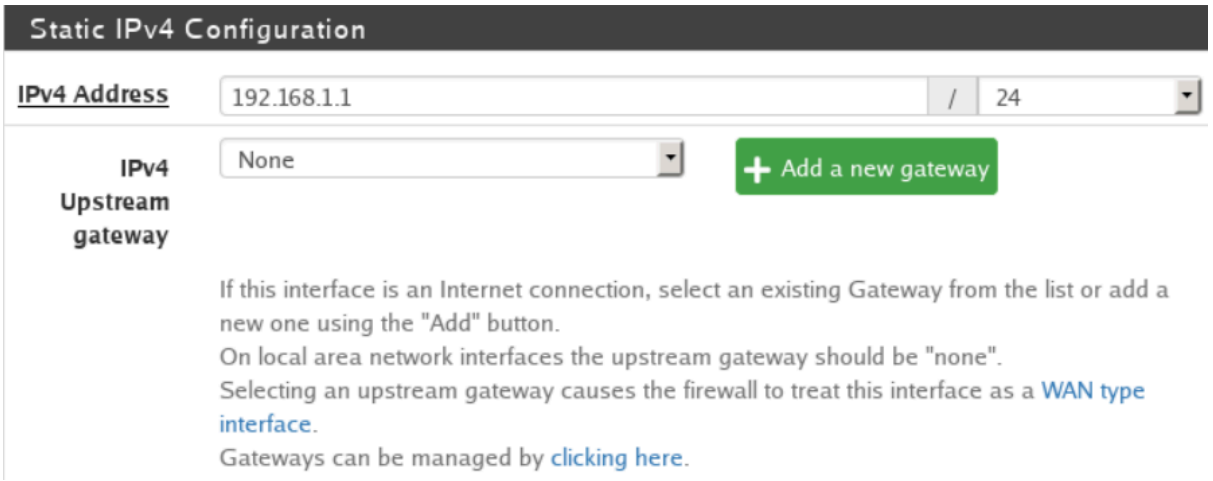


On ajoute le widget Traffic graph pour afficher l'activité de chaque réseaux :

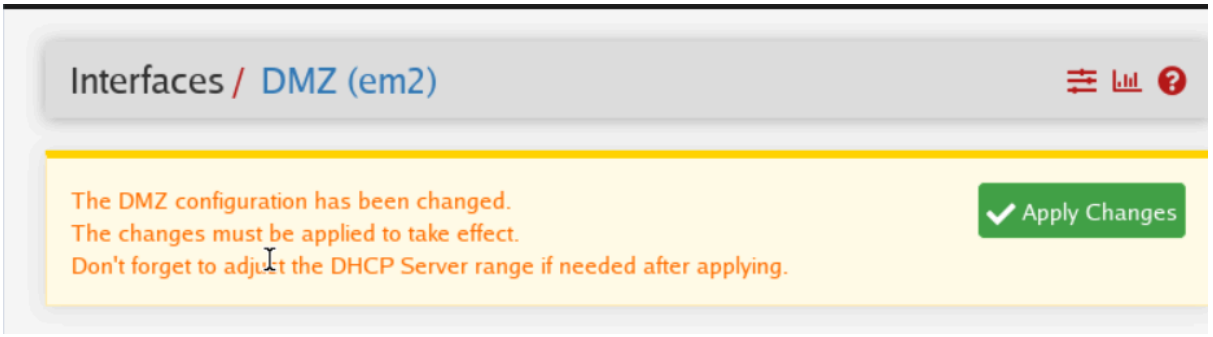


Et on peut à présent débuter la configuration de la DMZ :

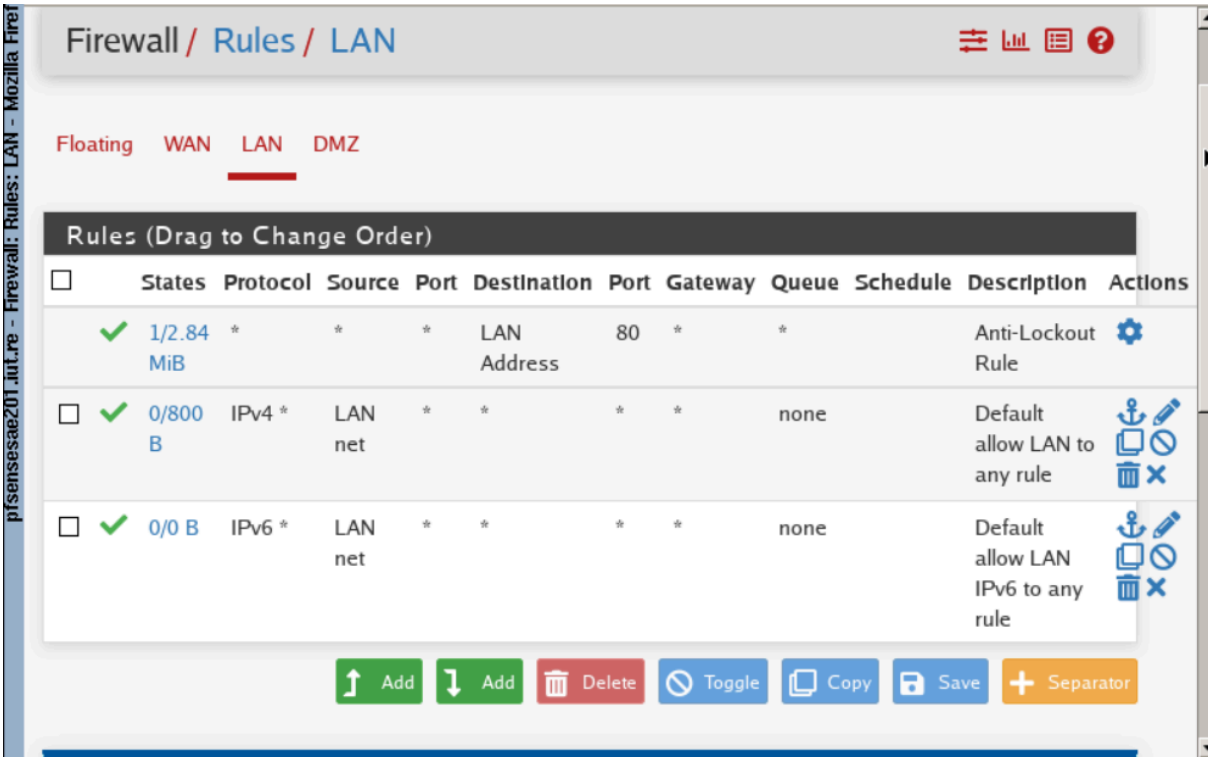




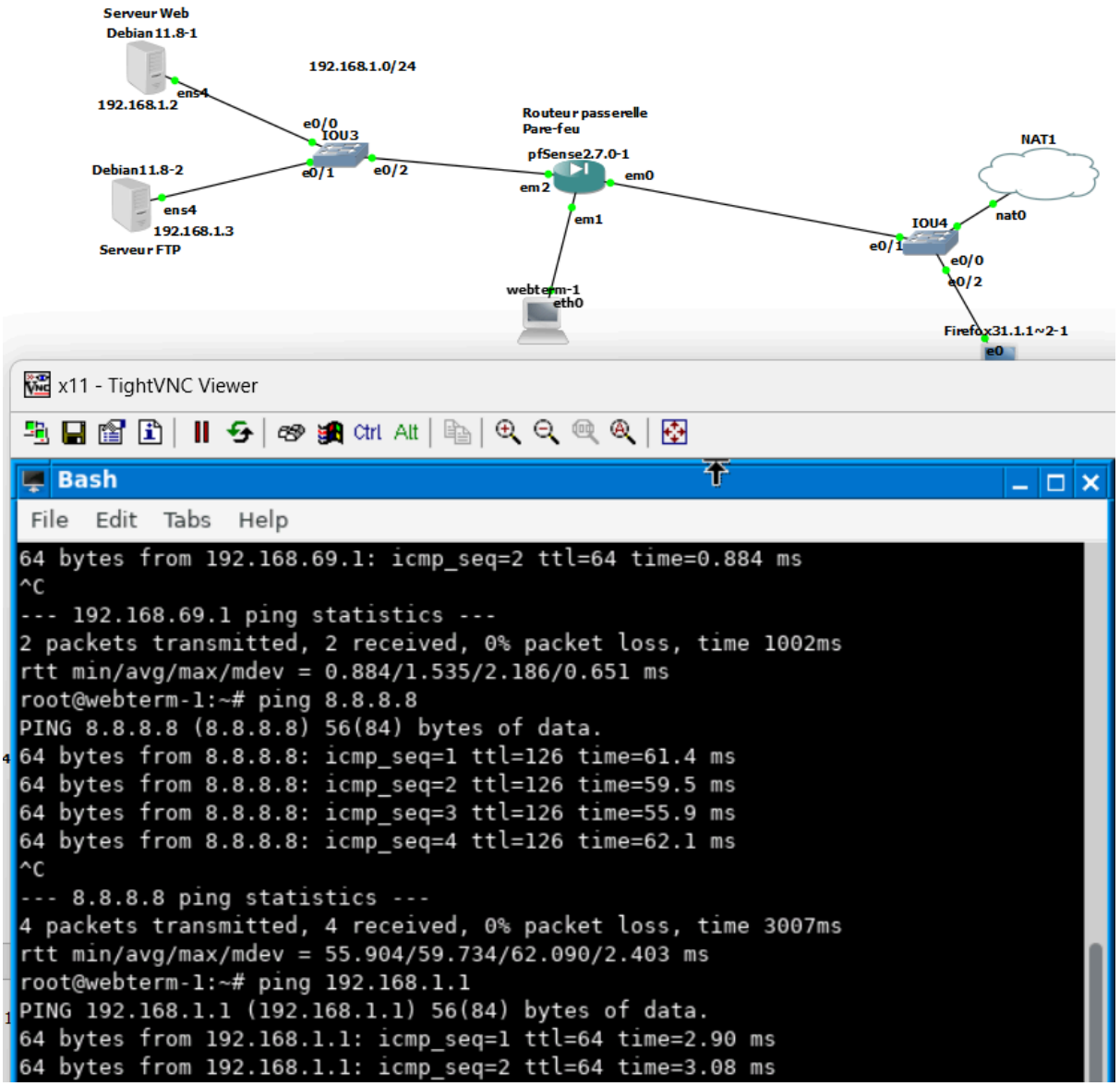
Configuration sauvegardée et appliquée :



On peut donc attaquer la configuration des règle du pare feu selon l'interface, ici le LAN est configuré pour accéder à tous les réseaux :



Et voici des pings du LAN qui est bel et bien autorisée à se connecter à toutes les machines (LAN qui est représenté par la machine reliée par em1) :

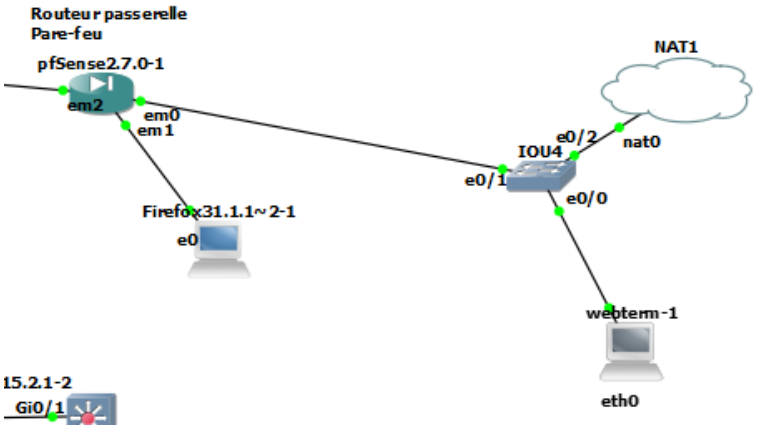


Et voici un test de connectivité de la DMZ qui n'est donc pas autorisée à accéder à internet sans les règles :

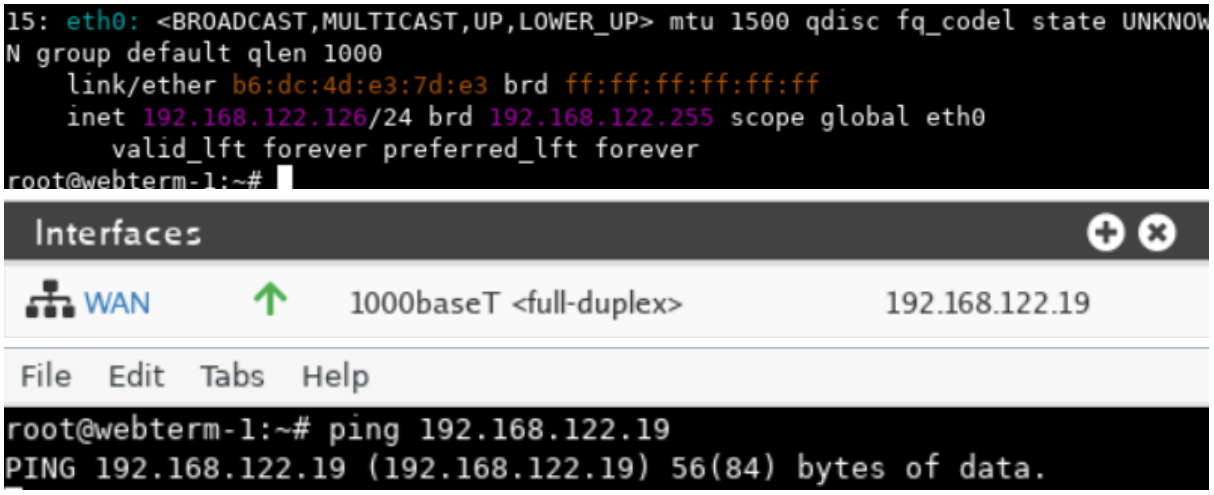
```
debian@debian:/home$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^Xc^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4088ms
```

