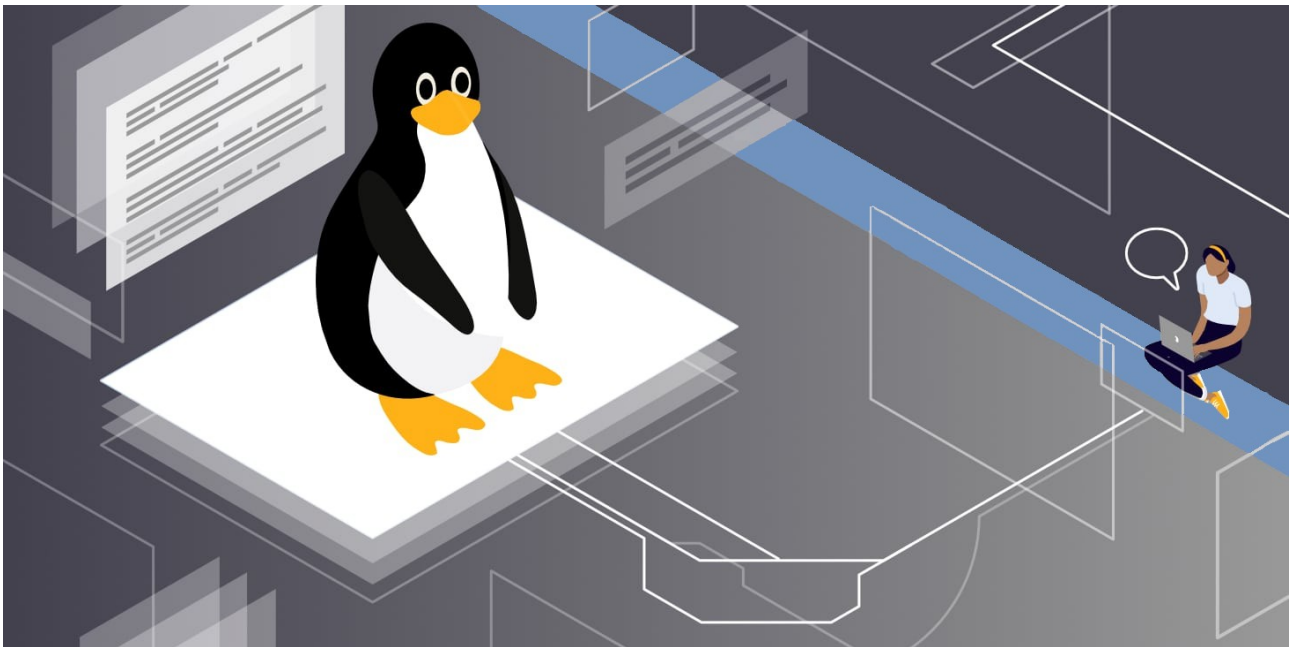
Fournitures de services sous Debian



SOMMAIRE

1. Contexte et cahier des charges

Notre client souhaite avoir accès à plusieurs services sous Debian. Il demande à ce qu'il y ait 2 réseaux différents, un pour les utilisateurs du secrétariat et l'autre le poste serveur ainsi que le service informatique.

Il désire également que le réseau secrétariat ait un adressage automatique des adresses IP et que le réseau soit performant et sécurisé.

Lorsque nous prenons le projet en main, nous installons Debian sur le poste serveur ainsi que les paquets suivants : Apache2, Bind9, OpenSSL, proFTP, isc-dhcp-serveur, dnsutils.

Nous utiliserons un switch de niveau 2 Cisco 3560 et le logiciel Putty pour se connecter en SSH.

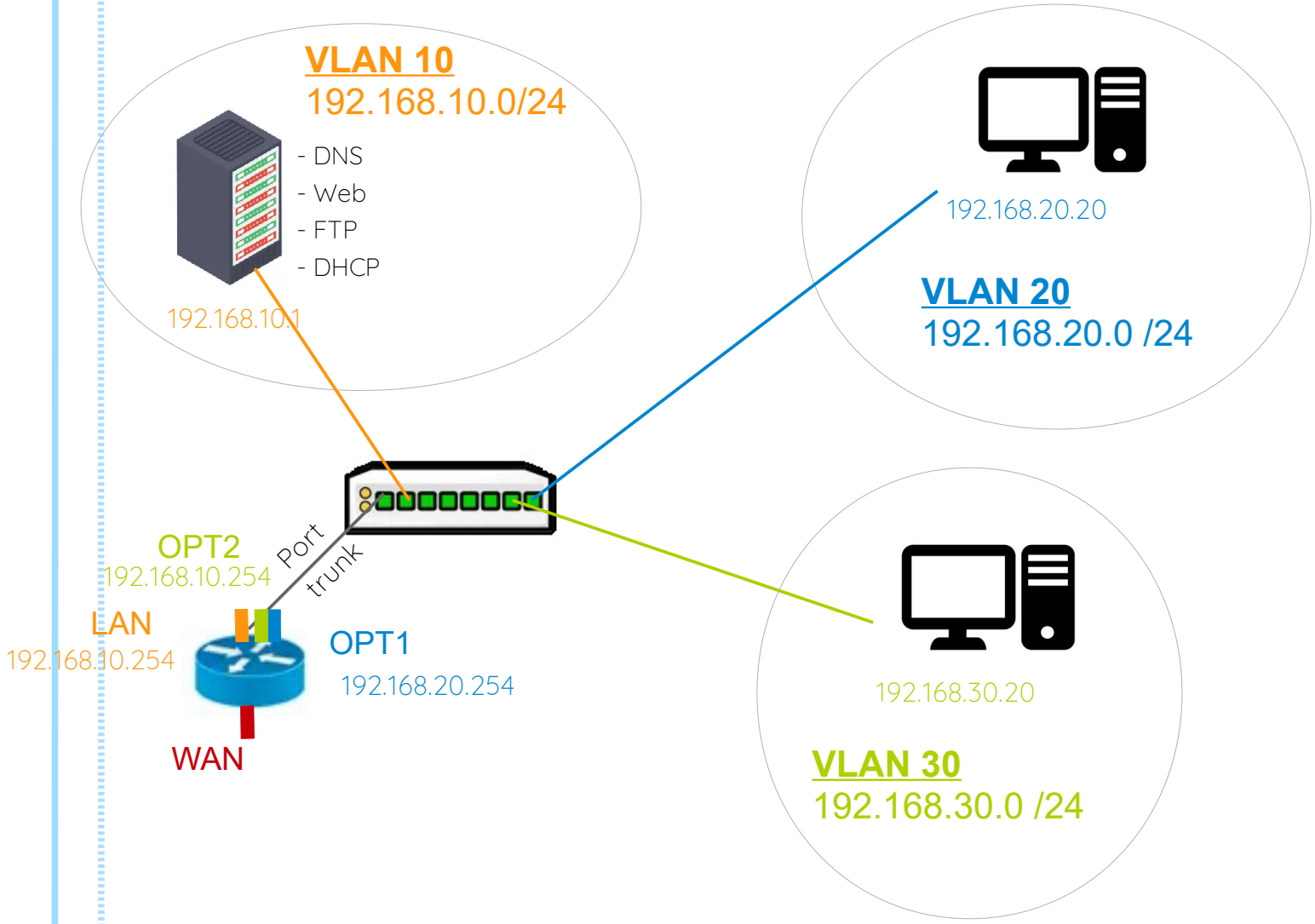
Nous nous servirons de PFsense, un parefeu et routeur virtuel en guise de routeur dans notre démonstration.

Pour que le réseau soit performant et sécurisé, nous créerons 3 VLANs.

