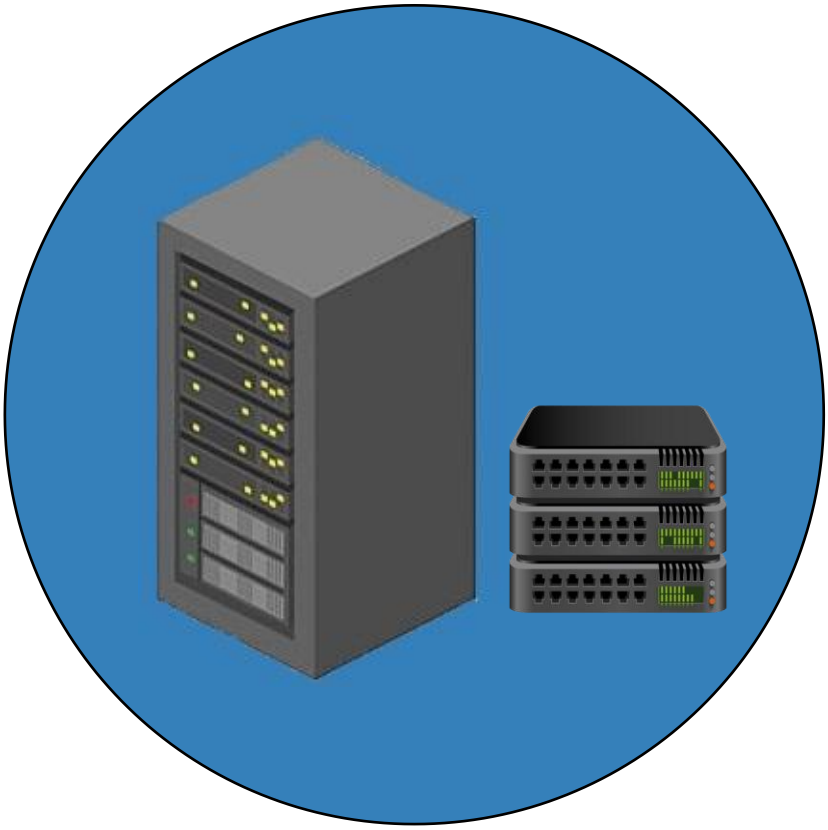
PR04 : SIO SISR – Restructuration de réseaux par VLAN et déploiement de services



Sommaires

Contexte :

Vous venez d'être embauché comme assistant administrateur dans un centre de formation qui souhaite restructurer son réseau pour séparer les salles de formation du réseau du service administratif, dans le but sécuriser les accès aux ressources administratives.

Actuellement tous les ordinateurs, serveurs et imprimantes sont situés dans un seul réseau logique 192.168.100.0/24. **Votre mission consiste à proposer une solution permettant de séparer les flux formation et les flux administratifs, en utilisant les matériels disponibles**. **Vous devez proposer une maquette à faire valider par l'administrateur du réseau**.

Certaines applications, notamment des sites web des enseignes de l'école doivent être accessibles à tous en consultation. Par contre, l'administration et la mise à jour des sites sont réservées aux développeurs du centre de formation et ne peut se faire que depuis que le réseau administratif.

Objectifs :

1- Déployer un serveur DHCP pour délivrer des adresses à tous les utilisateurs du parc informatique.

2- Mettre en place un serveur WEB avec un accès sécurisé à intranet pour nos utilisateurs.

3- Créer un service DNS pour utiliser des noms d'hôtes FQDN (Full qualified Domaine Name)

4- Scinder le réseau en 3 VLANS, un pour chaque secteur demandé.

5- Effectuer un routage inter-vlan pour que les différents réseaux communiquent ensemble.

6- Filtrage du trafic par ACL.

7- Effectuer une démonstration significative montrant la sécurité de la fiabilité de notre projet, répondant au cahier des charges.

Cahier des charges :

1- Proposer une architecture de réseau avec tous les composants nécessaires et les paramètres essentiels (services et paramètres TCP/IP) permettant de modéliser la réponse au cahier des charges.

2- Tous les serveurs de l'entreprise doivent être regroupés dans un seul réseau pour faciliter le filtrage des flux vers les serveurs. Dans votre PoC, le réseau serveurs hébergera notamment un serveur web mutualisé, un serveur FTP, un DNS et un serveur DHCP.

3- Le serveur Web hébergera deux sites (www.mbway.lan et www.digitalschol.lan) accessibles à tous les utilisateurs.

4- Le serveur FTP sera accessible uniquement depuis le poste des développeurs, situé dans le réseau administratif.

5- Les salles de formation ne peuvent pas accéder au sous-réseau du personnel administratif.