On peut donc configurer des règle sur la DMZ qui n'en a pas pour le moment :

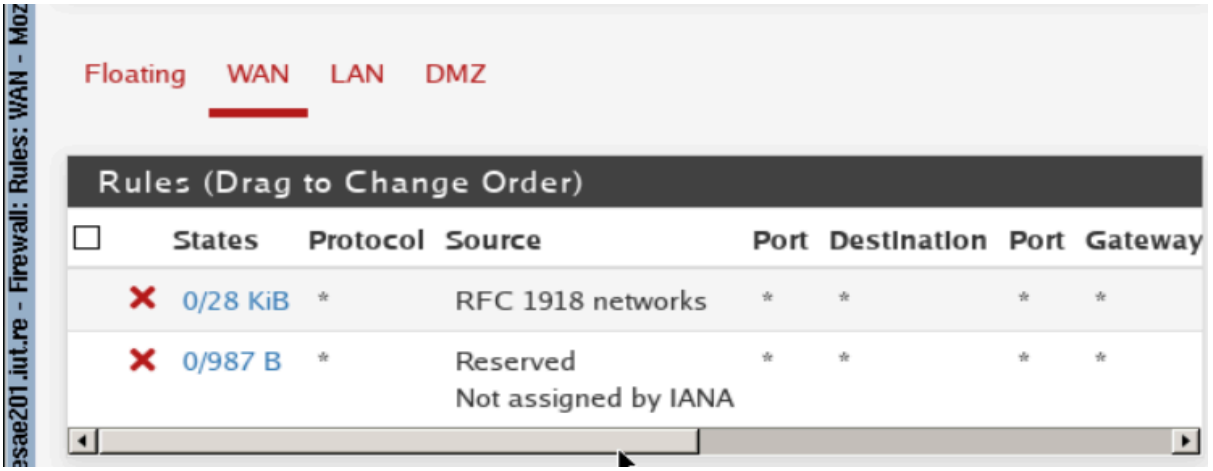
On effectue donc des modifications dans la topologie pour faire des test entre le NAT, la DMZ et le LAN :

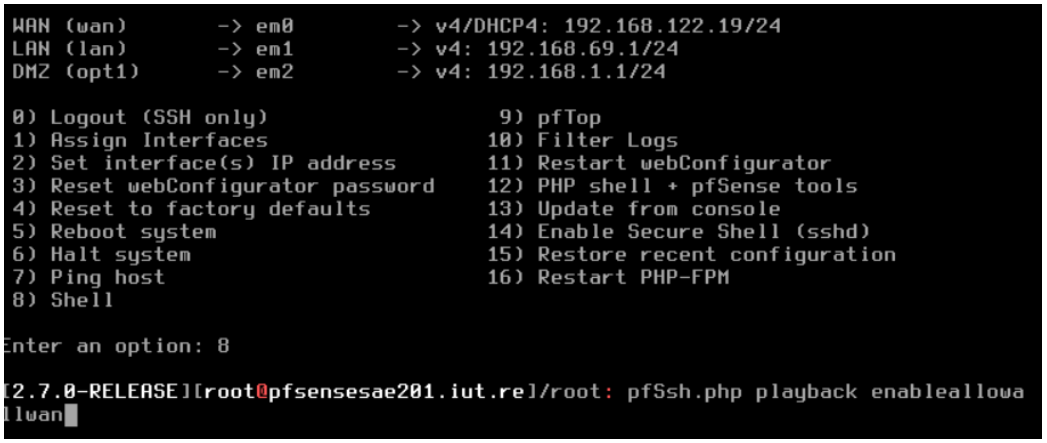


Voici donc une machine du WAN tentant de ping l'interface routeur WAN depuis le WAN :

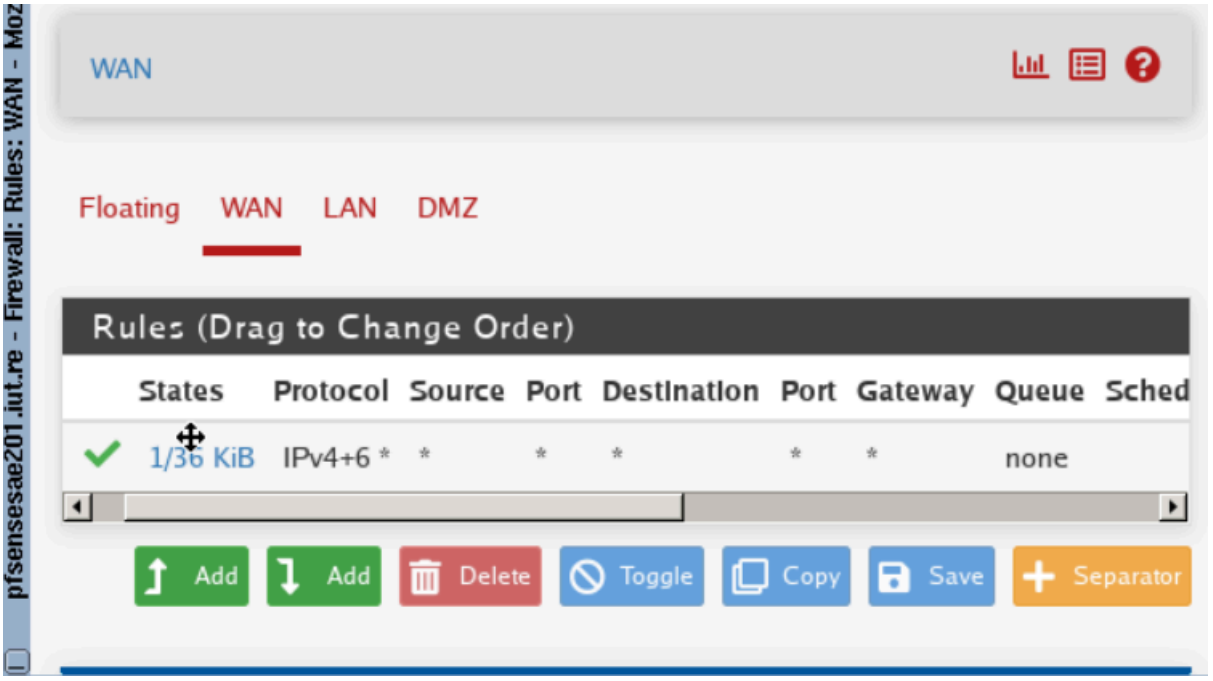


On va modifier les règle du WAN pour que le WAN puisse accéder à la DMZ :





Et donc on obtient après la commande :

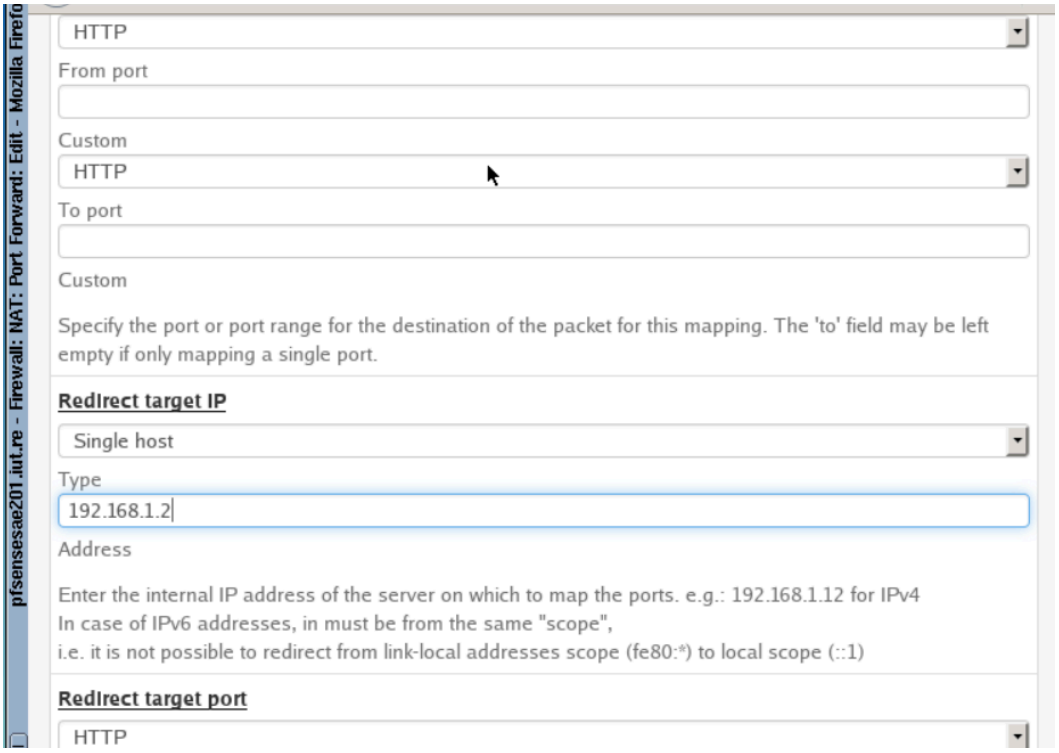


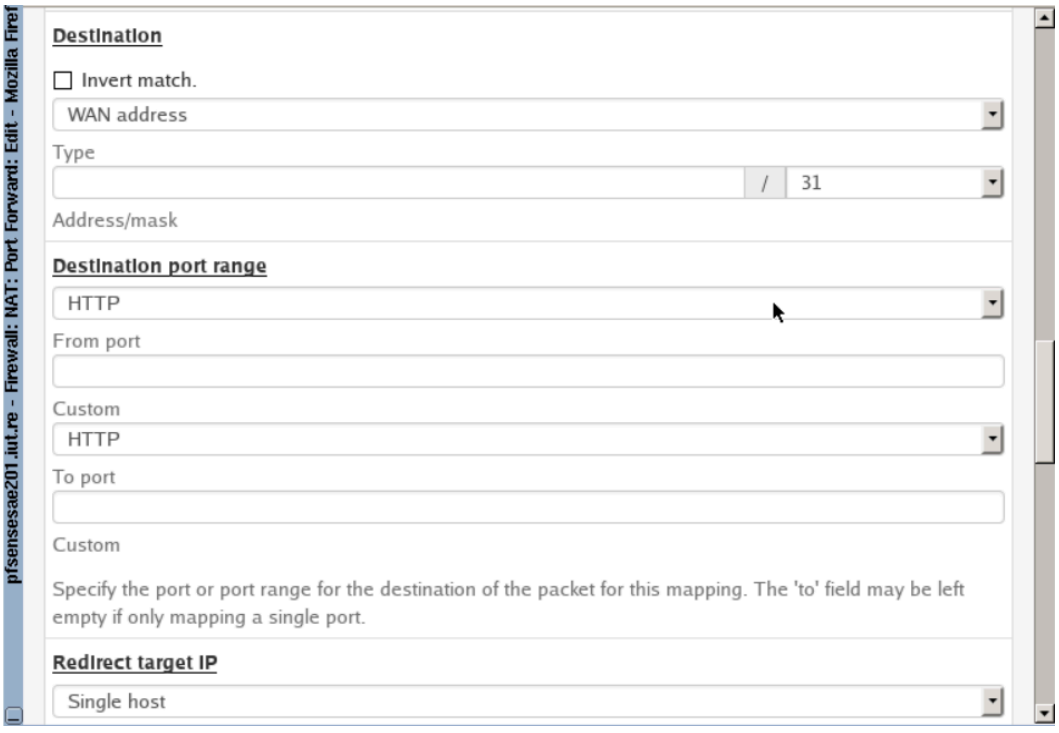
On peut à présent accéder à l'interface WAN du routeur pfsense depuis le NAT :

```
root@webterm-1:~# ping 192.168.122.19
PING 192.168.122.19 (192.168.122.19) 56(84) bytes of data.
64 bytes from 192.168.122.19: icmp_seq=1 ttl=64 time=1.69 ms
64 bytes from 192.168.122.19: icmp_seq=2 ttl=64 time=0.955 ms
64 bytes from 192.168.122.19: icmp_seq=3 ttl=64 time=0.641 ms
```