1.1. Objectifs

- le poste serveur héberge les serveurs FTP, DNS, Web
- mettre en place et configurer un switch
- mettre en place et configurer un routeur virtuel
- introduire des VLAN et que ces derniers puissent communiquer entre eux
- accès au serveur FTP seulement par le poste administrateur se trouvant dans le réseau du service informatique et poste serveur
- mettre en place des tests de validation répondant aux demandes

1.2. Architecture



2. Mise en œuvre

2.1. Poste serveur

2.1.1. Configuration IP

2 cartes réseaux : une pour aller vers internet (le temps d'installer et configurer les différents services. Elle pourra être retirée ensuite) et l'autre pour aller sur le réseau service informatique.

Paramétrage des 2 cartes réseaux :

- Nous modifions le fichier *interfaces* pour entrer les adresses ip statiques avec la commande nano /etc/network/interfaces

```
  GNU nano 3.2                                        /etc/network/interfaces

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

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
        address 192.168.10.1
        netmask 255.255.255.0
        gateway 192.168.10.254
```

- Redémarrage du service avec la commande

```
root@restructurationvlan:~# systemctl restart networking.service
root@restructurationvlan:~#
```

- Et vérification

```
root@restructurationvlan:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:38:01:10 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.1/24 brd 192.168.10.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe38:110/64 scope link
       valid_lft forever preferred_lft forever
```

→ les adresses IP se sont bien mises sur chaque carte réseau.

- Vérification ping vers réseau utilisateur

```
root@restructurationvlan:/etc# ping 192.168.1.20
PING 192.168.1.20 (192.168.1.20) 56(84) bytes of data.
64 bytes from 192.168.1.20: icmp_seq=1 ttl=128 time=0.590 ms
64 bytes from 192.168.1.20: icmp_seq=2 ttl=128 time=0.516 ms
64 bytes from 192.168.1.20: icmp_seq=3 ttl=128 time=1.28 ms
^C
--- 192.168.1.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 55ms
rtt min/avg/max/mdev = 0.516/0.793/1.275/0.343 ms
```

→ ok

2.1.2. Paramétrage Apache 2

Nous avons au préalable téléchargé le paquet Apache2 et vérifié son bon fonctionnement.

Nous allons donc à présent mettre en place le serveur web composé d'un site web.

- Dans le dossier html nous copions le fichier *index.html* avec la commande mv "nom du fichier" "nouveau nom du fichier" pour avoir un fichier de secours si besoin.

- Création du dossier portant le nom de notre site web, ici "esicad".

Vérification : existence du dossier test et du fichier *index.html.old*

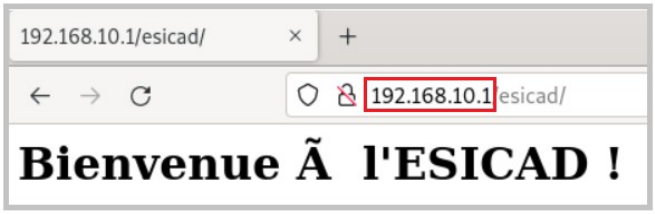
```
root@restructurationvlan:/var/www/html# ls -l
total 20
drwxr-xr-x 2 root root  4096 juin   9 19:38 esicad
-rw-r--r-- 1 root root 10701 sept. 17  2023 index.html.old
```

→ ok

- Création du fichier *index.html* dans le dossier test avec la commande nano "nom du fichier"

```
 GNU nano 3.2                                              index.html

<html>
<body>
<h1> Bienvenue à l'ESICAD ! </h1>
</body>
</html>
```

- Vérification de l'accès au site avec l'adresse IP du serveur à partir du navigateur Firefox.



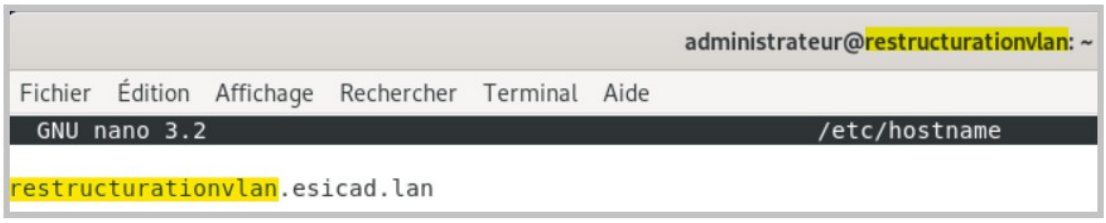
→ ok

2.1.3. Configuration du serveur DNS

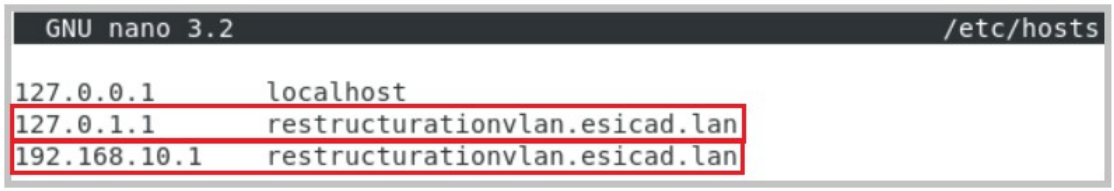
Le DNS (Domain Name System) a pour fonction principale la traduction des noms de domaines en adresses IP. Il fait correspondre un nom de domaine avec une adresse IP que ce soit pour un site internet mais aussi pour des ordinateurs internes au réseau local.

Pour notre projet, Bind9 et dnsutils ont été installés au préalable sur le poste

- Renseigner le nom FQDN du serveur, c'est à dire le 'nomduserveur.nomdusite' dans le fichier *hostname*. Pour cela, nous utilisons la commande nano /etc/hostname



- Puis, dans le fichiers *hosts*, nous entrons une ligne "localhost", une ligne avec le nom FQDN et l'ip localhost et enfin une ligne avec le nom FQDN et l'ip de notre serveur. Cela permet d'associer l'adresse IPV4 du serveur au nom FQDN.



- On indique le domaine et la zone de recherche. Cela permet au serveur d'être intégré à la zone DNS : nano /etc/resolv.conf

- Déclaration de la zone DNS "test.lan" et sa zone inverse pour que les adresses IP puissent être traduites en noms de domaines dans le dossier bind : nano /etc/bind/named.conf.local

```
  GNU nano 3.2                                          named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "esicad.lan" {
        type master;
        file "/etc/bind/db.esicad.lan";
        notify yes;
};

zone "10.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.10.168.192.in-addr.arpa";
};
```

- Création des zones :
  - dans le dossier bind : cp "db.local" "db.esicad.lan" et nano "db.10.168.192.in-addr.arpa"
  - vérification de l'existence des 2 fichiers :

```
root@restructurationvlan:/etc/bind# ls -l
total 56
-rw-r--r-- 1 root root 2761 juil.  7  2023 bind.keys
-rw-r--r-- 1 root root  237 juil.  7  2023 db.0
-rw-r--r-- 1 root bind  218 nov.  28  2023 db.10.168.192.in-addr.arpa
-rw-r--r-- 1 root root  271 juil.  7  2023 db.127
-rw-r--r-- 1 root root  237 juil.  7  2023 db.255
-rw-r--r-- 1 root root  353 juil.  7  2023 db.empty
-rw-r--r-- 1 root bind  346 juin   9 18:42 db.esicad.lan
-rw-r--r-- 1 root root  270 juil.  7  2023 db.local
-rw-r--r-- 1 root bind  463 juil.  7  2023 named.conf
-rw-r--r-- 1 root bind  498 juil.  7  2023 named.conf.default-zones
-rw-r--r-- 1 root bind  346 nov.  28  2023 named.conf.local
-rw-r--r-- 1 root bind  866 sept. 26  2023 named.conf.options
-rw-r----- 1 bind bind   77 sept. 17  2023 rndc.key
-rw-r--r-- 1 root root 1317 juil.  7  2023 zones.rfc1918
```