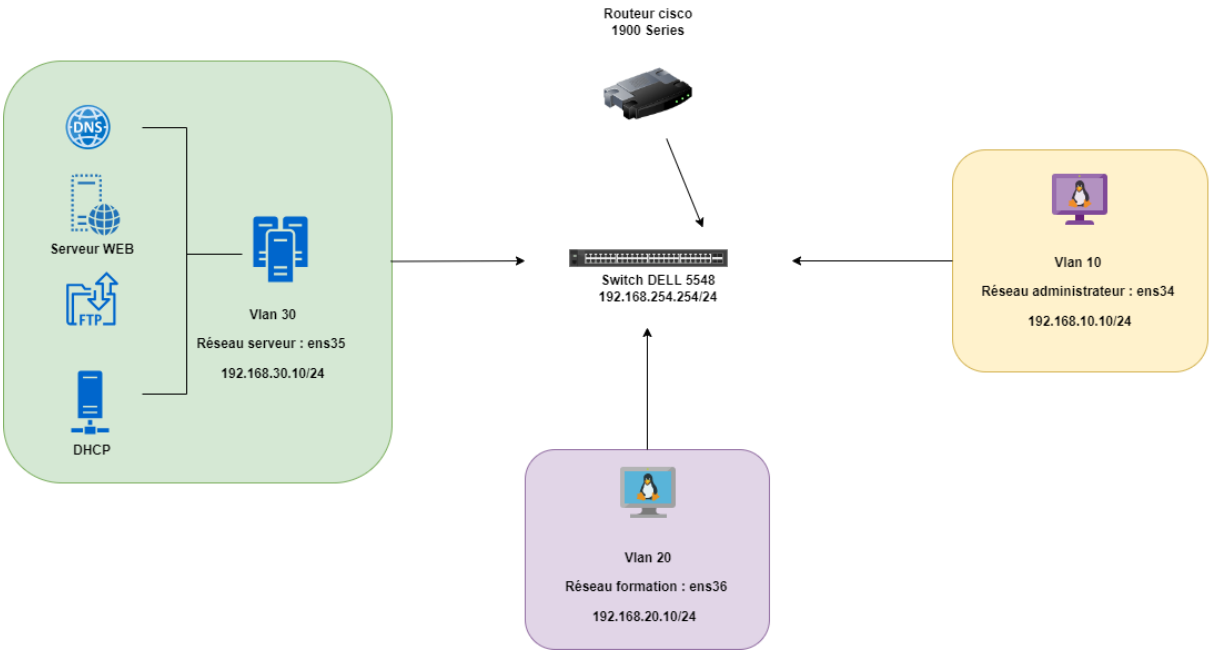
6- Le poste de l'administrateur peut accéder aux postes et équipements des salles de formation.

7- Tous les sous-réseaux reçoivent leurs paramètres TCP/IP d'un serveur DHCP commun situé dans le réseau dédié aux serveurs.

Solution :

Pour répondre au problème du campus, je vais mettre en place une VM sous Linux qui hébergera mon serveur WEB contenant mon intranet, un serveur DHCP, un serveur DNS et FTP, ainsi que deux VM pour le réseau Formation et Administration.

Schéma ASI :



Prérequis :

Tout d'abord nous allons devoir installer puis paramétrer une machine virtuelle qui nous servira de Serveur WEB (Apache2), DNS (bind9), FTP (Proftpd) et DHCP (isc-dhcp-server).

Ensuite on va configurer deux VM sous Debian : une pour le réseau administrateur et une autre pour le réseau formation.

Les cartes réseau seront configurer en mode « Accès par pont »

Nous utiliserons et configurerons un switch DELL 5548 de niveau deux ainsi qu'un routeur Cisco 1900 Series

Nous allons maintenant scinder notre projet en 6 parties :

- Configuration du serveur et mise en place des services
- Configuration de notre switch
- Configuration de notre routeur
- Configuration de nos postes administrateur et formation
- Filtrage du trafic par ACL
- Démonstration

Configuration du serveur et mise en place des services :

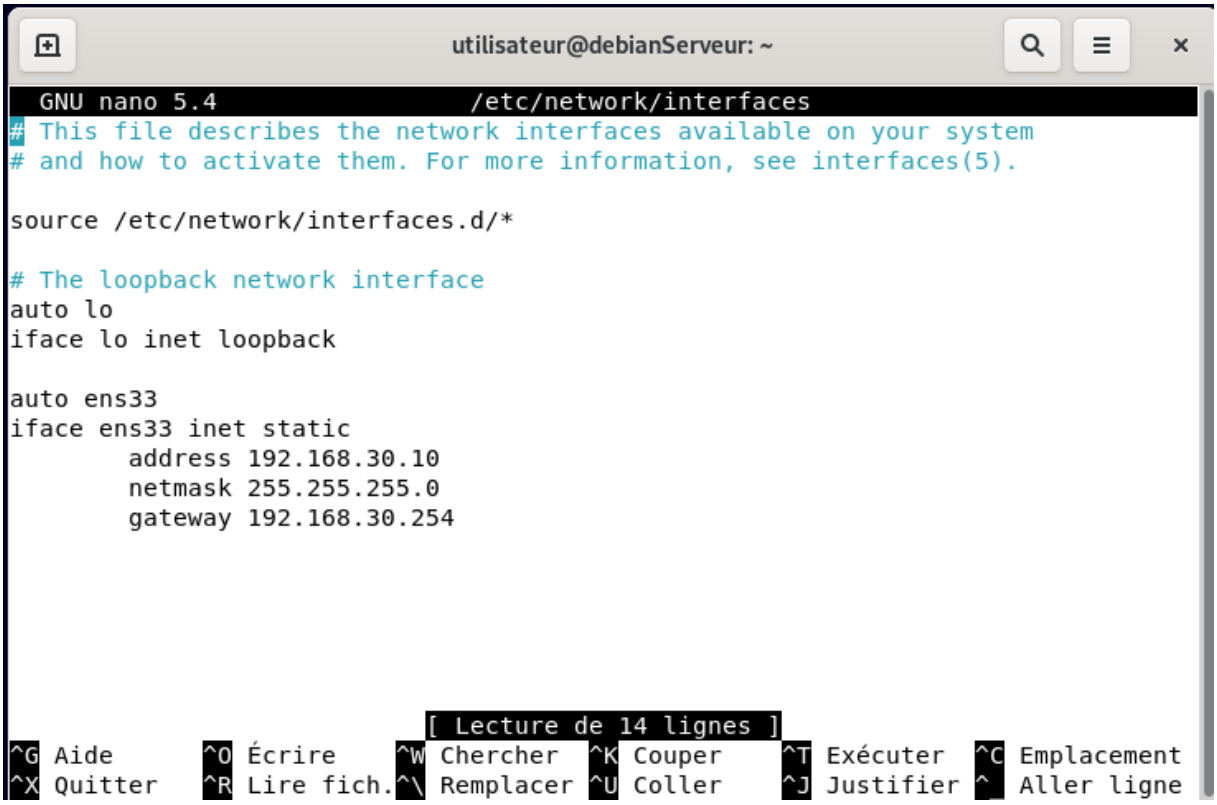
Création de la DMZ :