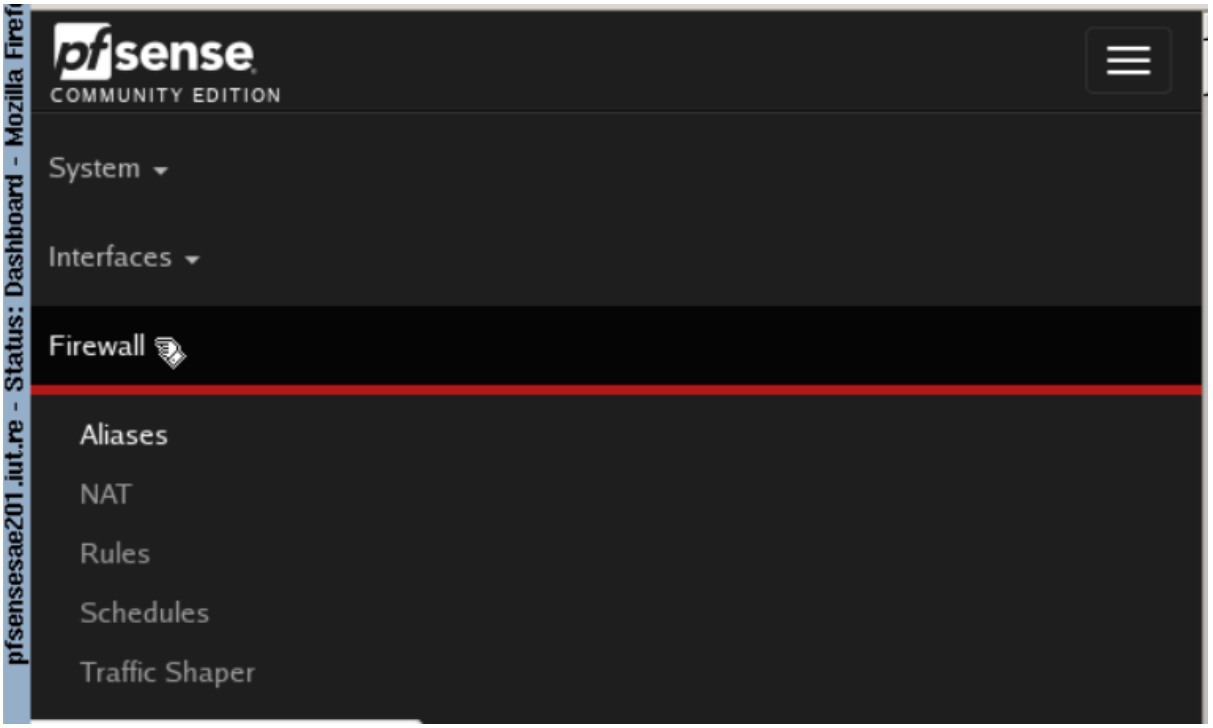
On va configurer les services de la DMZ pour que le WAN y accède.

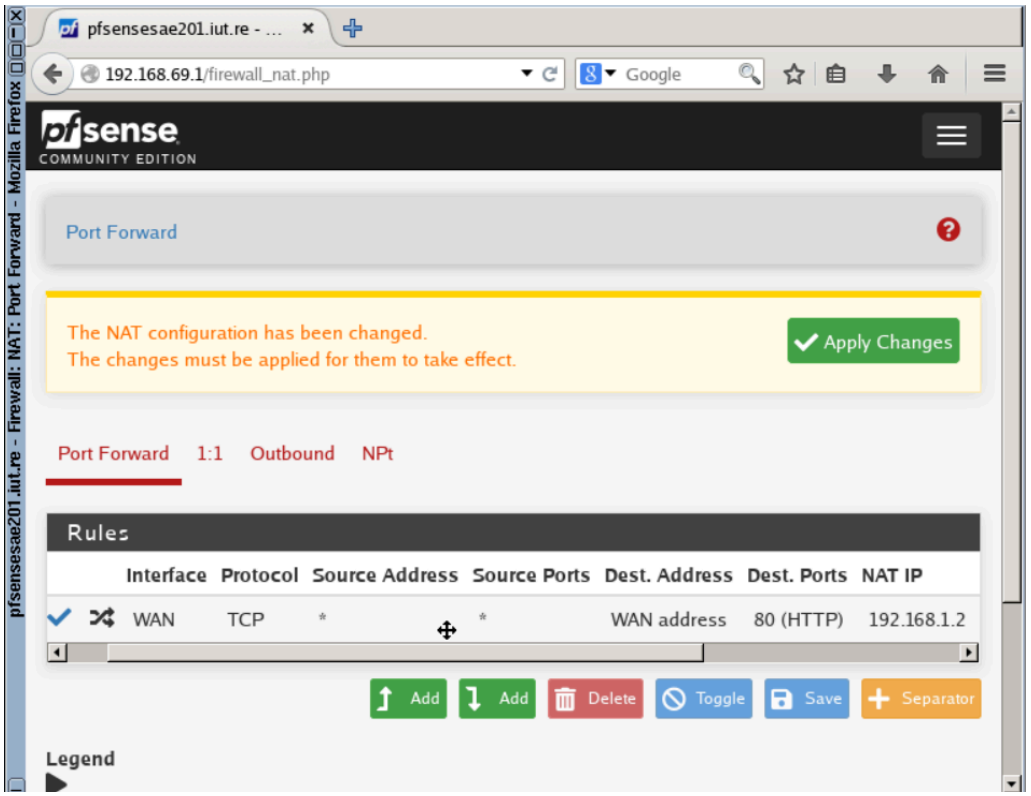
On configure le paramètre NAT du Pare feu pour ajouter lesdits services :



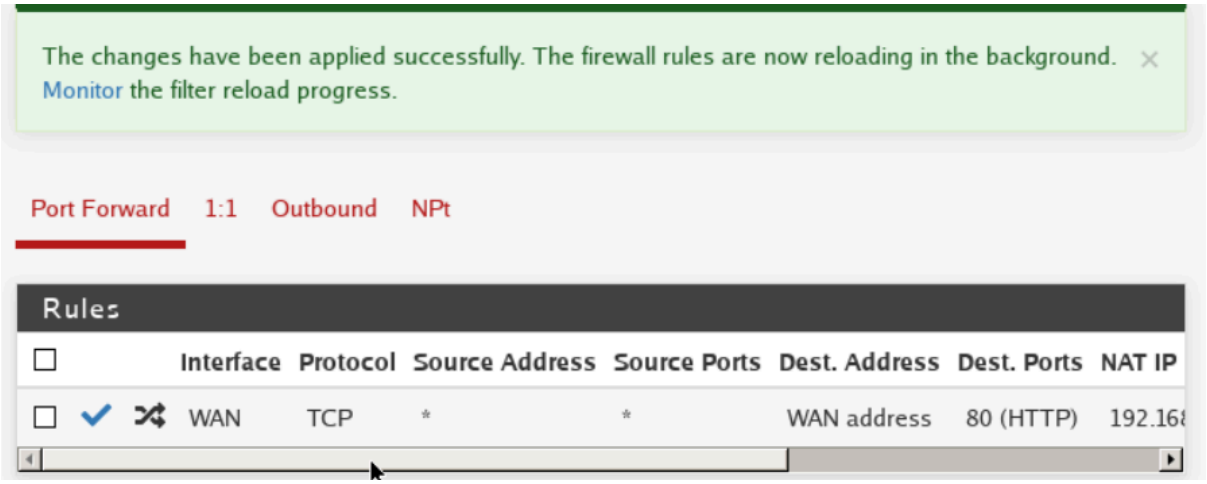


Et voici l'ajout des règles :

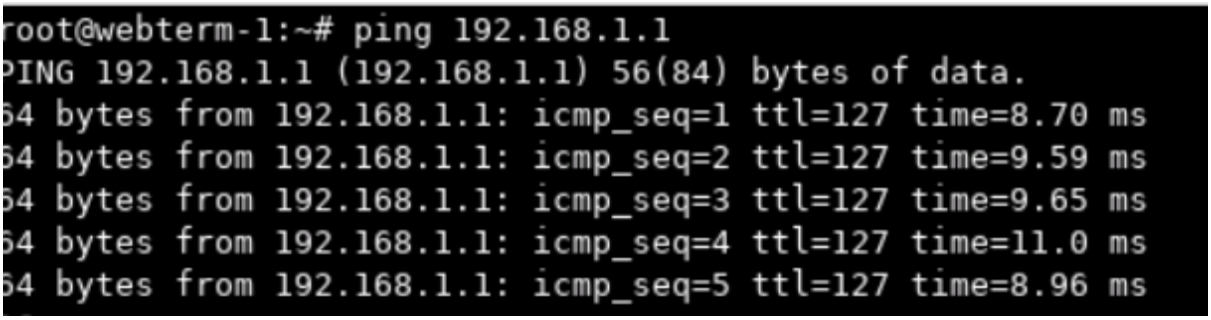




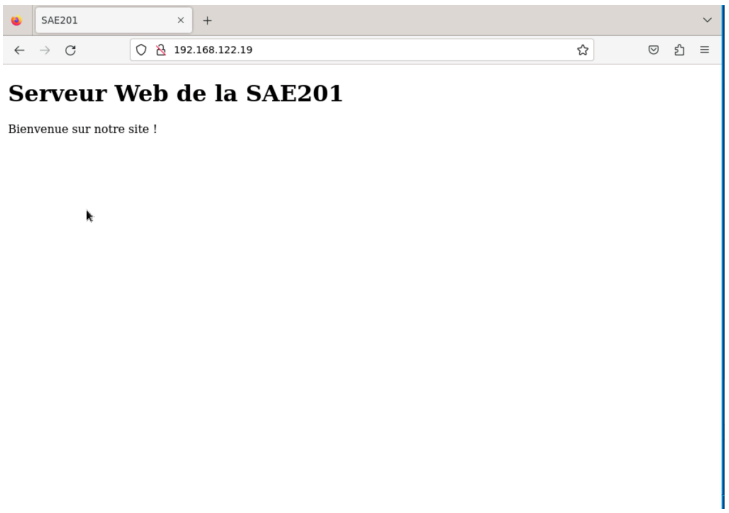
Nos configurations sont bel et bien sauvegardées :



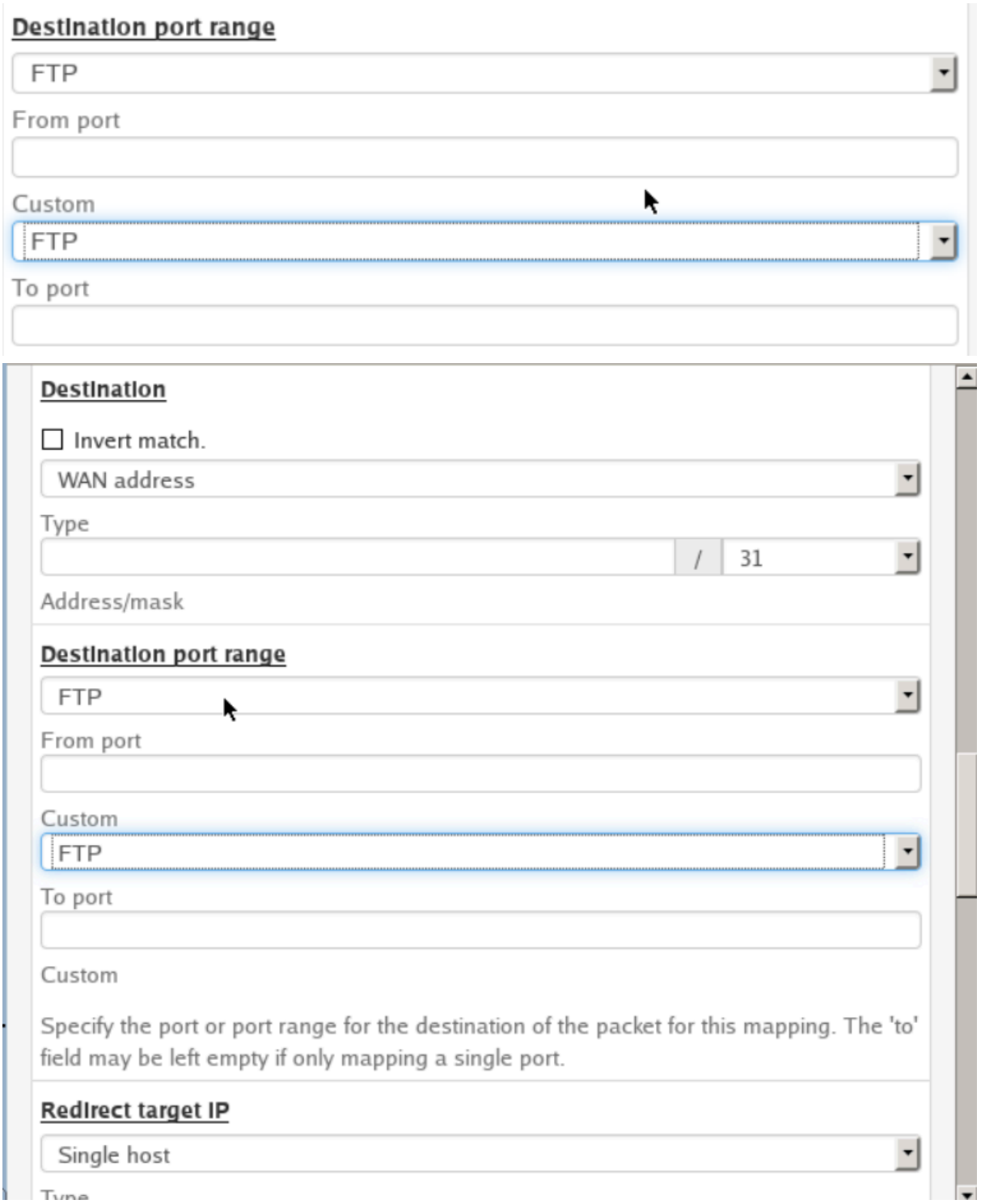
Et le WAN peut donc accéder à la DMZ :