→ ok

- Déclaration de notre serveur (restructurationvlan.test.lan) et enregistrement DNS du poste serveur dans le fichier *db.test.lan*

```
 GNU nano 3.2                                          db.esicad.lan

;
; BIND data file for local loopback interface
;
$TTL    10800
$ORIGIN esicad.lan.
@       IN      SOA     restructurationvlan.esicad.lan. root.restructurationvlan.esicad.lan (
        20160505;
        3h;
        1h;
        1w;
        1h)
;
@       IN      NS      restructurationvlan.esicad.lan.
restructurationvlan IN          A       192.168.10.1
www     IN      A       192.168.10.1
localhost       IN      A       127.0.0.1
```

- **TTL** signifie "Time To Live", c'est le temps de rafraîchissement en secondes de l'entrée
- **SOA** signifie "Start Of Authority", partie essentielle du fichier de zone DNS, il indique la zone du domaine. 1 zone = 1 SOA
- **IN** signifie "Internet"
- **NS** signifie "Name Server", indique les serveurs de noms qui sont chargés du domaine.
- **A** signifie "Address". Il associe un nom d'hôte a une adresse IP

- Configuration des zones DNS inversées dans le fichier *db.10.168.192.in-addr.arpa*

```
 GNU nano 3.2                                  db.10.168.192.in-addr.arpa

$TTL 10800
$ORIGIN 10.168.192.in-addr.arpa.
@       IN SOA restructurationvlan.esicad.lan. root.esicad.lan. (
        20160505;
        3h;
        1h;
        1w;
        1h);
@       IN NS   restructurationvlan.esicad.lan.
1       IN PTR restructurationvlan.esicad.lan
```

- Vérification de la syntaxe des fichiers de configuration *named.conf*. L'article "-z" permet de trouver et vérifier les zones DNS.

```
root@restructurationvlan:/etc/bind# named-checkconf -z
zone test.lan/IN: loaded serial 20160505
zone 10.168.192.in-addr.arpa/IN: loaded serial 20160505
zone localhost/IN: loaded serial 2
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1
```

→ ok

Après redémarrage de Bind9, nous testons le bon fonctionnement de notre serveur :

- Recherche sur la zone directe

```
root@restructurationvlan:/etc/bind# dig restructurationvlan.esicad.lan

; <<>> DiG 9.11.5-P4-5.1+deb10u9-Debian <<>> restructurationvlan.esicad.lan
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51423
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 94ce26ae0fb1fab4f93f75d66665dbd861908967209e233a (good)
;; QUESTION SECTION:
;restructurationvlan.esicad.lan.        IN      A

;; ANSWER SECTION:
restructurationvlan.esicad.lan. 10800 IN A      192.168.10.1

;; AUTHORITY SECTION:
esicad.lan.             10800   IN      NS      restructurationvlan.esicad.lan.

;; Query time: 0 msec
;; SERVER: 192.168.10.1#53(192.168.10.1)
```

- Recherche sur la zone inversée

```
root@restructurationvlan:/etc/bind# dig -x 192.168.10.1

; <<>> DiG 9.11.5-P4-5.1+deb10u9-Debian <<>> -x 192.168.10.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31208
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7af6f0fa3a305166656d277e6665dc053c04081d763ee177 (good)
;; QUESTION SECTION:
;1.10.168.192.in-addr.arpa.     IN      PTR

;; ANSWER SECTION:
1.10.168.192.in-addr.arpa. 10800 IN     PTR     restructurationvlan.esicad.lan.10.168.192.in-addr.arpa.

;; AUTHORITY SECTION:
10.168.192.in-addr.arpa. 10800  IN      NS      restructurationvlan.esicad.lan.

;; ADDITIONAL SECTION:
restructurationvlan.esicad.lan. 10800 IN A      192.168.10.1
```

→ les lignes encadrées reprennent bien les informations configurées dans les 2 fichiers.

Vérification du bon fonctionnement de notre serveur DNS :

- Ping vers notre site internet (DNS)

```
root@restructurationvlan:/etc/bind# ping www.esicad.lan
PING www.esicad.lan (192.168.10.1) 56(84) bytes of data.
64 bytes from restructurationvlan.esicad.lan (192.168.10.1): icmp_seq=1 ttl=64 time=0.043 ms
64 bytes from restructurationvlan.esicad.lan (192.168.10.1): icmp_seq=2 ttl=64 time=0.152 ms
64 bytes from restructurationvlan.esicad.lan (192.168.10.1): icmp_seq=3 ttl=64 time=0.128 ms
64 bytes from restructurationvlan.esicad.lan (192.168.10.1): icmp_seq=4 ttl=64 time=0.134 ms
64 bytes from restructurationvlan.esicad.lan (192.168.10.1): icmp_seq=5 ttl=64 time=0.120 ms
64 bytes from restructurationvlan.esicad.lan (192.168.10.1): icmp_seq=6 ttl=64 time=0.135 ms
^C
--- www.esicad.lan ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 112ms
rtt min/avg/max/mdev = 0.043/0.118/0.152/0.037 ms
```

→ le serveur DNS a été correctement installé et paramétré.

2.1.4. Serveur FTP

Le protocole FTP est prévu pour pouvoir réaliser des uploads et des downloads, c'est a dire pouvoir transférer des fichiers vers un serveur ou pour les télécharger par exemple.

Dans notre projet, seul le poste administrateur a le droit d'y accéder et il a tous les droits.

- Installation de proFTPD
- Vérification du statut

```
root@restructurationvlan:~# systemctl status proftpd.service
● proftpd.service - LSB: Starts ProFTPD daemon
   Loaded: loaded (/etc/init.d/proftpd; generated)
   Active: active (running) since Sun 2023-10-15 12:31:24 CEST; 5min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 890 ExecStart=/etc/init.d/proftpd start (code=exited, status=0/SUCCES
    Tasks: 1 (limit: 4689)
   Memory: 9.5M
   CGroup: /system.slice/proftpd.service
           └─912 proftpd: (accepting connections)
```

→ ok

- Configuration FTP dans /etc/proftpd/conf.d/ sur le fichier *ftp-perso.conf*. Nous avons créé ce fichier personnalisé au lieu de modifier directement le fichier *proftpd.conf* pour éviter que la configuration ne soit écrasée par une mise à jour proFTPD.

```
  GNU nano 3.2                                  ftp-perso.conf

#Nom du serveur
ServerName "restructurationvlan.esicad.lan"

#Message de connexion
DisplayLogin "La connexion au serveur FTP s'est bien effectuée"

#Désactiver IPV6
UseIPV6 off

#Spécifie le répertoire FTP auquel l'utilisateur est autorisé à accéder
DefaultRoot état

#Autoriser la connexion uniquement aux membres du groupe "ftpadmin"
<Limit LOGIN>
DenyGroup !ftpadmin
</Limit>
```

→ ok

- Création du groupe 'ftpadmin'

```
root@restructurationvlan:~# addgroup ftpadmin
Ajout du groupe « ftpadmin » (GID 1001)...
Fait.
root@restructurationvlan:~#
```