On va d'abord aller dans le fichier /etc/network/interfaces afin de paramétrer notre carte réseaux avec la commande sudo nano /etc/network/interfaces.

On redémarrera les interfaces avec la commande sudo systemctl restart networking.service pour que les changements prennent effet.

Il ne faut pas oublier d'activer l'accès par pont.

Pour notre serveur on lui définit comme adresse IP : 192.168.30.10



Maintenant que l'interface réseau est configurée on va pouvoir installer les services nécessaires.

Configuration du serveur FTP :

Nous devons installer au préalable ProFTPD sur notre VM SERVEUR Debian11

Installation de Proftpd :

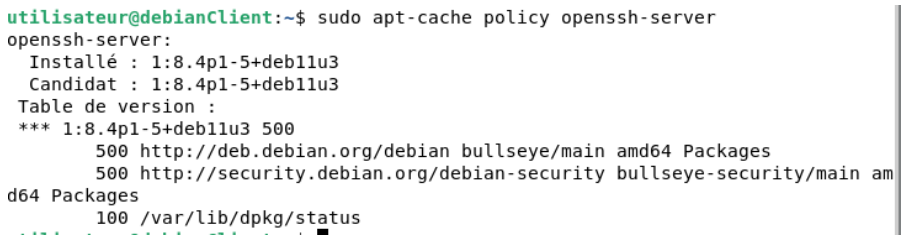
- sudo apt-get update
- sudo apt install proftpd

Une fois installé nous pouvons choisir de créer des utilisateurs mais nous allons nous connecter en mode utilisateur pour nos tests.

FTP n'est pas un protocole sécurisé. Pour éviter la transmission d'informations en clair, il est nécessaire de crypter les données en transit. Nous utiliserons donc principalement le protocole SFTP.

Pour cela, nous devrons installer OpenSSH sur les serveurs ainsi que sur les postes clients via la commande :

- sudo apt install openssh-server



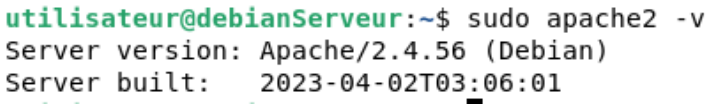
Configuration du Serveur Web :

On va devoir configurer Apache 2 avec nos 2 intranets et mettre en place le protocole HTTPS

Dans un premier temps faire sudo apt-get update afin d'avoir les dernières versions disponibles (il faut le faire avant toutes installations sous Debian)

Installation d'Apache2 :

- sudo apt install Apache2



On constate donc qu'Apache2 est correctement installé.

Configuration des deux intranets :

On va créer deux dossiers MBWay et DigitalSchool :

Aller dans le dossier cd /var/www/html est créé deux dossiers avec la commande mkdir :

```
utilisateur@debianServeur:~$ cd /var/www/html/
utilisateur@debianServeur:/var/www/html$ ls -l
total 20
drwxr-xr-x 2 root root  4096 15 avril 12:46 digitalschool
-rw-r--r-- 1 root root 10701  7 avril 21:19 index.html
drwxr-xr-x 2 root root  4096 13 avril 18:18 mbway
utilisateur@debianServeur:/var/www/html$
```

Pour chaque dossier nous allons créer un fichier index.html que nous allons ensuite configurer :

```
utilisateur@debianServeur:/var/www/html$ cd mbway
utilisateur@debianServeur:/var/www/html/mbway$ ls -l
total 4
-rw-r--r-- 1 root root 123 13 avril 18:18 index.html
utilisateur@debianServeur:/var/www/html/mbway$ cd ..
utilisateur@debianServeur:/var/www/html$ cd digitalschool
utilisateur@debianServeur:/var/www/html/digitalschool$ ls -l
total 8
-rw-r--r-- 1 root root 131 14 avril 11:13 index.html
```

Fichier index.html pour DigitalSchool :

```
utilisateur@debianServeur:/var/www/html/digitalschool$ cat index.html
<!DOCTYPE html>
<html>
<body>
        <h1>Bienvenue sur le site de Digitalschool</h1>
        <p>Ceci est la page d'acceuil.</p>
</body>
</html>
```

On va créer le même type pour MBWay.

Nous avons donc créé un dossier où nous importerons nos fichiers html, PHP, JavaScript etc… respectifs à chaque site. Ici nous n'avons déposé qu'un « index.html » pour le moment.