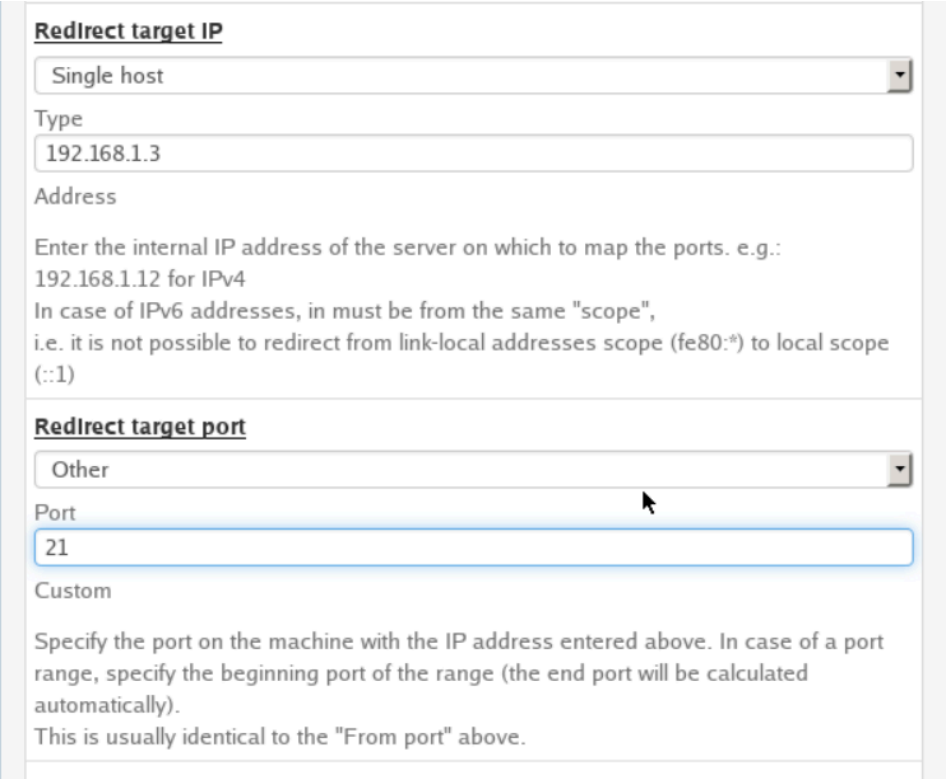


Le protocole HTTP est donc fonctionnel depuis le NAT vers la DMZ :

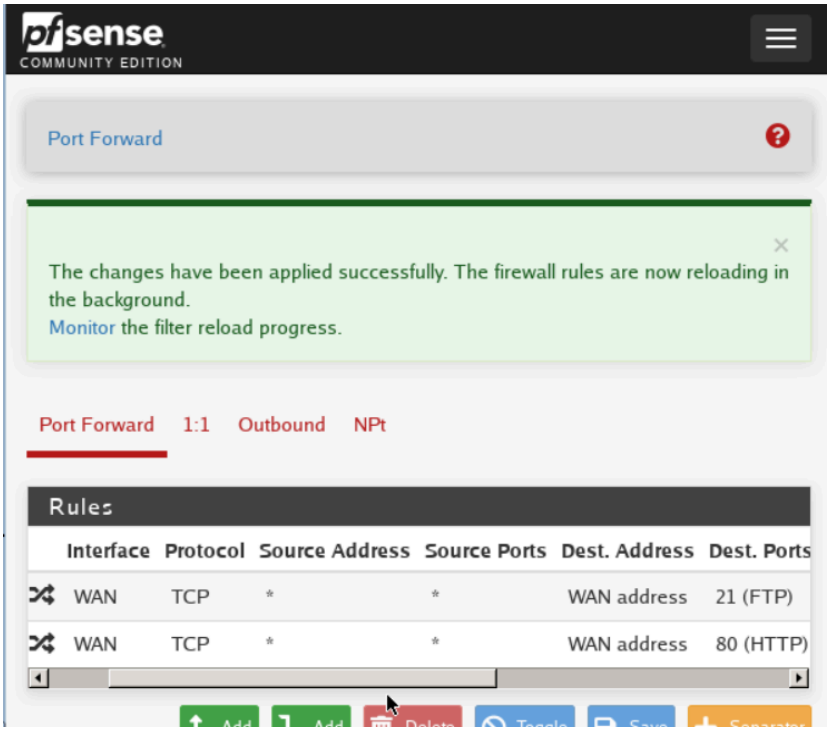


On peut donc faire la même chose que le HTTP pour le FTP :





On sauvegarde nos configurations :



Et on peut alors tester le ftp depuis le WAN (NAT) :

```
root@webterm-1:~# apt install ftp
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  tnftp
The following NEW packages will be installed:
  ftp tnftp
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 166 kB of archives.
After this operation, 319 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
root@webterm-1:~# ftp 192.168.122.19
Connected to 192.168.122.19.
220 Bienvenue !
Name (192.168.122.19:root): ftpuser1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -l
```

Voici les vérification du fonctionnement de ftp sur la machine NAT :

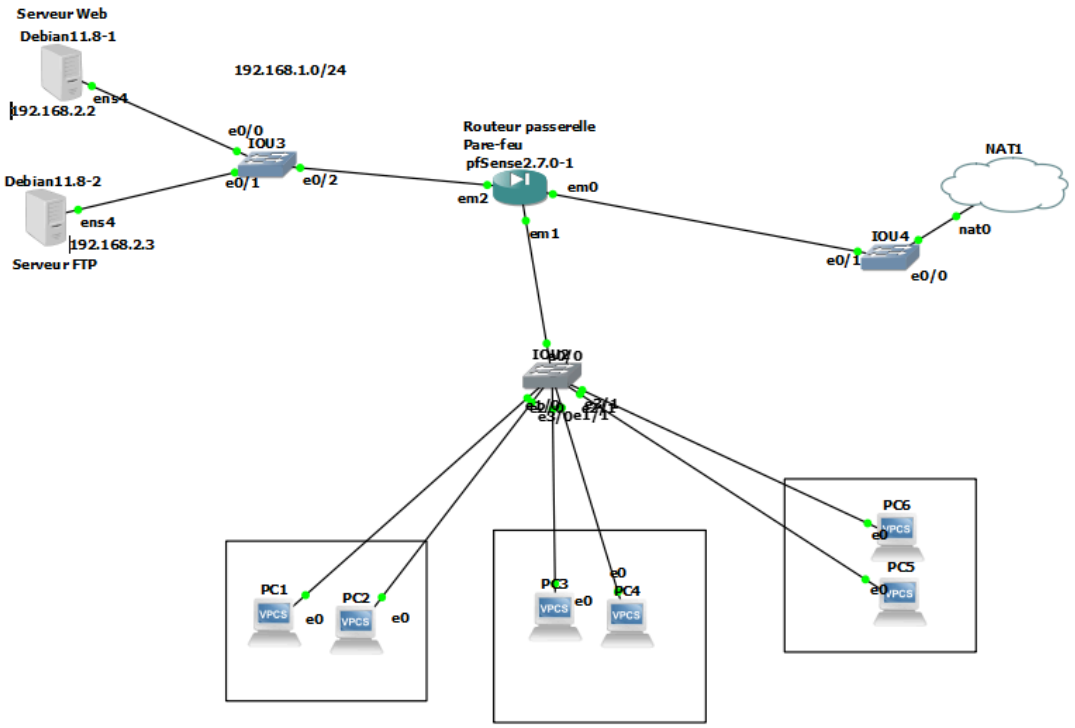
```
ftp> ls -l
229 Entering Extended Passive Mode (|||29378|)
150 Here comes the directory listing.
-rwxrwx---    1 1002     1004            0 Jun 14 10:17 test
-rwxrwx---    1 1003     1004            0 Jun 14 10:25 test2
-rwxrwx---    1 1002     1004            0 Jun 14 10:41 test3
-rwxrwx---    1 1002     1004            0 Jun 14 10:48 test4
226 Directory send OK.
ftp>
```

On peut à présent configurer les routes du switch vers les différents réseaux via pfsense :

```
Switch(config)#ip route 0.0.0.0 0.0.0.0 192.168.69.1
Switch(config)#
```

DE LA RÉUNION Institut Universitaire de Technologie	**Projet de la SAE2.01**	**Objectif** Mise en place d'une infrastructure de réseau d'entreprise	**ALAMELOU** **BOUGHLEM** **CHEREAU**

On fait également le choix de passer à la nouvelle topologie proposée pour la SAE qui est également autorisée :

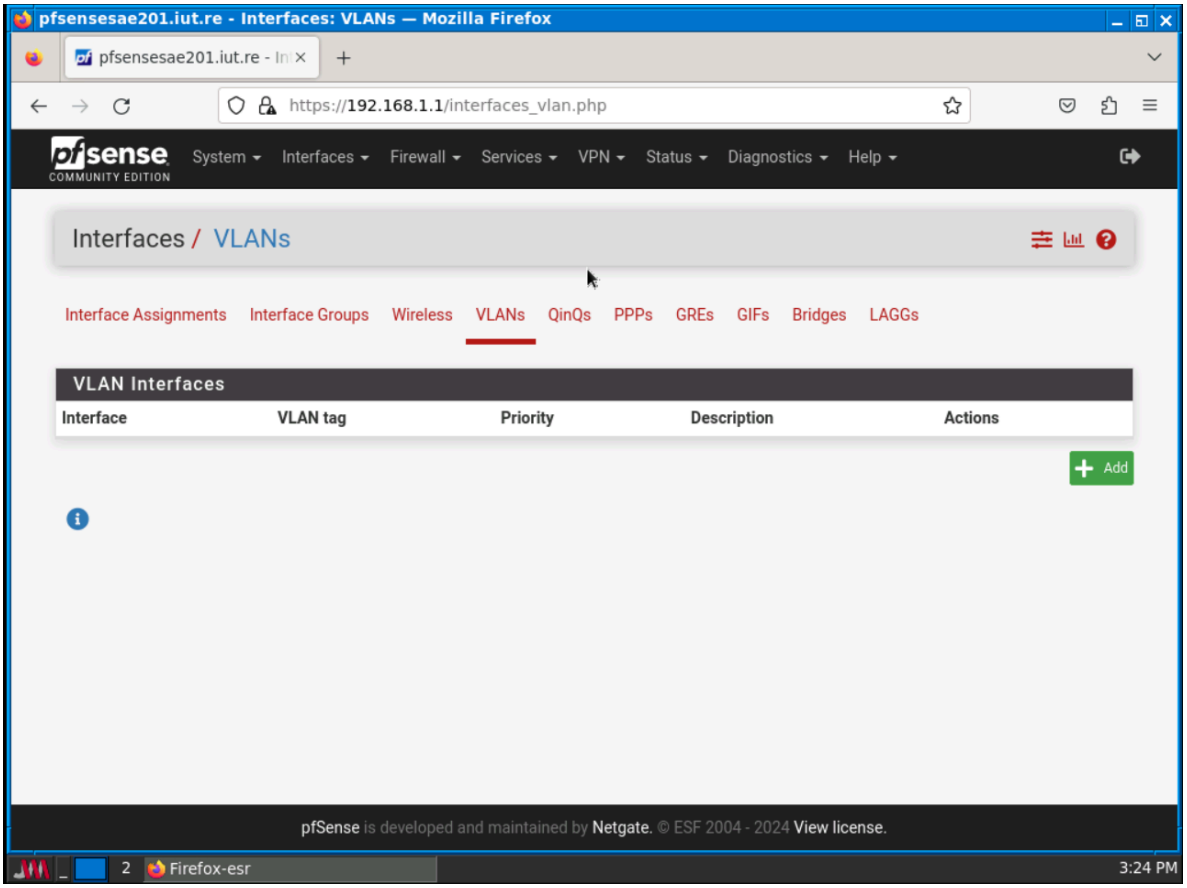


Voici donc la configuration du switch de niveau 2 :