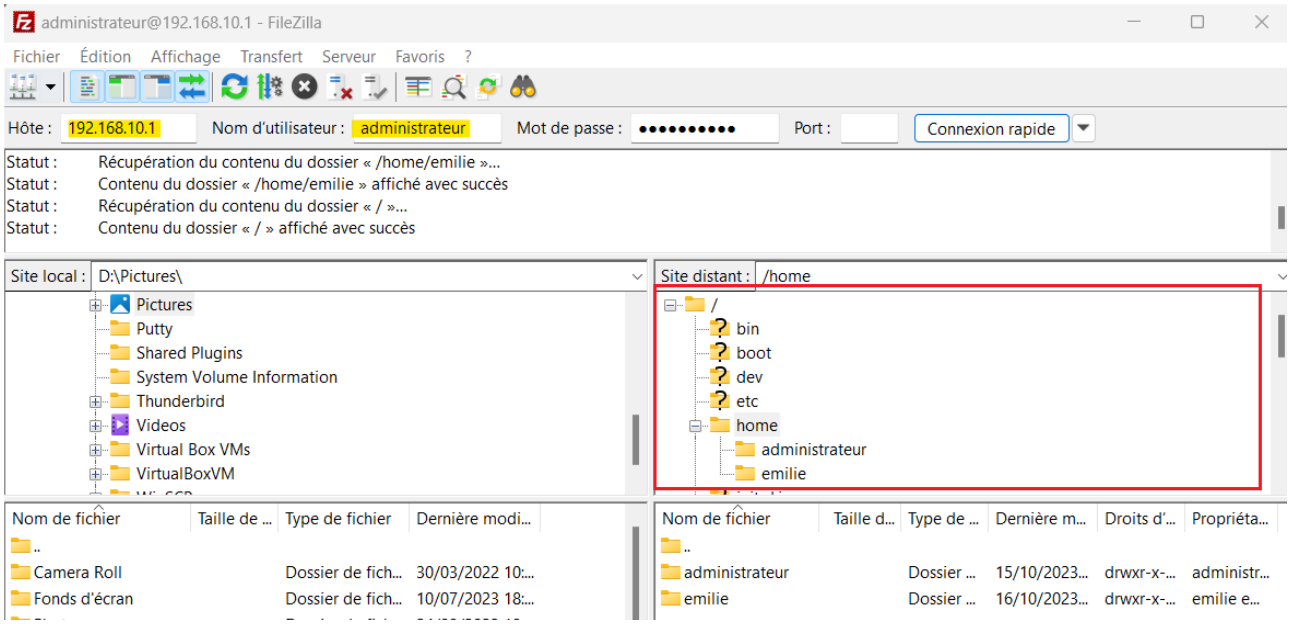
- Création de l'utilisateur 'administrateur'

```
root@restructurationvlan:~# adduser administrateur
Ajout de l'utilisateur « administrateur » ...
Ajout du nouveau groupe « administrateur » (1002) ...
Ajout du nouvel utilisateur « administrateur » (1001) avec le groupe « administrateur » ...
Création du répertoire personnel « /home/administrateur »...
Copie des fichiers depuis « /etc/skel »...
Nouveau mot de passe :
```

- Ajout de l'utilisateur au groupe 'ftpadmin'

```
root@restructurationvlan:~# adduser administrateur ftpadmin
Ajout de l'utilisateur « administrateur » au groupe « ftpadmin »...
Adding user administrateur to group ftpadmin
Fait.
root@restructurationvlan:~#
```

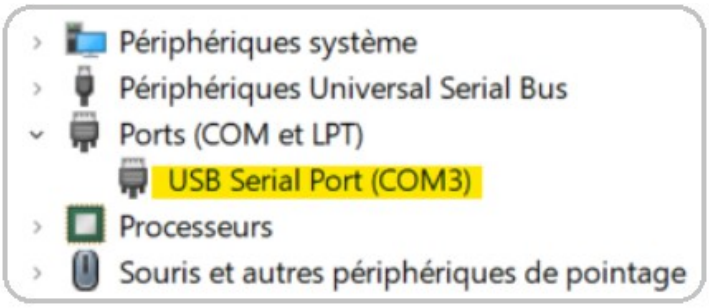
- Test de connexion à la session administrateur depuis un client FTP. Dans notre projet, nous en avons choisi un qui supporte les connexions chiffrées : Filezilla



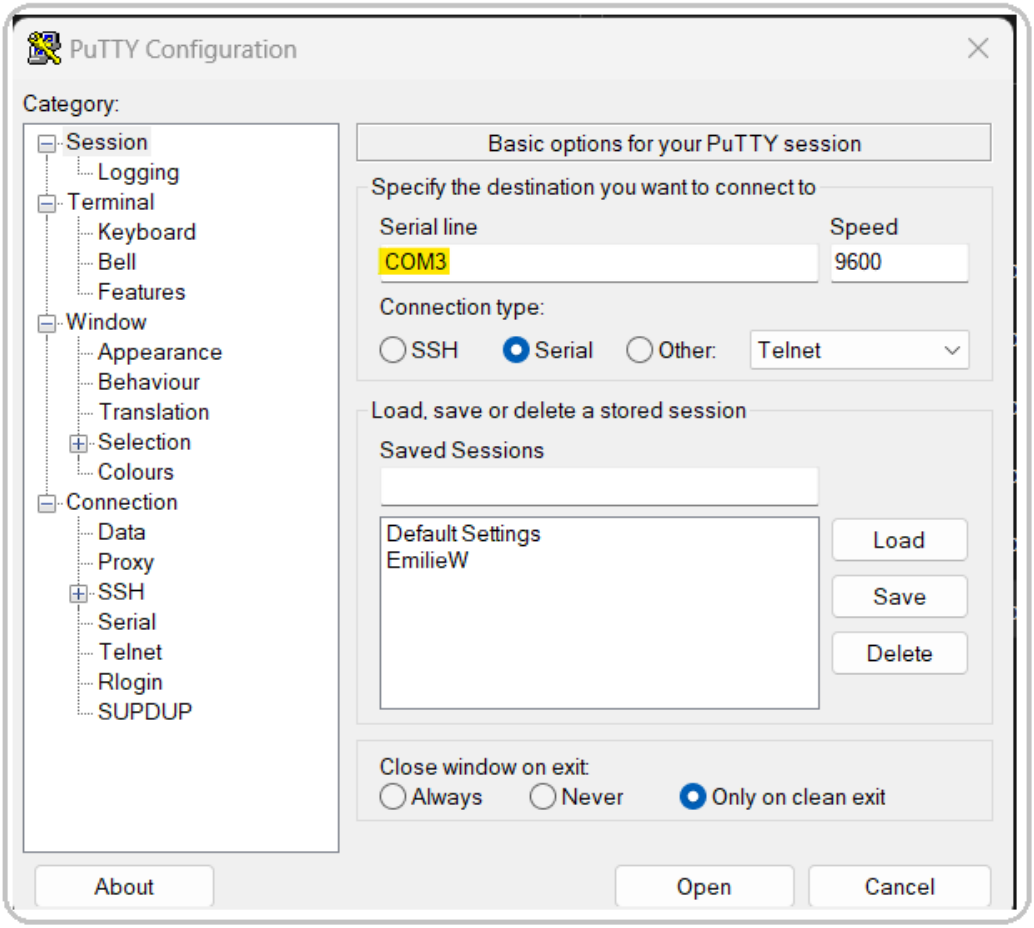
2.2. Switch

Nous allons à présent configurer notre switch Cisco 3560. Notre premier objectif est d'instaurer une connexion sécurisée avec le SSH (protocole réseau sécurisé et crypté permettant de se connecter à distance dans notre cas au switch) pour que des personnes habilitées puissent se connecter au switch à distance. Enfin, notre second objectif est de créer nos VLAN dessus.

Pour pouvoir configurer le switch, il est nécessaire d'avoir un câble console bleu RJ45 et un ordinateur avec un logiciel permettant d'émuler un terminal. Dans notre cas, nous avons choisi le logiciel Putty.
Avant de commencer la configuration de base du switch, il faut regarder dans le gestionnaire de périphériques le nom du port série pour pouvoir ensuite se connecter correctement avec Putty. Le nom de ce port comment par "COM" et est accompagné d'un chiffre. Dans notre cas, c'est "COM3".