Configuration des Virtual hosts :

Grâce à l'étape précédente nous avons créé nos intranets.

De ce fait pour y accéder nous sommes obligés d'écrire : http://192.168.30.5/mbway/ pour accéder au site d'MBWay par exemple.

Pour corriger cela nous allons donc configurer des Virtual Host pour accéder à nos sites depuis l'adresse http://www.mbway.lan

On va aller dans le dossier /etc/apache2/sites-available/

On a un fichier 000-default.conf avec une configuration par default que l'on va copier (commande cd) et créer deux fichiers un pour MBWay et un autre pour DigitalSchool.

```
utilisateur@debianServeur:~$ cd /etc/apache2/sites-available/
utilisateur@debianServeur:/etc/apache2/sites-available$ ls -l
total 36
-rw-r--r-- 1 root root 1332  2 avril  2023 000-default.conf
-rw-r--r-- 1 root root 6387 14 avril 19:53 default-ssl.conf
-rw-r--r-- 1 root root 1351 13 avril 15:31 digitalschool.conf
-rw-r--r-- 1 root root 6390 16 avril 09:20 digitalschool-ssl.conf
-rw-r--r-- 1 root root 1636 14 avril 19:38 mbway.conf
-rw-r--r-- 1 root root 6374 16 avril 09:19 mbway-ssl.conf
utilisateur@debianServeur:/etc/apache2/sites-available$
```

On va ensuite les configurer :

```
utilisateur@debianServeur:/etc/apache2/sites-available$ cat mbway.conf
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # 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.mbway.lan

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/mbway
```

```
utilisateur@debianServeur:/etc/apache2/sites-available$ cat digitalschool.conf
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port
        # 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) th
        # 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.digitalschool.lan

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/digitalschool
```

On modifie ServerName par le nom de nos serveurs : www.mbway.lan et www.digitalschool.lan.

On lui indique la route avec DocumentRoot : /var/www/html/mbway et /var/www/html/digitalschool .

On va activer nos sites avec la commande a2ensite + le nom du virtual Host

Maintenant si on ajoute www.mbway.lan en alias à notre serveur dans le fichier /etc/hosts de notre client, vous pouvez voir la page d'accueil de notre site dans notre navigateur.

On vérifie que nos sites fonctionnent :

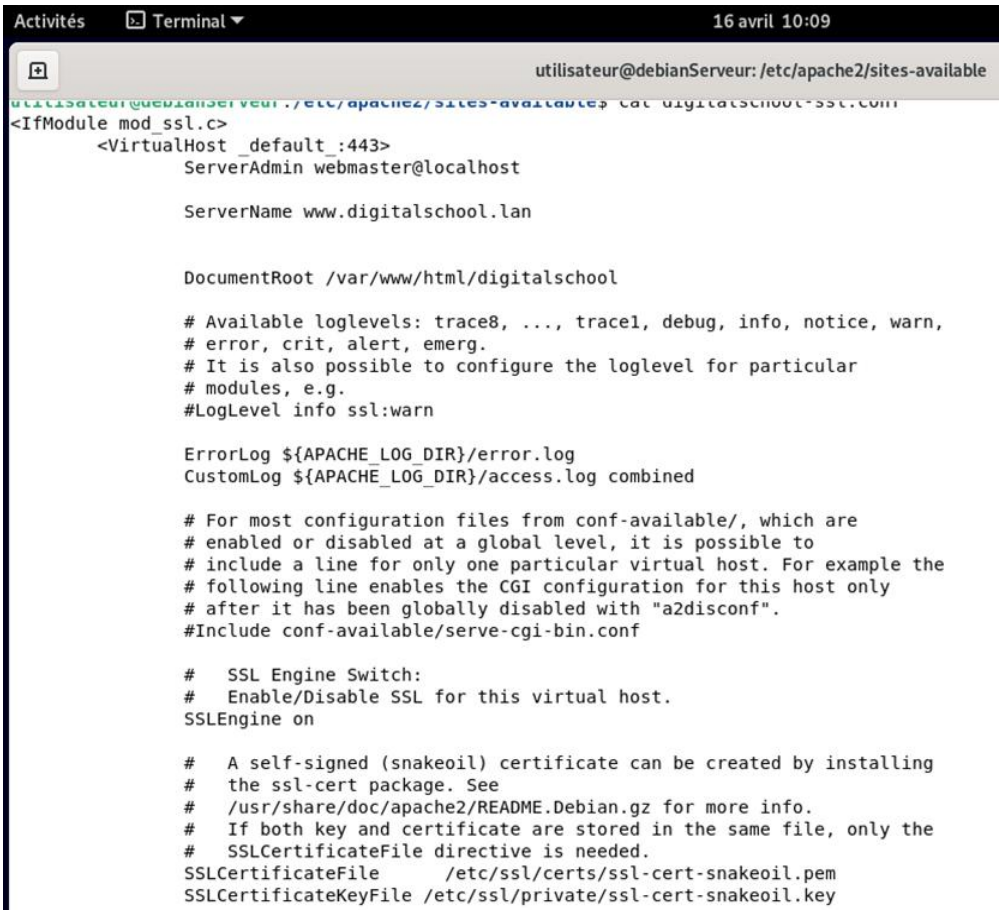
Mise en place du protocole HTTPS :