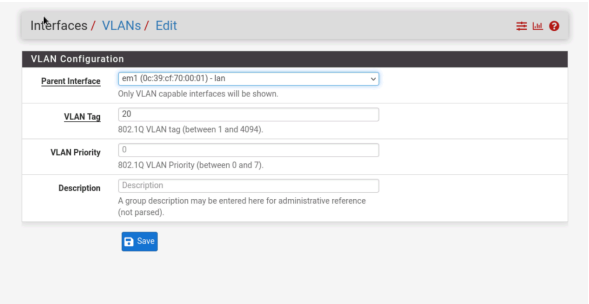
```
enable
configure terminal
vlan 10
name DIRECTION
exit
vlan 20
name GESTION
exit
vlan 30
name COMTECH
exit

interface range G1/0-3
switchport mode access
switchport access vlan 10
exit

interface range G2/0-3
switchport mode access
switchport access vlan 20
exit

interface range G3/0-3
switchport mode access
```

```
switchport access vlan 30
exit

interface G0/0
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 10,20,30
exit
```

On peut donc configurer pfsense (vlan) en branchant un pc directement sur le pfsense :

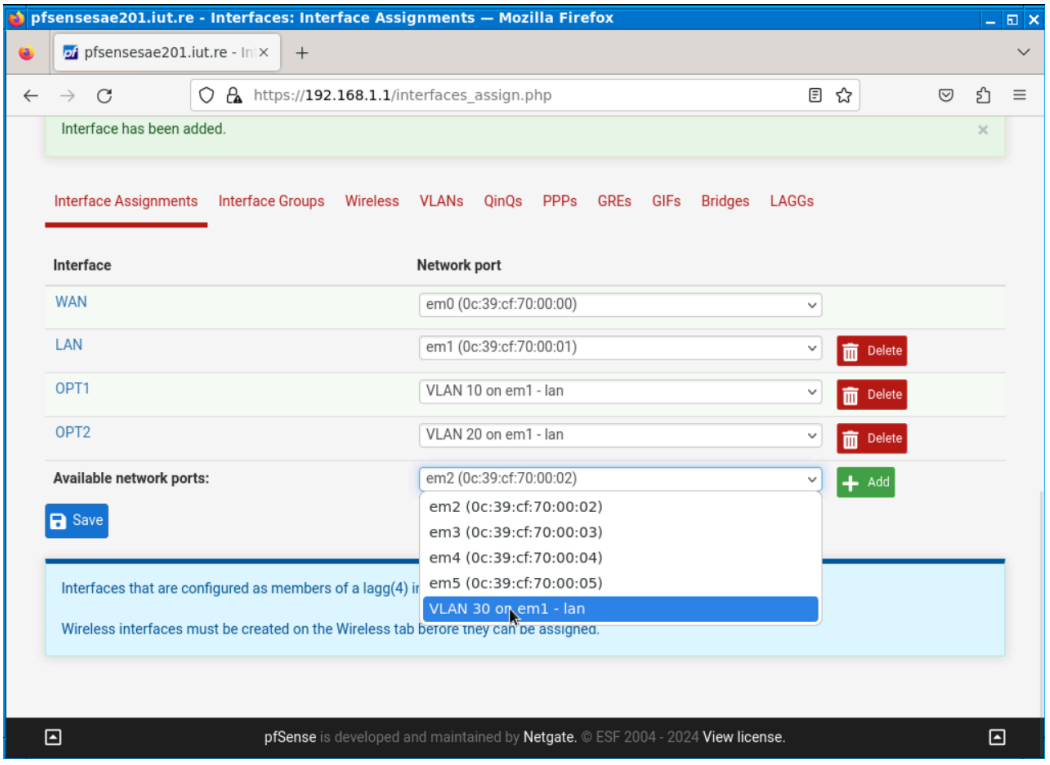


On peut donc ajouter les VLAN dans pfsense, avec comme exemple ci-dessous le VLAN 20 :

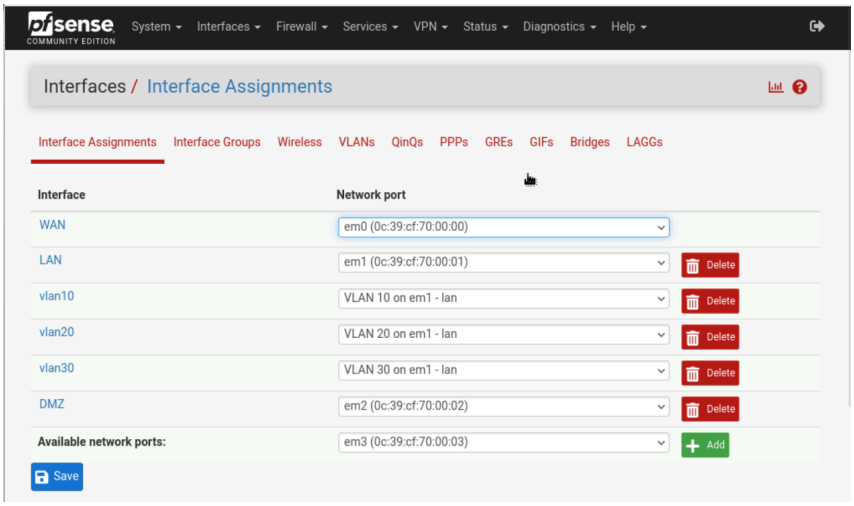


On ajoute ensuite les interfaces, chose faite pour tous les VLAN ainsi que la DMZ, voici un petit avant-après pour des configurations :

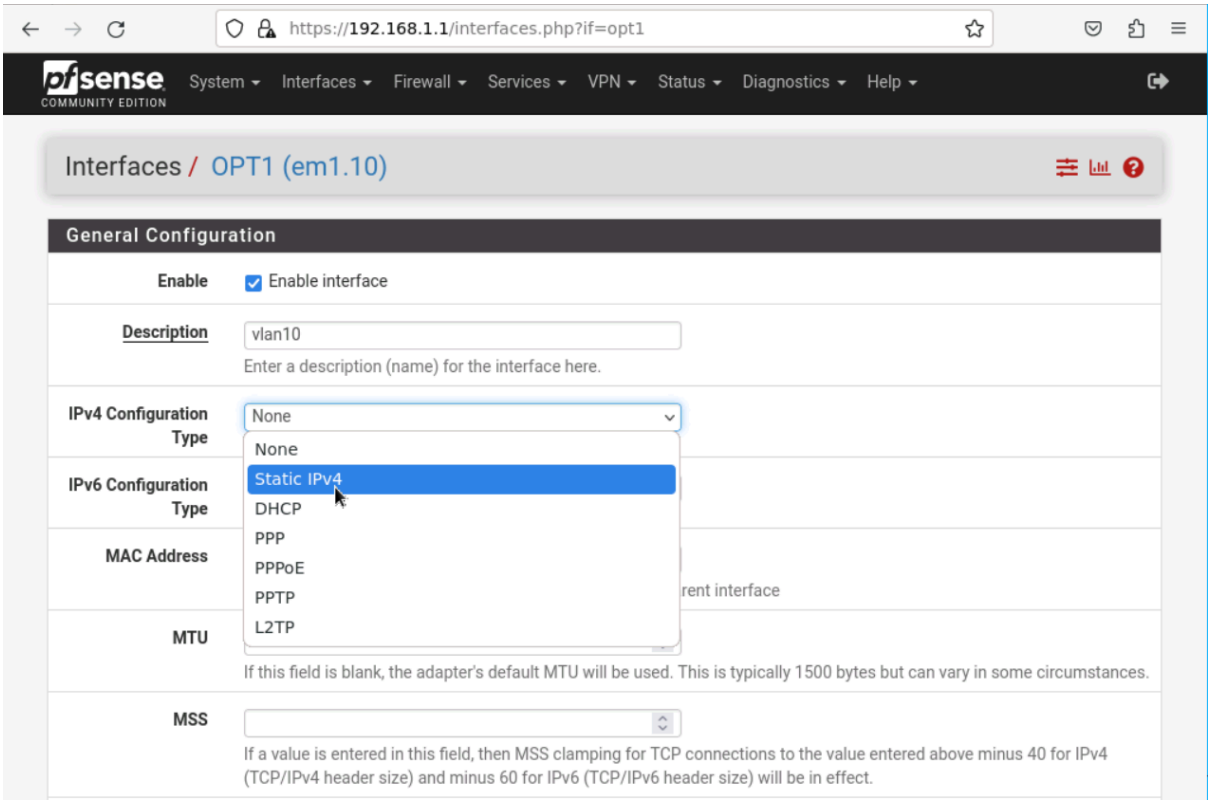
AVANT

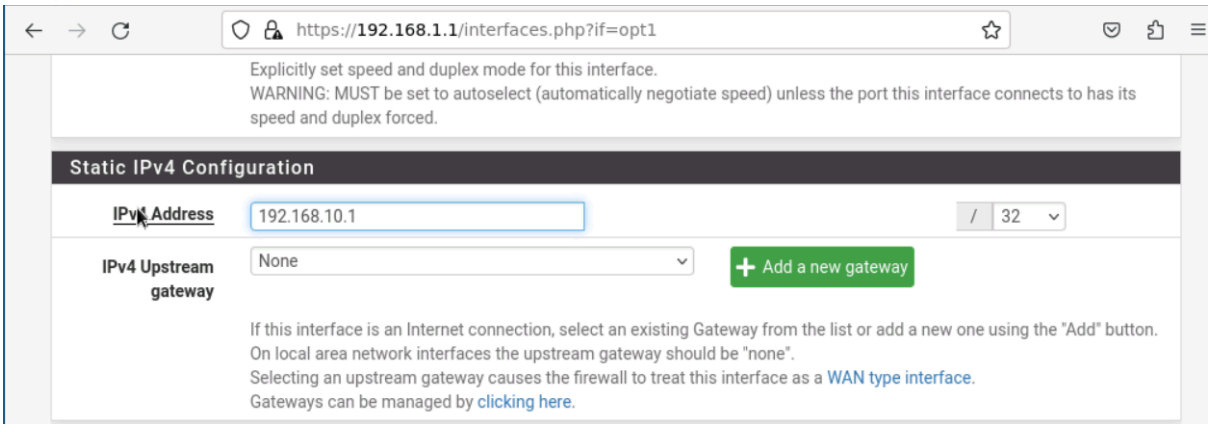


APRÈS

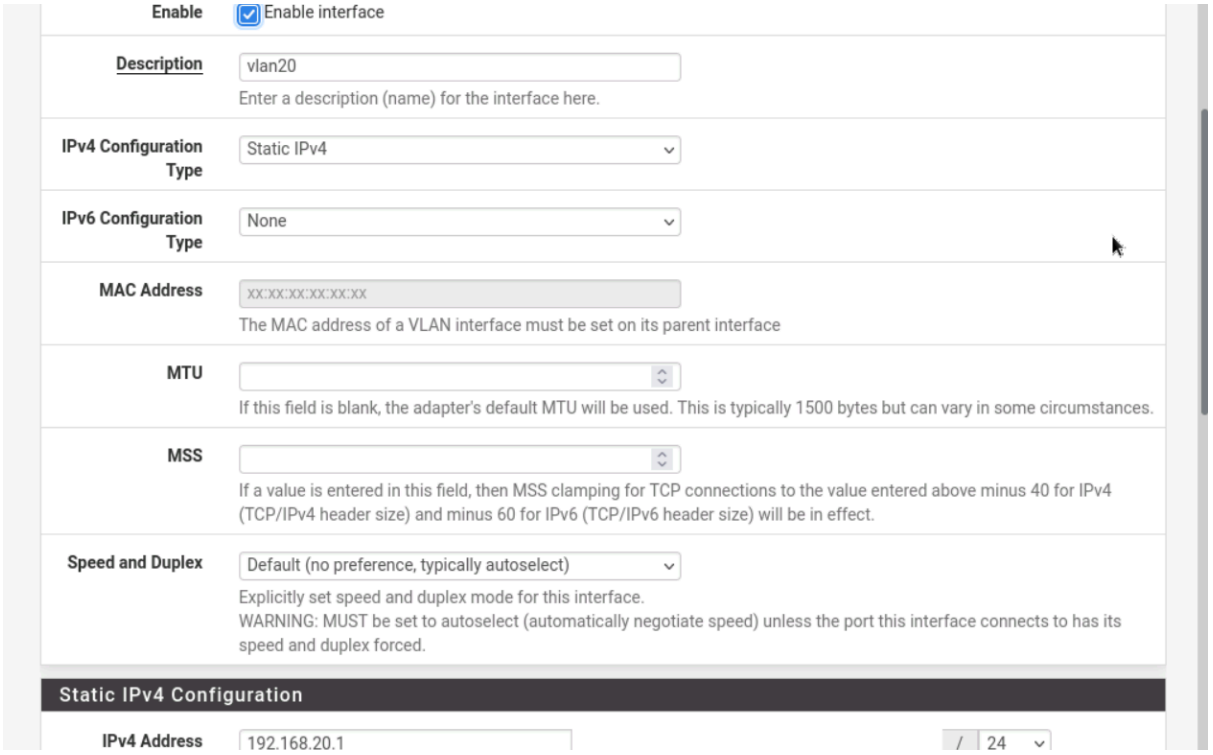


On peut alors adresser les interfaces pour les trois VLAN, en commençant par le VLAN 10 :