Une fois sur Putty, dans l'encadré "Serial line" nous entrons le nom du port :

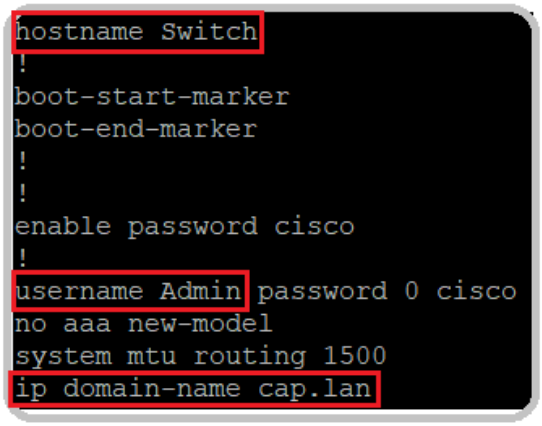


Lors de cette première connexion au switch, nous avons utilisé une connexion au protocole telnet qui est un équivalent du SSH mais en non sécurisé. Nous nous en servons seulement le temps de configurer le switch afin de se connecter en SSH.

2.2.1. Configuration d'une connexion SSH

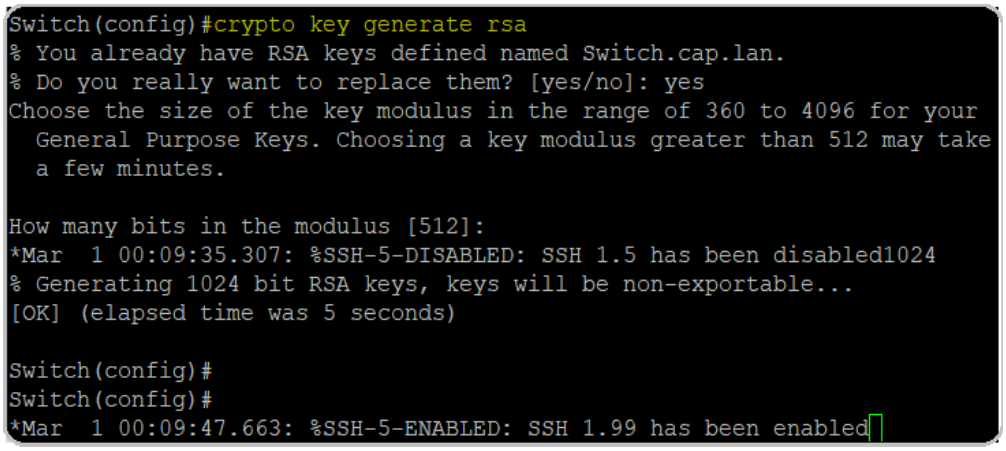
- création d'un nom d'hôte, d'un nom de domaine et d'un mot de passe pour le mode privilégié.
  Il faut être en mode configuration pour entrer les commandes suivantes :
  hostname 'Switch' , ip domain-name 'cap.lan' , enable password 'cisco'

- Vérification avec la commande sh run (une fois sortis du mode configuration):



→ ok

- Génération d'une paire de clés asymétriques, une méthode de chiffrement pour assurer une sécurité optimale pour le SSH.



→ dans la capture d'écran ci-dessus, nous avions déjà créé ces clés-ci. Nous réalisons à nouveau la procédure pour que vous puissiez la voir.

- Activation du protocole SSH avec la commande ip ssh version 2. Nous entrons ensuite en mode configuration VTY pour accepter, dans l'ordre des commandes :
    - seulement les connexions SSH au switch
    - que des connexions SSH vers d'autres équipements (facultatif)
    - et enfin enregistrer le compte utilisateur existant comme compte permettant de mettre en place la connexion.

```
Switch(config)#ip ssh version 2
Switch(config)#line vty 0 4
Switch(config-line)#transport input ssh
Switch(config-line)#transport output ssh
Switch(config-line)#login local
```

- Vérification : est-ce que la configuration a bien été prise en compte : sh run

```
!
line con 0
line vty 0 4
 login local
 transport input ssh
 transport output ssh
line vty 5 15
 login
!
end
```

→ oui

- Configuration d'une adresse IP et d'une passerelle afin de pouvoir se connecter en SSH. Pour cela, nous utilisons le VLAN par défaut, le VLAN1. Nous aurions aussi pu créer une vlan spécialement dédiée à cela.

```
Switch(config)#int vlan 1
Switch(config-if)#ip address 192.168.1.1 255.255.255.0
Switch(config-if)#ex
Switch(config)#ip default-ga
Switch(config)#ip default-gateway 192.168.1.2
Switch(config)#
```

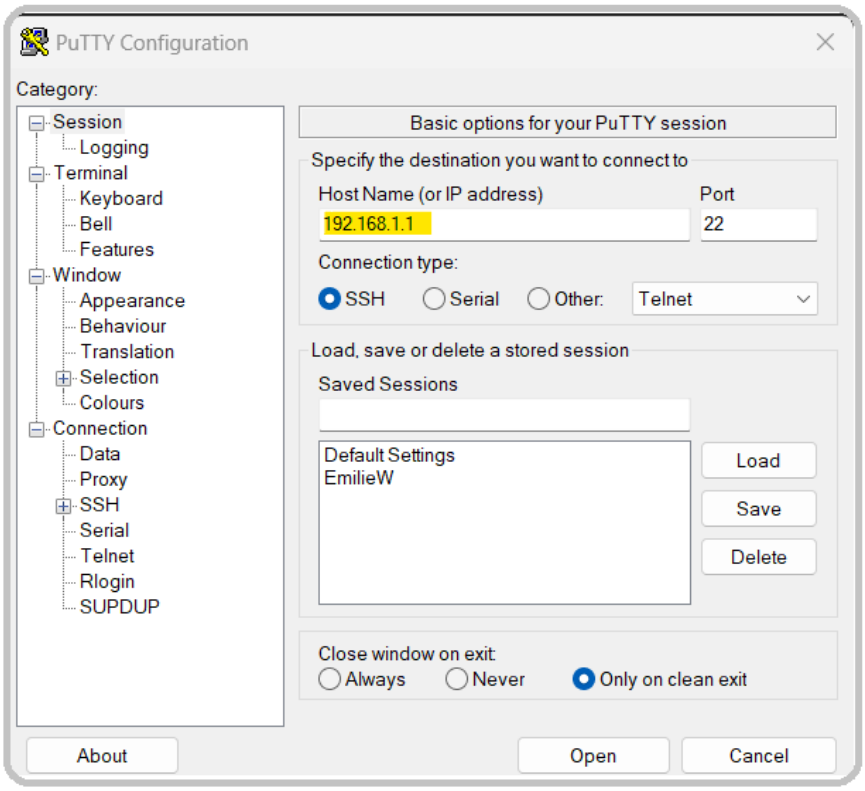
- Vérification : est-ce qu'un ordinateur étant dans le même réseau que le switch, peut communiquer avec lui ? switch : 192.168.1.1 et ordinateur : 192.168.1.20

```
C:\Users\KHDORBES>ping 192.168.1.1

Envoi d'une requête 'Ping'  192.168.1.1 avec 32 octets de données :
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=128
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=128
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=128
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=128

Statistiques Ping pour 192.168.1.1 :
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
```

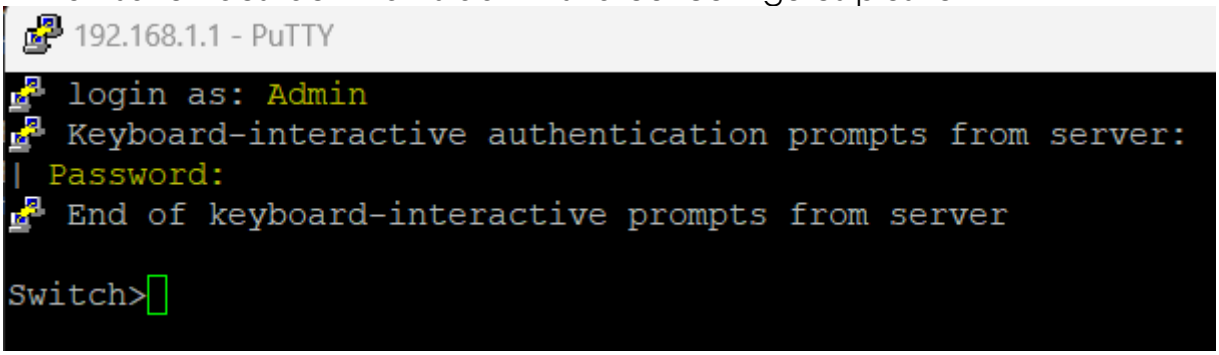
- Pour que la configuration soit persistante, nous entrons la commande copy running-config startup-config .