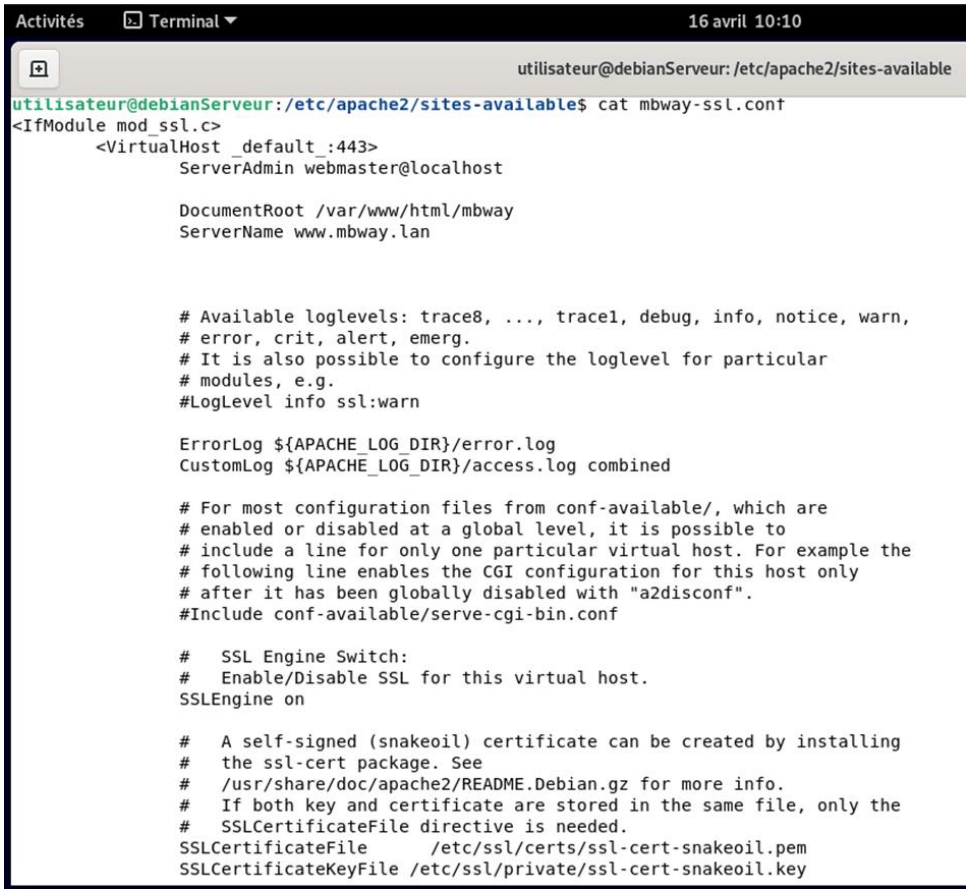
La connexion à nos sites intranets ne se fait de base qu'en HTTP ce qui n'est absolument pas sécurisé. Pour éviter de faire circuler des informations en clair, nous devons crypter les données qui transitent par le biais du protocole HTTPS.

On a un fichier avec une configuration déjà présente sous Debian :

```
utilisateur@debianServeur:/etc/apache2/sites-available$ ls -l
total 36
-rw-r--r-- 1 root root 1332  2 avril  2023 000-default.conf
-rw-r--r-- 1 root root 6387 14 avril 19:53 default-ssl.conf
-rw-r--r-- 1 root root 1351 13 avril 15:31 digitalschool.conf
-rw-r--r-- 1 root root 6390 16 avril 09:20 digitalschool-ssl.conf
-rw-r--r-- 1 root root 1636 14 avril 19:38 mbway.conf
-rw-r--r-- 1 root root 6374 16 avril 09:19 mbway-ssl.conf
utilisateur@debianServeur:/etc/apache2/sites-available$
```

On va donc copier le fichier default-ssl et créer deux fichiers pour chaque site puis les configurer :


Activités
Terminal
16 avril 10:09
utilisateur@debianServeur: /etc/apache2/sites-available
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
ServerAdmin webmaster@localhost
ServerName www.digitalschool.lan
DocumentRoot /var/www/html/digitalschool
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
CustomLog ${APACHE_LOG_DIR}/access.log combined
For most configuration files from conf-available/, which are
enabled or disabled at a global level, it is possible to
include a line for only one particular virtual host. For example the
following line enables the CGI configuration for this host only
after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
SSL Engine Switch:
Enable/Disable SSL for this virtual host.
SSLEngine on
A self-signed (snakeoil) certificate can be created by installing
the ssl-cert package. See
/usr/share/doc/apache2/README.Debian.gz for more info.
If both key and certificate are stored in the same file, only the
SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key


Activités
Terminal
16 avril 10:10
utilisateur@debianServeur: /etc/apache2/sites-available
utilisateur@debianServeur:/etc/apache2/sites-available$ cat mbway-ssl.conf
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/mbway
ServerName www.mbway.lan
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
CustomLog ${APACHE_LOG_DIR}/access.log combined
For most configuration files from conf-available/, which are
enabled or disabled at a global level, it is possible to
include a line for only one particular virtual host. For example the
following line enables the CGI configuration for this host only
after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
SSL Engine Switch:
Enable/Disable SSL for this virtual host.
SSLEngine on
A self-signed (snakeoil) certificate can be created by installing
the ssl-cert package. See
/usr/share/doc/apache2/README.Debian.gz for more info.
If both key and certificate are stored in the same file, only the
SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

On donne le nom de notre serveur :

ServerName www.mbway.lan et ServerName www.digitalschool.lan ainsi que la route avec DocumentRoot /var/www/html/mbway et /var/www/html/digitalschool .