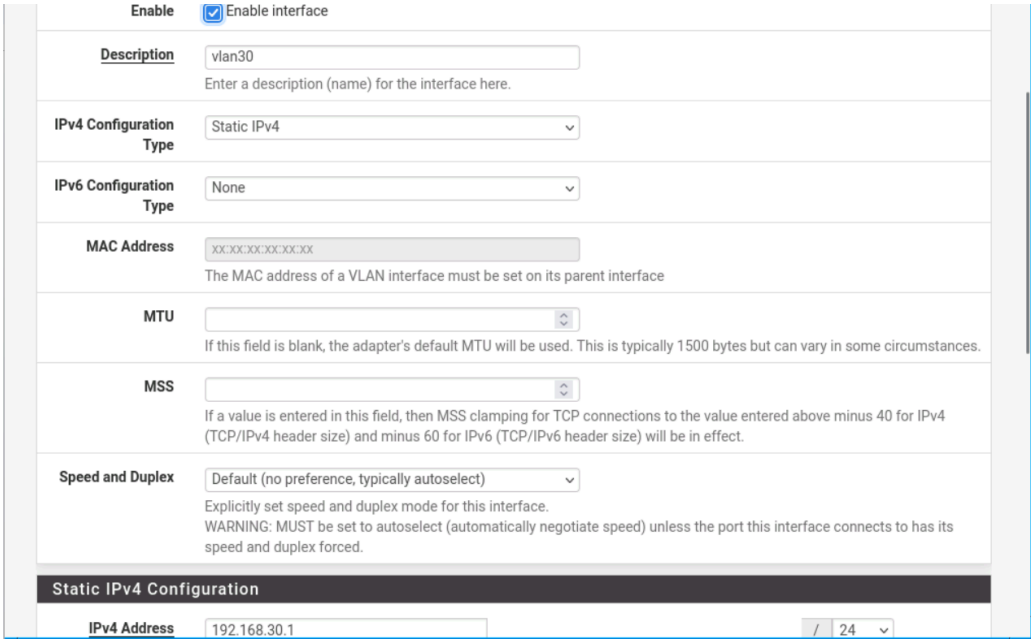




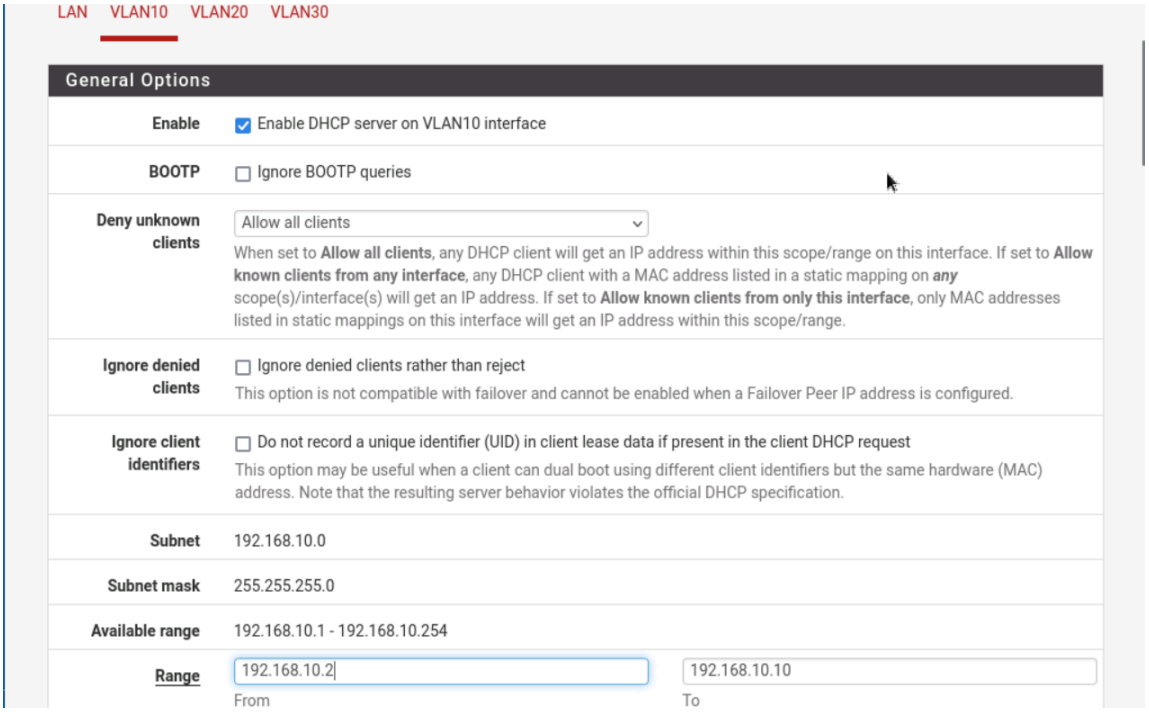
Puis le VLAN 20 :



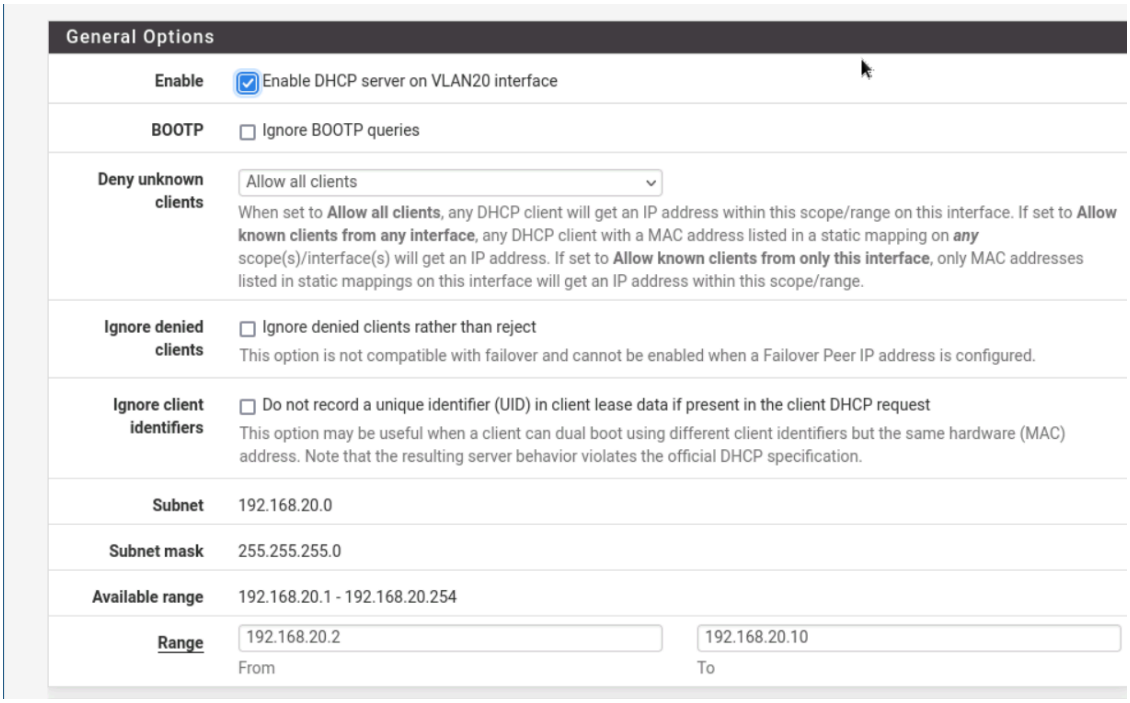
Puis le VLAN 30 :



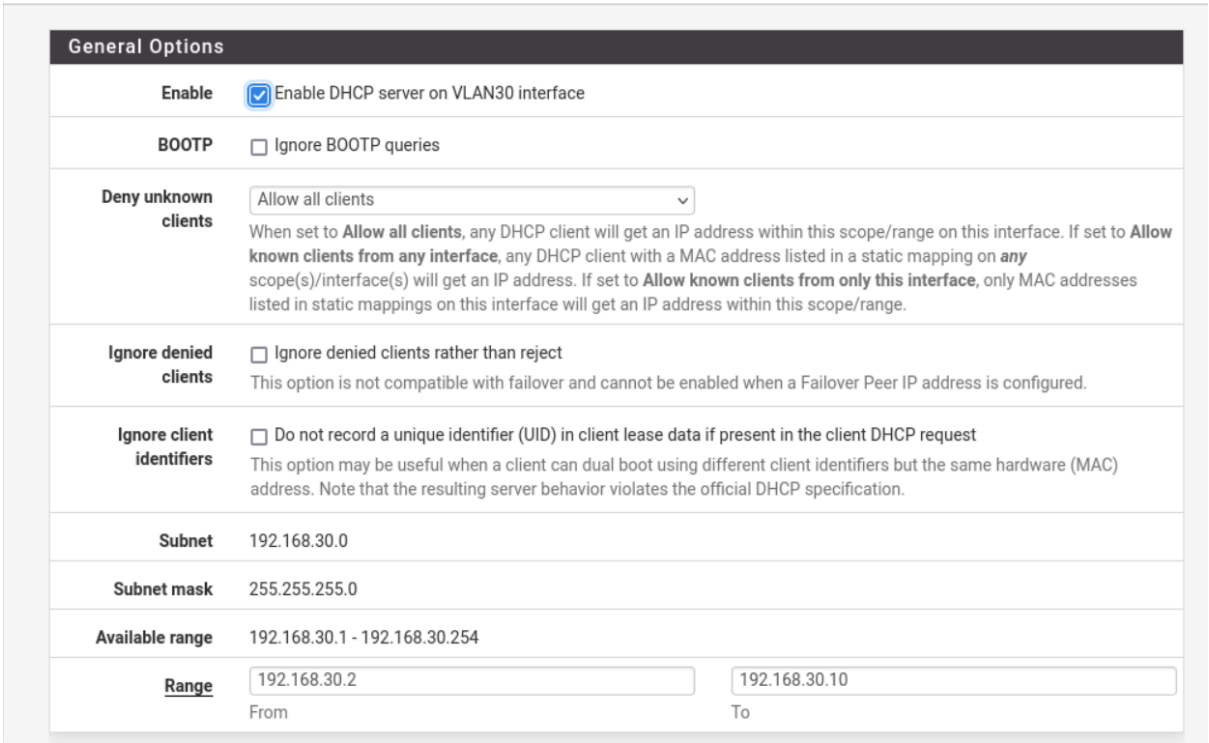
Puis on configure DHCP pour les trois VLAN en commençant par le VLAN 10 :



Puis le VLAN 20 :



Puis le VLAN 30 :



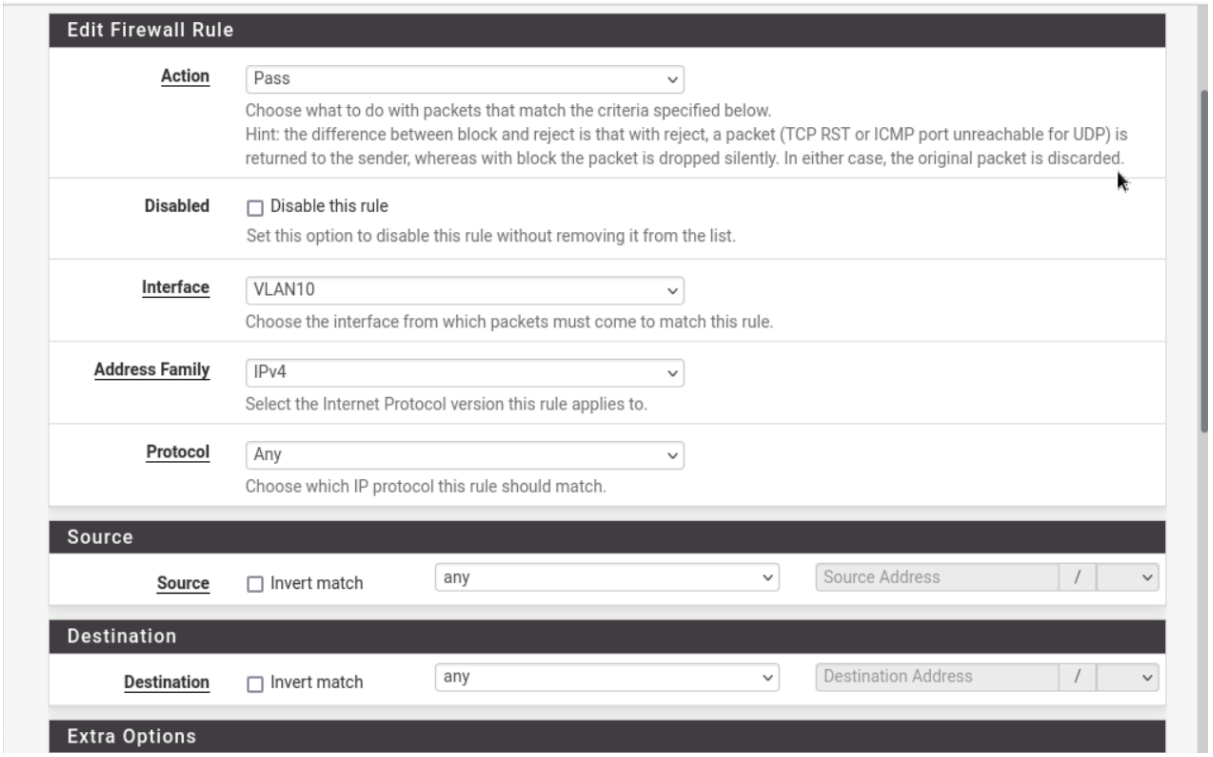
Et le DHCP est bien fonctionnel sur les PCs :

```
PC1> ip dhcp
DDORA IP 192.168.10.2/24 GW 192.168.10.1

PC1>
```