A présent, nous pouvons tester la connexion SSH.

- Avec notre ordinateur qui est dans le même réseau du switch : connexion sur Putty avec l'adresse du switch



- Utilisation des identifiants administrateur configurés plus tôt



→ la connexion s'est bien établie → ok

2.2.3. Configuration des VLAN sur le switch

Les 2 VLAN que nous allons créer vont permettre de séparer 2 zones, celle du service informatique où se trouve le serveur et celle du secrétariat.
Les commandes que nous utilisons ci-dessous sont valables pour les switch de la marque Cisco mais peuvent être différentes chez des concurrents comme HP par exemple.

Voici la procédure pour la création d'un VLAN. Nous avons effectué la même pour la seconde.

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 20
Switch(config-vlan)#name secretariat
Switch(config-vlan)#exit
Switch(config)#int range fa0/11-16
Switch(config-if-range)#switchpor
Switch(config-if-range)#switchport mor
Switch(config-if-range)#switchport mod
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchp
Switch(config-if-range)#switchport acce
Switch(config-if-range)#switchport access vlan 20
```

Dans l'ordre des commandes :

- création du VLAN 20
- attribution d'un nom, ici "secrétariat"
- attribution des interfaces 11 à 16 du switch à ce VLAN
- passage du port en mode Access et attribution au VLAN 20 (le port ne recevra que les paquets qui lui sont destinés et ne verra pas les autres VLAN)

Pour activer les ports du switch, nous utilisons la commande no shut

Vérification de nos interfaces et des ports qui ont été attribués avec la commande sh vlan

```
VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                 Gi0/1, Gi0/2
10   info                              active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                 Fa0/9, Fa0/10
20   secretariat                       active    Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                 Fa0/15, Fa0/16
```

Nous allons à présent passer le port Fa0/24 en mode Trunk. Ce port est celui sur lequel sera connecté notre routeur. Le passer en mode Trunk permet la visibilité de tous les VLAN et surtout le passage de tous les paquets donc une communication entre les VLANs.

Pour passer correctement ce port en mode Trunk sur le switch Cisco 3560, il faut entrer les commandes suivantes :

- conf t
- int fa0/24
- switchport trunk encapsulation dot1q (encapsulation permet d'élargir la compatibilité des échanges)
- switchport mode trunk

Vérification

- avec sh run

```
!
interface FastEthernet0/24
 switchport trunk encapsulation dot1q
 switchport mode trunk
```

- avec sh interface trunk

```
Switch#sh interface trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/24      on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/24      1,10,20,30

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      1,10,20,30
Switch#
```

→ les vlan 1, 10, 20, 30 peuvent communiquer ensemble.

2.3.Routeur

Le routeur va permettre de faire le lien entre les différents réseaux et VLANs.
Comme indiqué dans la partie "**Contexte**", nous avons fait le choix d'utiliser un routeur virtuel.

Pour le mettre en place, il faut installer un routeur virtuel sur une machine virtuelle. Pour les besoins du projet, notre VM possède 3 cartes réseaux. Nous y avons également préinstallé PFsense. Il nous manque plus qu'à le configurer.

2.3.1. Configuration du routeur virtuel

Voici l'interface qui apparaît lorsque nous sommes sur la VM PFsense

```
 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
 8) Shell
```

- Il faut assigner les interfaces aux cartes réseaux (option 1 : Assign Interfaces) :
  - **WAN** : hn0
  - **LAN** : hn1
  - **OPT1** : hn2
  - **OPT2** : hn3

```
Enter an option: 1


Valid interfaces are:

hn0     00:15:5d:38:01:17   (up) Hyper-V Network Interface
hn1     00:15:5d:38:01:18   (up) Hyper-V Network Interface
hn2     00:15:5d:38:01:1a (down) Hyper-V Network Interface

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y|n]? n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(hn0 hn1 hn2 or a): hn0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(hn1 hn2 a or nothing if finished): hn1

Enter the Optional 1 interface name or 'a' for auto-detection
(hn2 a or nothing if finished): hn2
```

-

- Attribution d'adresses IP pour chaque interface (option 2 : Set Interface(s) IP address)

  - **WAN** : en DHCP pour aller sur internet
  - **LAN** : ip static dans le même réseau que le VLAN 10 : 192.168.10.254
  - **OPT1** : ip static dans le même réseau que le VLAN 20 : 192.168.20.254
  - **OPT2** : ip static dans le même réseau que le VLAN 30 : 192.168.30.254

- Nous vérifions ensuite qu'un utilisateur de chaque réseau puisse faire un ping sur l'interface concernée avant d'activer les VLANs.

  - ping de Pfsense vers utilisateur 192.168.10. 30

```
Enter a host name or IP address: 192.168.10.30

PING 192.168.10.30 (192.168.10.30): 56 data bytes
64 bytes from 192.168.10.30: icmp_seq=0 ttl=128 time=5.568 ms
64 bytes from 192.168.10.30: icmp_seq=1 ttl=128 time=1.358 ms
64 bytes from 192.168.10.30: icmp_seq=2 ttl=128 time=1.184 ms

--- 192.168.10.30 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.184/2.703/5.568/2.027 ms
```

→ ok

- ping de l'utilisateur 192.168.10.30 vers l'interface PFsense + ping sur Google pour voir si l'interface WAN est correctement configurée

```
Statistiques Ping pour 192.168.10.2:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms

C:\Users\Fabien>ping 8.8.8.8

Envoi d'une requête 'Ping'  8.8.8.8 avec 32 octets de données :
Réponse de 8.8.8.8 : octets=32 temps=113 ms TTL=112
Réponse de 8.8.8.8 : octets=32 temps=48 ms TTL=112
Réponse de 8.8.8.8 : octets=32 temps=56 ms TTL=112
Réponse de 8.8.8.8 : octets=32 temps=166 ms TTL=112

Statistiques Ping pour 8.8.8.8:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 48ms, Maximum = 166ms, Moyenne = 95ms
```

→ ok

- Ping du 192.168.20.20 vers 192.168.20.2