On va activer a2enmod ssl puis a2ensite default-ssl

On va activer nos sites avec la commande a2ensite + le nom du virtual Host

Puis redemarrer apache2 : service apache2 reload

La recommandation est de créer/acheter un certificat pour chaque site plutôt que d'utiliser la configuration de base d'Apache2

C'est donc ce que nous allons faire :

Dans un premier temps nous allons verifier que le module SSL est bien activée :

- sudo a2enmod ssl

Nous allons ensuite générer nos clés privées et des CSR(certificate signing request) :
Pour Mbway : openssl req -newkey rsa:2048 -nodes -keyout /etc/ssl/private/mbway.key -out /etc/ssl/certs/mbway.csr

Et pour Digitalschool : openssl req -newkey rsa:2048 -nodes -keyout /etc/ssl/private/digitalschool.key -out /etc/ssl/certs/ digitalschool.csr

Nous devrons répondre à une suite de question :

```
Mot de passe :
root@debian11:/home/utilisateur# openssl req -newkey rsa:2048 -nodes -keyout /etc/ssl/private/mbway.key -out /etc/ssl/certs/mbway.csr
Generating a RSA private key
.....+++++
.......+++++
writing new private key to '/etc/ssl/private/mbway.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:fr
State or Province Name (full name) [Some-State]:paris
Locality Name (eg, city) []:paris
Organization Name (eg, company) [Internet Widgits Pty Ltd]:thomasit
Organizational Unit Name (eg, section) []:it
Common Name (e.g. server FQDN or YOUR name) []:thomas
Email Address []:thomas.fr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:root
An optional company name []:
root@debian11:/home/utilisateur#
```

Nous allons enfin passer a la création de nos certificats auto-signés :

Pour Mbway : sudo openssl req x509 -req -days 365 -in /etc/ssl/certs/mbway.csr -signkey

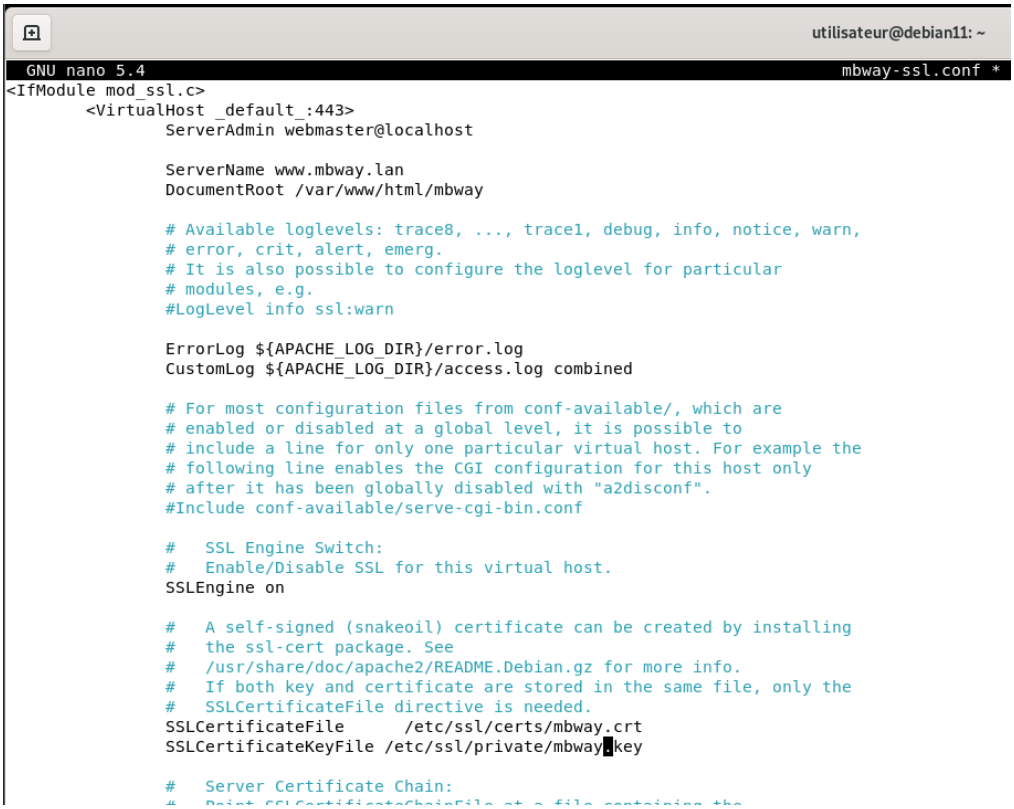
/etc/ssl/private/mbway.key -out /etc/ssl/certs/mbway.crt

Et pour digitalschool : sudo openssl req x509 -req -days 365 -in /etc/ssl/certs/digitalschool.csr -signkey /etc/ssl/private/digitalschool.key -out /etc/ssl/certs/digitalschool.crt

Nous devons ensuite modifier les lignes :