Et on peut tester la connectivité inter-VLAN :

```
PC2> ip dhcp
DDORA IP 192.168.20.2/24 GW 192.168.20.1

PC2> ping 192.168.10.2
192.168.10.2 icmp_seq=1 timeout
192.168.10.2 icmp_seq=2 timeout
84 bytes from 192.168.10.2 icmp_seq=3 ttl=63 time=4.263 ms
84 bytes from 192.168.10.2 icmp_seq=4 ttl=63 time=4.317 ms
84 bytes from 192.168.10.2 icmp_seq=5 ttl=63 time=3.750 ms
```

On fait configure à présent les règles pour les VLAN 10, 20 et 30 sur le pare feu (avec ici l'exemple du 10) :

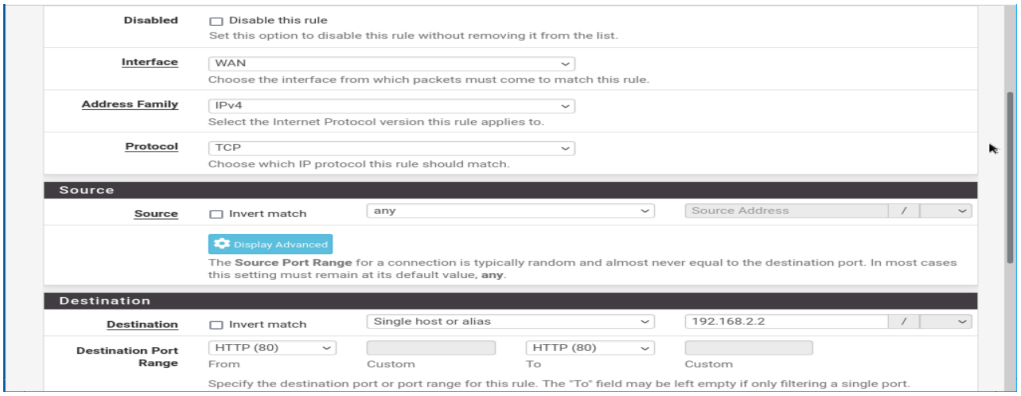


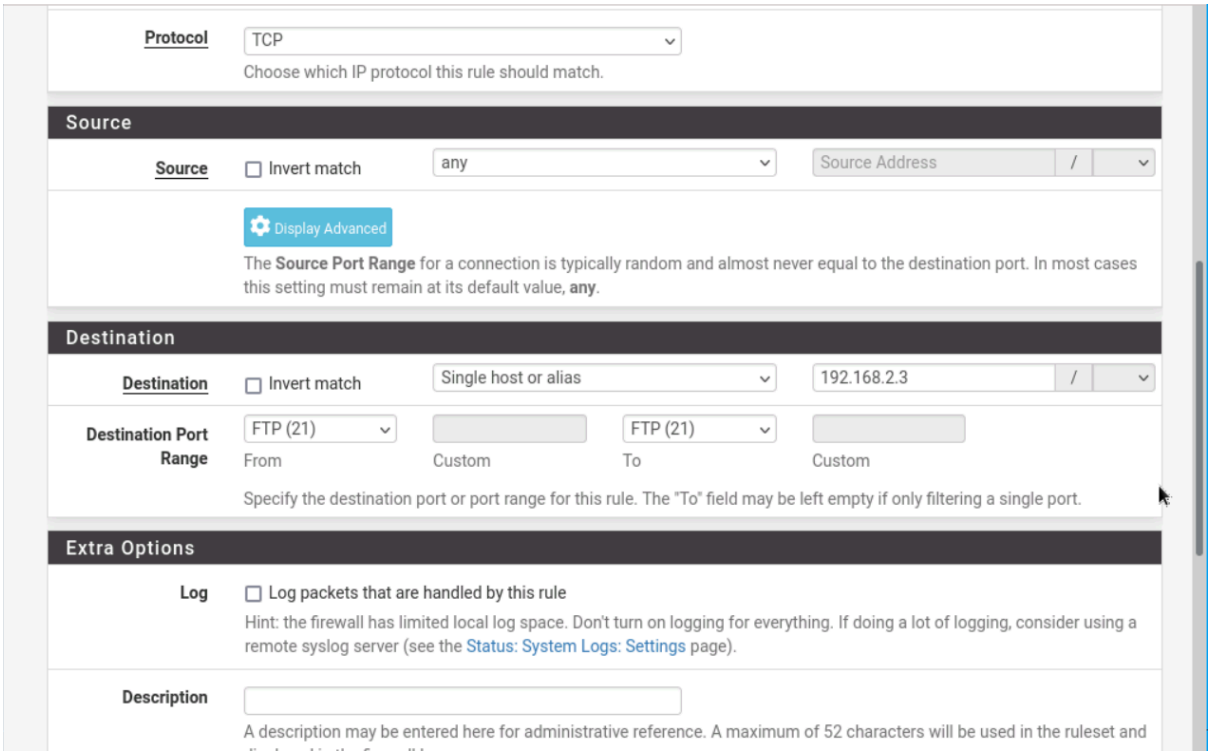
Puis on configure les règle du WAN pour qu'il accède à la DMZ :

```
[2.7.0-RELEASE][root@pfsensesae201.iut.re]/root: pfSsh.php playback enableallowa
llwan
Adding allow all rule...
Turning off block private networks (if on)...
Turning off block bogon networks (if on)...
Reloading the filter configuration...
```

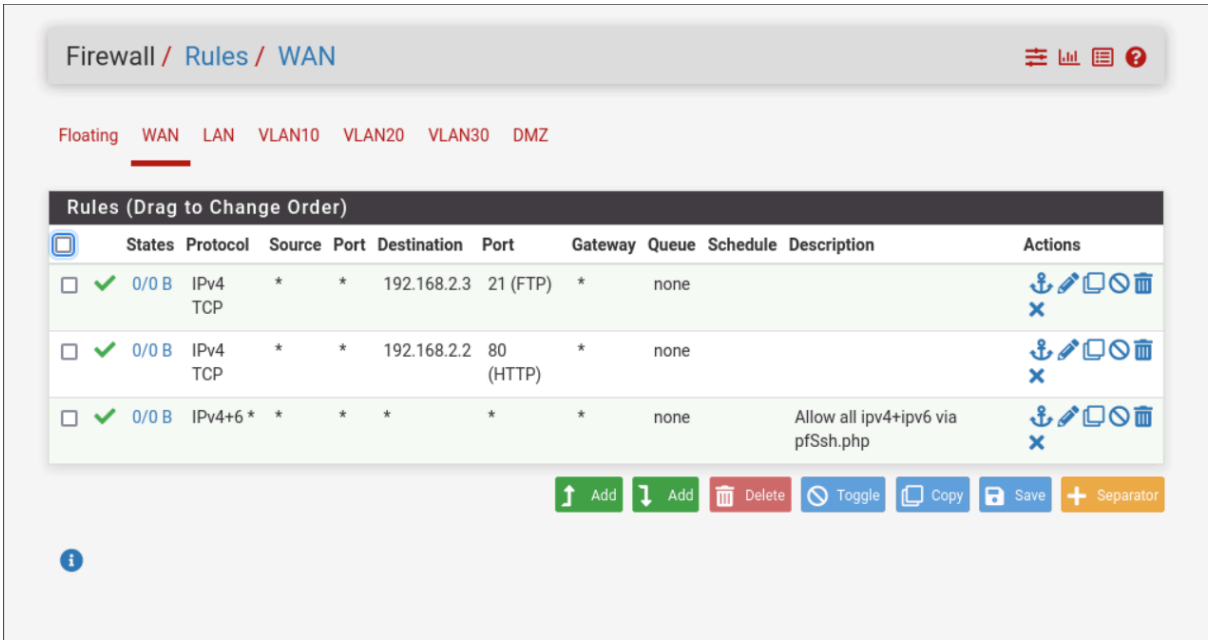
Et brièvement, voici les configurations pour chacun de nos services :

Voici les configurations pour notre service FTP :

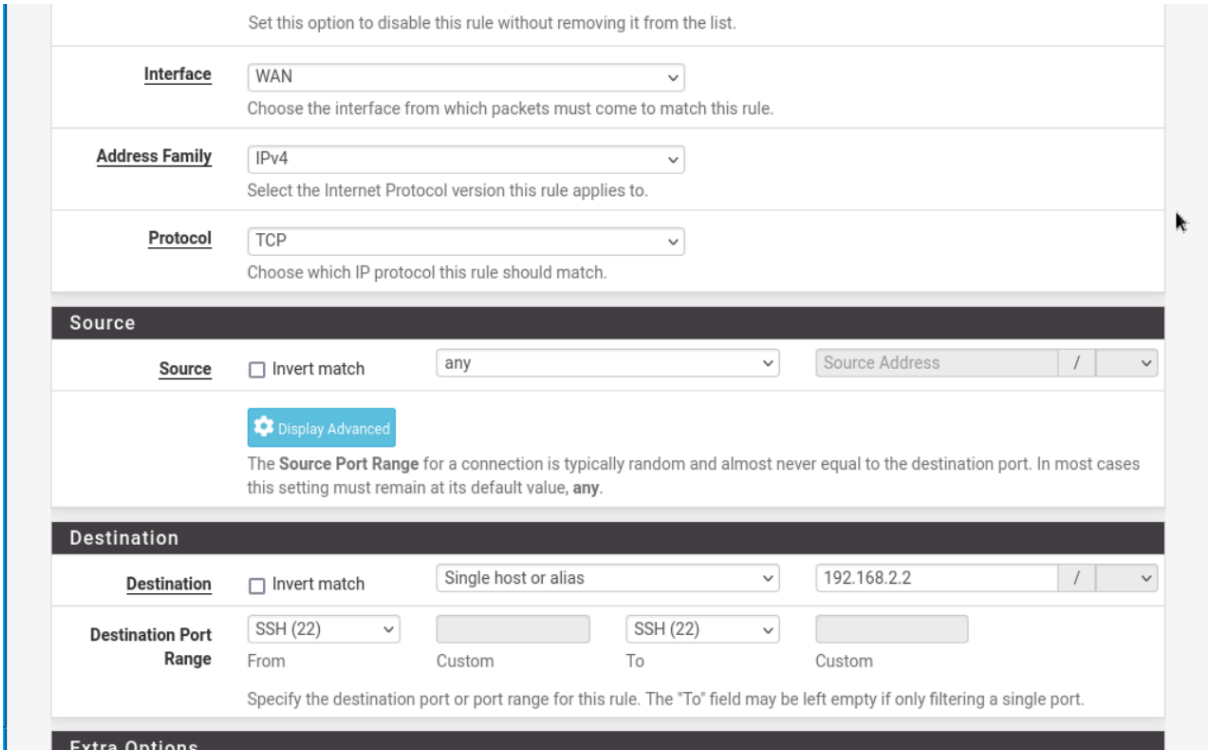




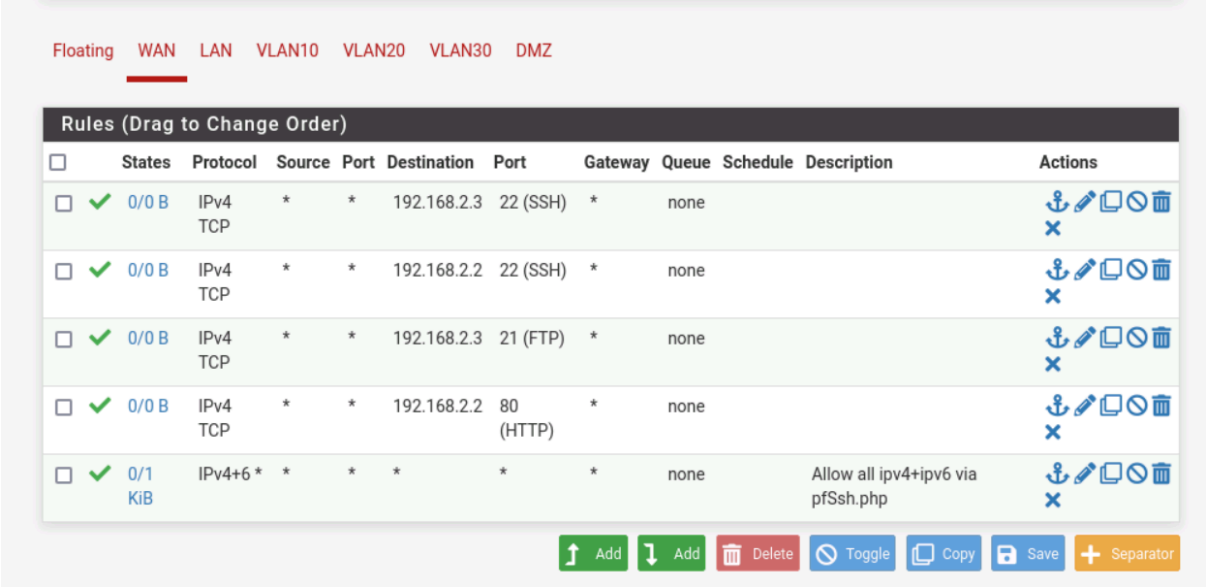
Et voici donc les configurations :