```
C:\Users\Fabien>ping 192.168.20.2

Envoi d'une requête 'Ping'  192.168.20.2 avec 32 octets de données :
Réponse de 192.168.20.2 : octets=32 temps<1ms TTL=64
Réponse de 192.168.20.2 : octets=32 temps<1ms TTL=64
Réponse de 192.168.20.2 : octets=32 temps<1ms TTL=64
Réponse de 192.168.20.2 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.20.2:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
```

→ ok

2.3.2. Attribution des VLAN

Sur notre virtualiseur, nous avons 2 types de cartes réseaux : cmexterne et cminterne.

Cmexterne est connectée à la carte wifi de l'ordinateur et est attribuée à la carte WAN du PFsense seulement pour accéder au net.

Cminterne est connectée à la carte du switch.

Nous utilisons Hyper-V comme solution de virtualisation. La procédure qui va suivre est valable pour cette solution mais peut différer par rapport à d'autres comme Virtual Box.

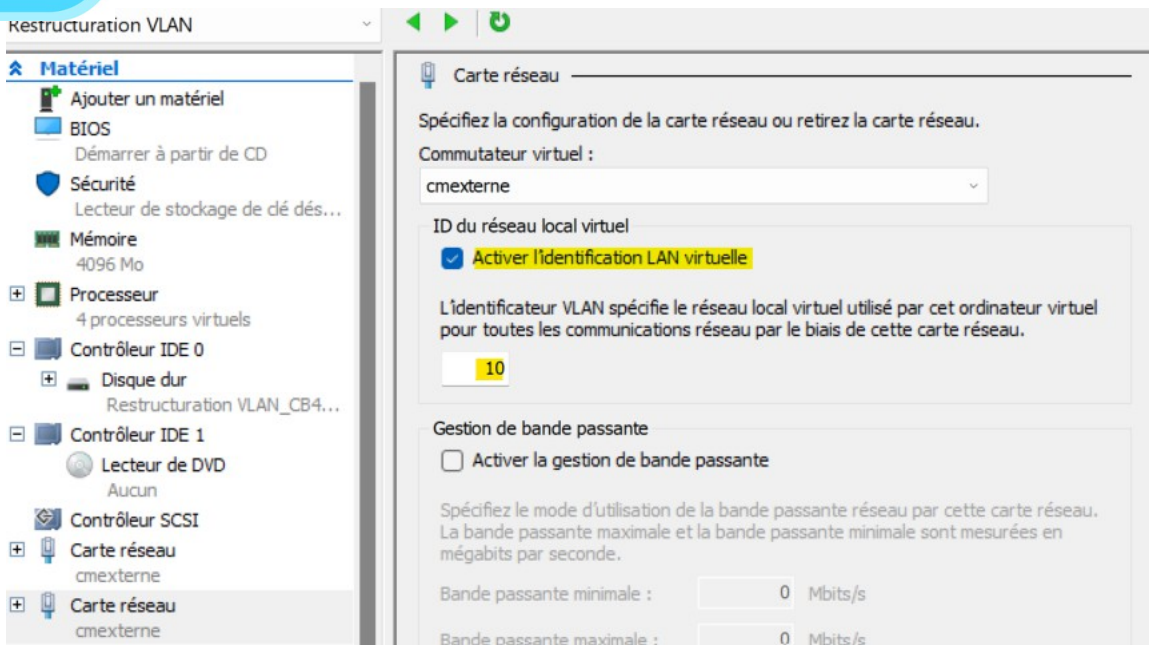
Les VLAN ont été activés au niveau du switch, mais il faut également autoriser les machines virtuelles sous Hyper-V à utiliser un VLAN.
Pour cela, nous nous rendons dans les paramètres de chaque machine virtuelle appartenant à un VLAN. Puis, dans l'onglet matériel, sur chaque carte réseau concerné, nous activons l'identification LAN virtuelle et notons le VLAN à laquelle elle est rattaché.

Machines concernées :

- Serveur "restructurationvlan" : carte réseau eth1
- Routeur PFsense : cartes réseaux LAN, OPT1, OPT2
- Machines utilisateurs : on active le vlan voulu sur la seule carte réseau disponible

Exemple sur le serveur (carte eth1) :

- activation de l'identification virtuelle et nous notons le VLAN auquel il appartient, ici le 10.



– Pour vérifier que le VLAN 10 fonctionne correctement, sur le serveur, nous tentons un ping vers la carte LAN du routeur (VLAN 10)

```
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.795 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=0.995 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=64 time=1.03 ms
64 bytes from 192.168.10.2: icmp_seq=4 ttl=64 time=0.611 ms
^C
--- 192.168.10.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 24ms
rtt min/avg/max/mdev = 0.611/0.858/1.033/0.171 ms
```

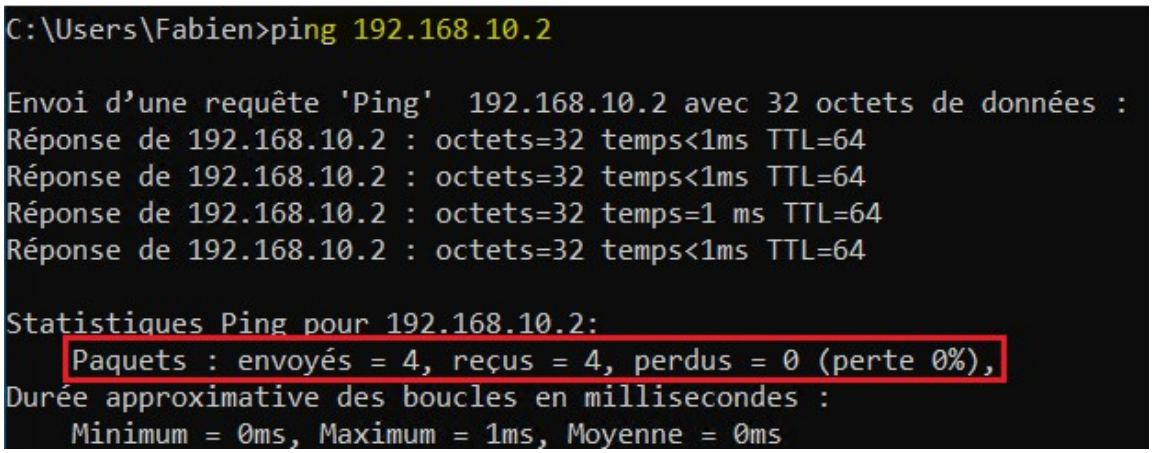
→ ok

Après avoir activé les VLAN 10 et 20 sur les postes concernés, nous vérifions que le VLAN 20 fonctionne et que les 2 VLAN communiquent entre eux.

– ping poste 192.168.20.20 vers routeur (VLAN20)

```
C:\Users\Fabien>ping 192.168.20.2

Envoi d'une requête 'Ping'  192.168.20.2 avec 32 octets de données :
Réponse de 192.168.20.2 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.20.2 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.20.2 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.20.2 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.20.2:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
```

- Ping VLAN 20 (192.168.20.20) vers passerelle VLAN 10 (192.168.10.2) :



Nos VLANs sont donc bien activés et communiquent entre eux. Nous en avons donc terminé avec ce point.

2.4. Serveur DHCP

Un serveur DHCP (Dynamic Host Configuration Protocol) permet de délivrer des adresses IP aux ordinateurs qui se connectent au réseau concerné.
Un serveur DHCP a besoin d'un accès à internet et d'un adressage fixe. C'est déjà le cas ici, nous allons donc passer à la suite.

2.4.1. Configuration sur le poste serveur

Pour activer le serveur DHCP sur le poste serveur Debian, nous allons tout d'abord paramétrer le service isc-dhcp que nous avons déjà téléchargé.

- Configuration du fichier *isc-dhcp-server* pour indiquer quelle carte réseau le serveur DHCP va recevoir et envoyer des requêtes.
  Nous ne sommes intéressés que par l'adressage IPv4 donc nous laissons l'IPv6 commenté.

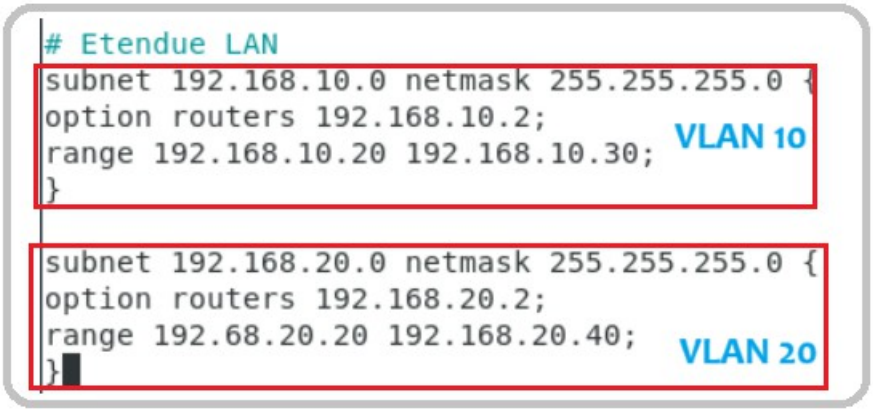
```
  GNU nano 3.2                /etc/default/isc-dhcp-server

# 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="eth1"
#INTERFACESv6=""
```

– Configuration du service DHCP dans le fichier *dhcpd.conf*.