- SSLCertificateFile
- SSLCertificateKeyFile

Pour les sites Mbway et Digitaschool dans mbway-ssl.conf et digitalschool-ssl.conf



```
utilisateur@debian11: ~
 GNU nano 5.4                                    digitalschool-ssl.conf
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost

                ServerName www.digitalschool.lan
                DocumentRoot /var/www/html/digitalschool

                # 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
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                # For most configuration files from conf-available/, which are
                # enabled or disabled at a global level, it is possible to
                # include a line for only one particular virtual host. For example the
                # following line enables the CGI configuration for this host only
                # after it has been globally disabled with "a2disconf".
                #Include conf-available/serve-cgi-bin.conf

                #   SSL Engine Switch:
                #   Enable/Disable SSL for this virtual host.
                SSLEngine on

                #   A self-signed (snakeoil) certificate can be created by installing
                #   the ssl-cert package. See
                #   /usr/share/doc/apache2/README.Debian.gz for more info.
                #   If both key and certificate are stored in the same file, only the
                #   SSLCertificateFile directive is needed.
                SSLCertificateFile      /etc/ssl/certs/digitalschool.crt
                SSLCertificateKeyFile /etc/ssl/private/digitalschool.key

                #   Server Certificate Chain:
```

On va activer nos sites avec la commande a2ensite + le nom du virtual Host

Puis redemarrer apache2 : sudo systemctl restart apache2

Maintenant si on ajoute www.mbway.lan en alias à notre serveur dans le fichier /etc/hosts de notre client, vous pouvez voir la page d'accueil de notre site dans notre navigateur :

Nous arrivons donc bien a nous connecter en https !

Configuration du DNS :

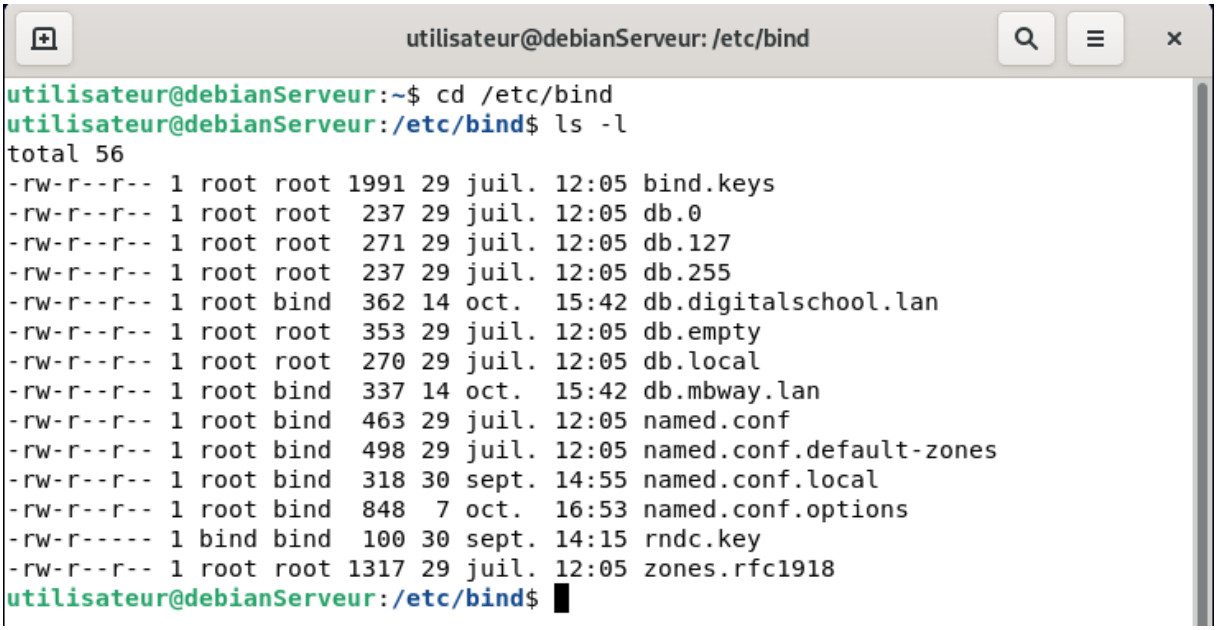
On commence par faire apt-get upgrade

Installation de BIND9 :

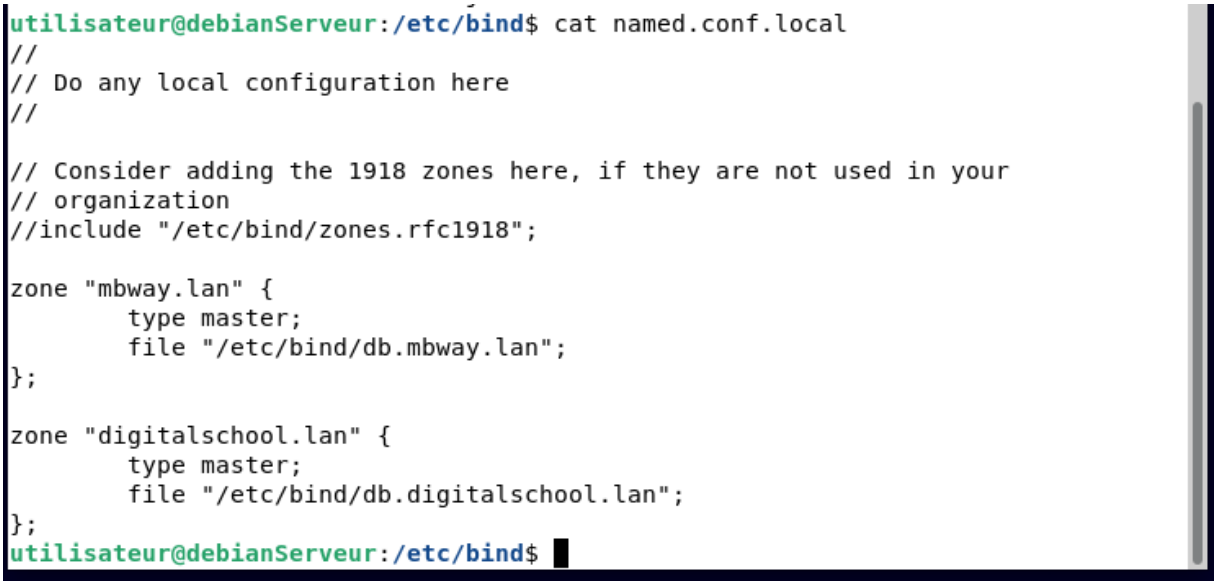
On fait ensuite sudo apt install Bind9 puis sudo apt install bind9utils.