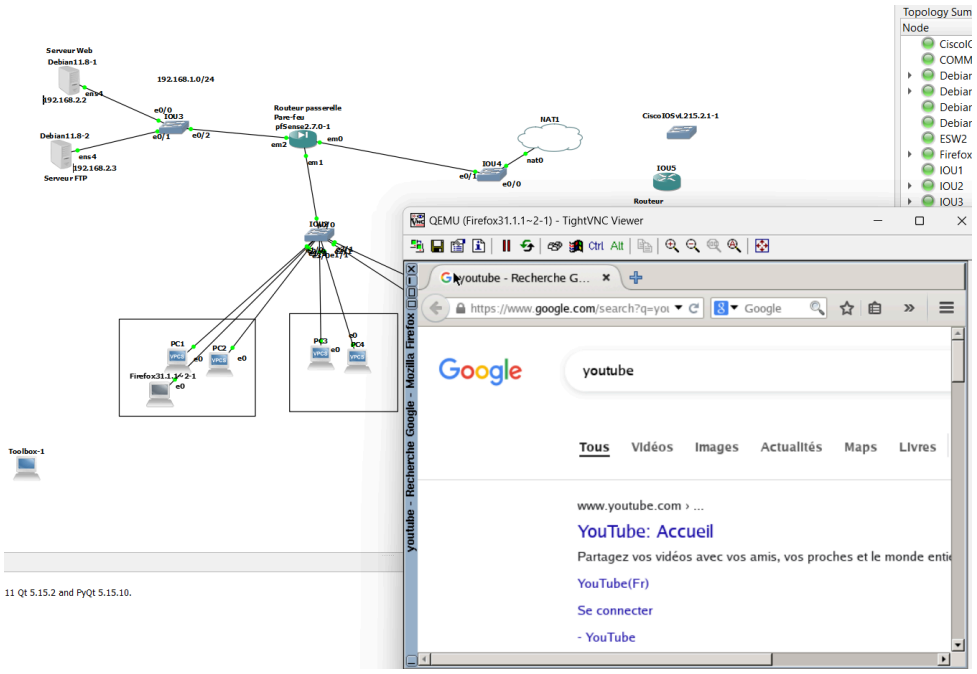
Et voici les règles SSH pour les différents serveur en commençant par le serveur HTTP :



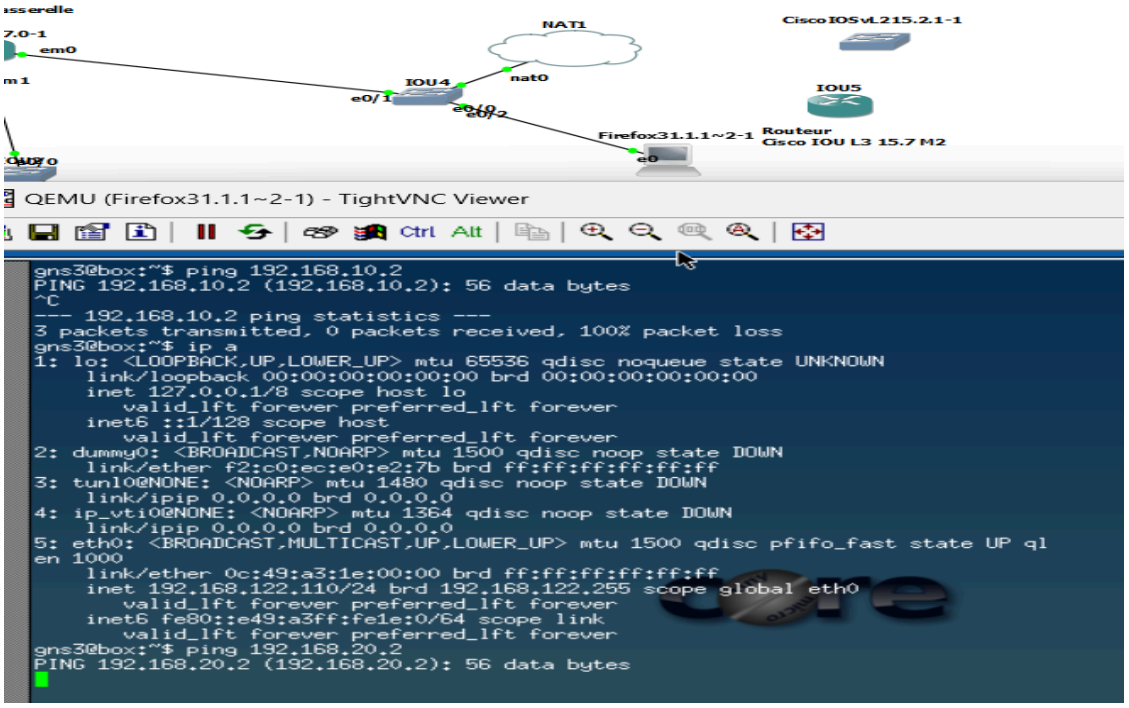
Puis le serveur FTP :



On peut alors effectuer les tests pour un PC du VLAN :

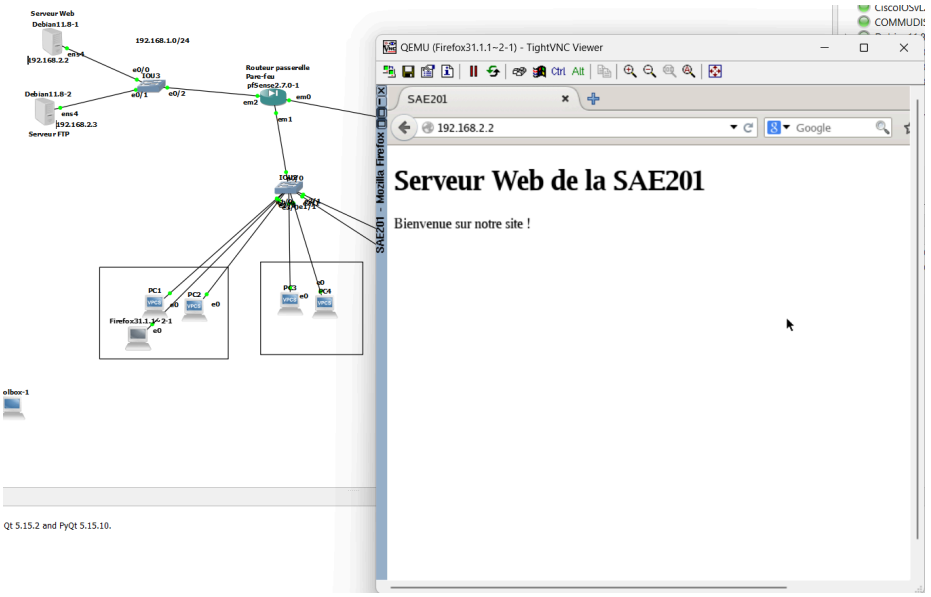


Et voici un test de connectivité du WAN vers le LAN :



Un échec logique et espéré car le WAN ne doit pas accéder au LAN !

On peut à présent vérifier qu'un utilisateur interne à l'entreprise peut accéder aux services réseaux installés soit Web et FTP (au moyen d'un objet webterm ou Firefox à la place d'un VPCS : Virtual PC Simulator) :

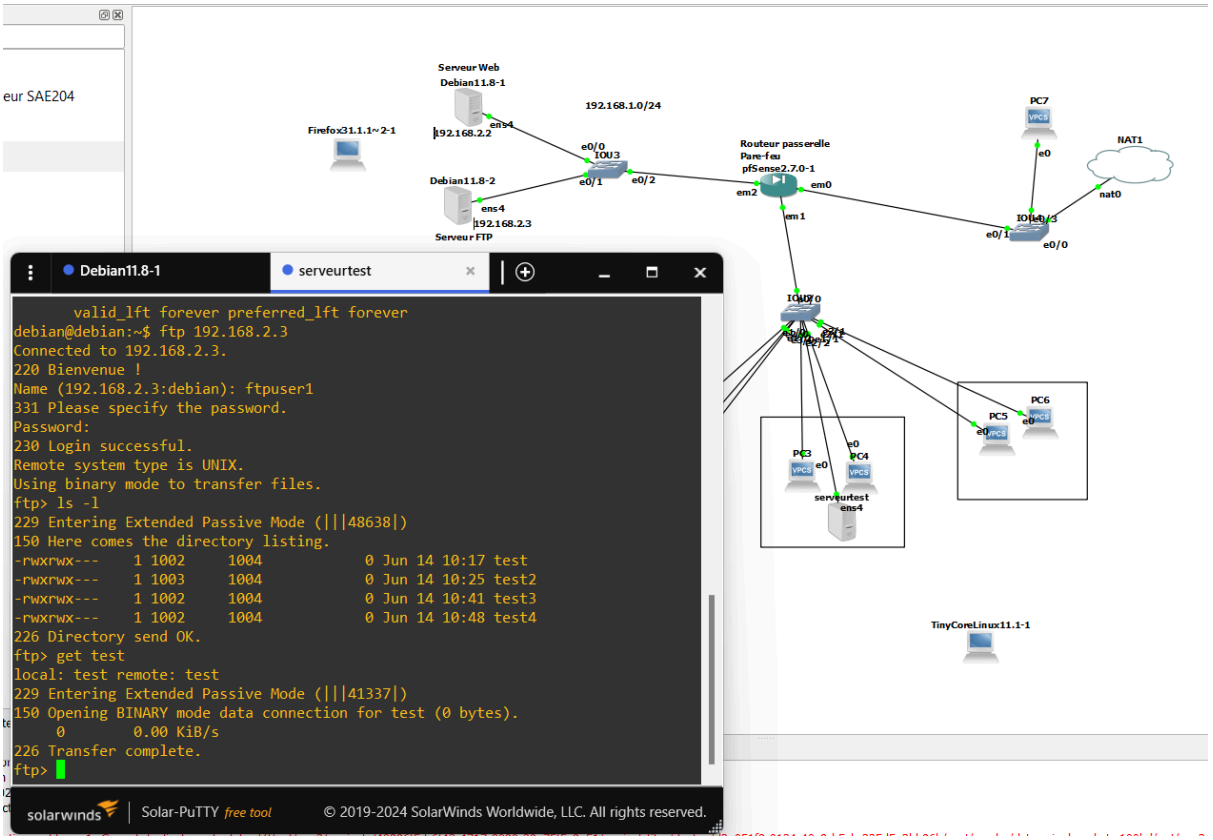


On peut alors installer ftp sur une machine du VLAN 20 qui peut donc accéder à nos services FTP :

```
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 0c:b9:55:a5:00:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 192.168.20.3/24 brd 192.168.20.255 scope global dynamic ens4
       valid_lft 7198sec preferred_lft 7198sec
    inet6 fe80::eb9:55ff:fea5:0/64 scope link
       valid_lft forever preferred_lft forever
debian@debian:~$ sudo apt install ftp
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  tnftp
The following NEW packages will be installed:
  ftp tnftp
0 upgraded, 2 newly installed, 0 to remove and 277 not upgraded.
Need to get 166 kB of archives.
After this operation, 319 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
oup default qlen 1000
    link/ether 0c:b9:55:a5:00:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 192.168.20.3/24 brd 192.168.20.255 scope global dynamic ens4
       valid_lft 7087sec preferred_lft 7087sec
    inet6 fe80::eb9:55ff:fea5:0/64 scope link
       valid_lft forever preferred_lft forever
debian@debian:~$ ftp 192.168.2.3
Connected to 192.168.2.3.
220 Bienvenue !
Name (192.168.2.3:debian): ftpuser1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -l
229 Entering Extended Passive Mode (|||48638|)
150 Here comes the directory listing.
-rwxrwx---    1 1002     1004            0 Jun 14 10:17 test
-rwxrwx---    1 1003     1004            0 Jun 14 10:25 test2
-rwxrwx---    1 1002     1004            0 Jun 14 10:41 test3
-rwxrwx---    1 1002     1004            0 Jun 14 10:48 test4
226 Directory send OK.
ftp>
```

Et voici un test de transfert de fichier pour tenter de récupérer le fichier test :



On a donc un FTP entièrement fonctionnel…

On peut également utiliser SSH depuis le serveur test en root pour gérer le serveur HTTP (serveur web apache2) :