```
  GNU nano 3.2                     /etc/dhcp/dhcpd.conf

# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#

# option definitions common to all supported networks...
option domain-name "test.lan";
option domain-name-servers 192.168.10.1;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
```

- Nous notons en premier notre nom de domaine, 'esicad.lan'.
- Option "**domaine-name-servers**" : notre serveur DNS est le même pour toutes les étendues déclarés, nous l'entrons.
- Les durées des baux pour les adresses sont comprises entre 1h et 2h
- "**ddns-update-style**" sert à définir le type de mise à jour du DNS. Nous n'en avons pas besoin ici donc nous le laissons à l'état "**none**".
- Le paramètre "**authoritative**" a été décommenté car ce serveur DHCP est le serveur officiel de notre réseau local.

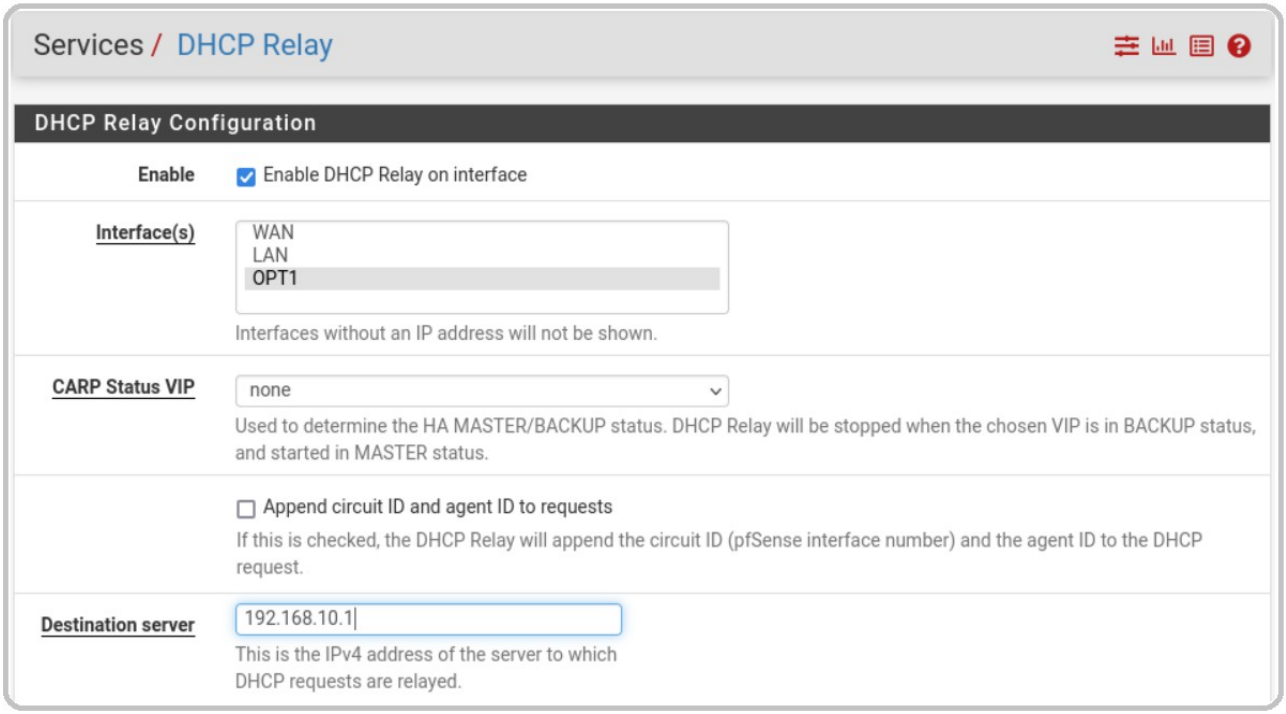
– Fin du fichier *dhcpd.conf* : les étendues sont déclarées avec l'adresse du réseau, son masque de sous-réseau, la passerelle de l'étendue et la plage d'adresses à distribuer pour les VLAN 10 et 20



– Redémarrage du service isc-dhcp-server

2.4.2. Activation du relais DHCP

Ensuite sur PFsense, nous activons le relais DHCP pour l'interface voulue : OPT1 (VLAN20)



- Sur un ordinateur utilisateur du réseau VLAN 20, nous vérifions que l'attribution automatique d'une adresse dans l'étendue déclarée fonctionne.

  Dans les paramètres Windows, nous avons bien passer l'adressage IP en DHCP.

  Ensuite, dans l'invite de commande, nous relâchons l'adresse IP que l'ordinateur pouvait avoir avec ip config / release et nous faisons en sorte qu'il en cherche une autre avec la commande ipconfig /renew.

```
Microsoft Windows [version 10.0.19045.3448]
(c) Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>ipconfig /release

Configuration IP de Windows


Carte Ethernet Ethernet :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv6 de liaison locale. . . . .: fe80::4f9d:28f:1ecf:5633%12
   Passerelle par défaut. . . . . :

C:\Windows\system32>ipconfig /renew

Configuration IP de Windows


Carte Ethernet Ethernet :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv6 de liaison locale. . . . .: fe80::4f9d:28f:1ecf:5633%12
   Adresse IPv4. . . . . .: 192.168.20.20
   Masque de sous-réseau. . . . . : 255.255.255.0
   Passerelle par défaut. . . . . : 192.168.20.2
```

→ l'adresse IP fait bien partie de la plage d'adresses du VLAN 20. Le serveur fonctionne donc correctement.

2.5. ACL sur routeur

Une ACL (Access Control List) est un ensemble de règles qui détermine si un utilisateur, un groupe, peuvent accéder ou non à des services et quelles opérations sont acceptées sur ces services.
Dans notre projet, une règle est nécessaire pour que seul le poste administrateur puisse se connecter en FTP.
Pour cela, nous nous connectons à PFsense et nous allons dans le parefeu.

- Sur l'interface LAN, nous supprimons la ligne qui accepte tout en ipv4 et nous ajoutons une ligne qui accepte le ftp que pour un ordinateur (admin) et autre ligne qui accepte les pings (ICMP)

☐	✓	0/0 B	IPv4 TCP	192.168.10.5/24	*	192.168.10.1/24	21 (FTP)	*	none	Accepter FTP pour poste administrateur
☐	✓	0/0 B	IPv4 ICMP any	*	*	*	*	*	none	Ping

– Vérification connexion FTP sur l'ordinateur 192.168.10.5

```
C:\Windows\system32>ipconfig

Configuration IP de Windows


Carte Ethernet Ethernet :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv6 de liaison locale. . . . .: fe80::4f9d:28f:1ecf:5633%12
   Adresse IPv4. . . . . . . . . . . . . .: 192.168.10.5
   Masque de sous-réseau. . . . . . . . . : 255.255.255.0
   Passerelle par défaut. . . . . . . . . : 192.168.10.2
```

```
C:\Windows\system32>ftp 192.168.10.1
Connecté à 192.168.10.1.
220 ProFTPD Server (restructurationvlan.test.lan) [192.168.10.1]
200 UTF-8 activÃ©
Utilisateur (192.168.10.1:(none)) : administrateur
331 Mot de passe requis pour administrateur
Mot de passe :
230 Utilisateur administrateur authentifié
ftp>
```

– Nous effectuons ensuite la même vérification sur un autre poste du VLAN 10 puis un poste du VLAN 20. Nous nous apercevons qu'ils peuvent aussi se connecter en FTP, ce qui n'aurait pas dû être le cas. Notre règle n'a peut-être pas été pensée et entrée correctement. Il aurait peut-être fallu entrer cette règle également sur l'interface OPT1.