Déclaration et création des zones :

On va commencer pas déclarer les zones dans /etc/bind



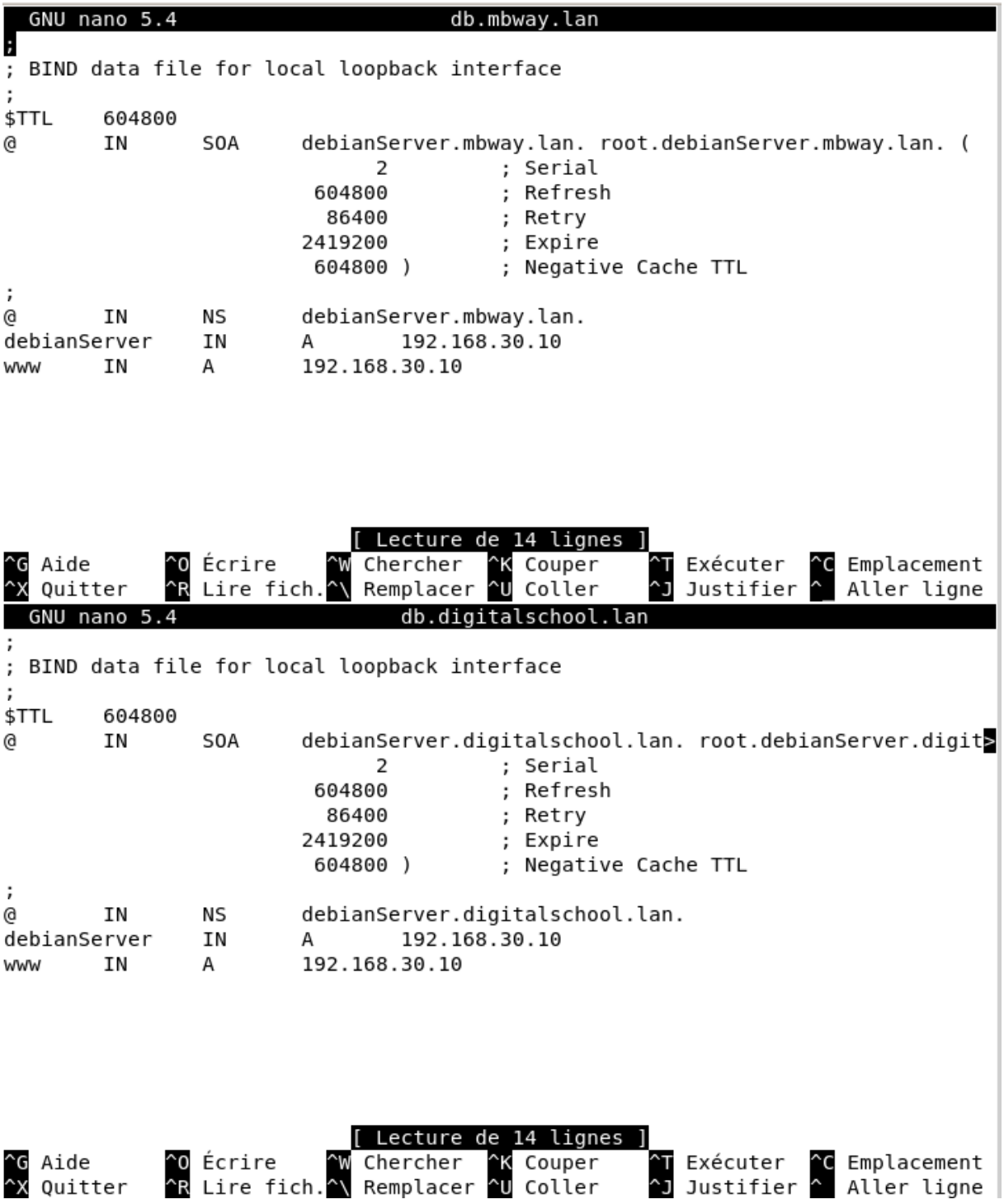
On va aller dans le fichiers named.conf.local et le configurer :



On le configure en mode master et ont créé deux zones pour chacun de nos sites et

on respecte la convention db.nomDomaine pour le déclarer.

On va ensuite créer nos fichiers de zones :



On lui donne les bons noms et adresse IP : debianRouteur.mbway.lan. et debianRouteur.digitalschool.lan.

On ajoute WWW pour avoir le www.mbway.lan et on voit également que l'on a un

enregistrement de type A.

On va également configurer le fichier named.conf.options avec le DNS de Google.

Les forwarders sont d'autres serveurs DNS vers lesquels le serveur DNS local peut envoyer les requêtes qu'il ne peut pas résoudre.

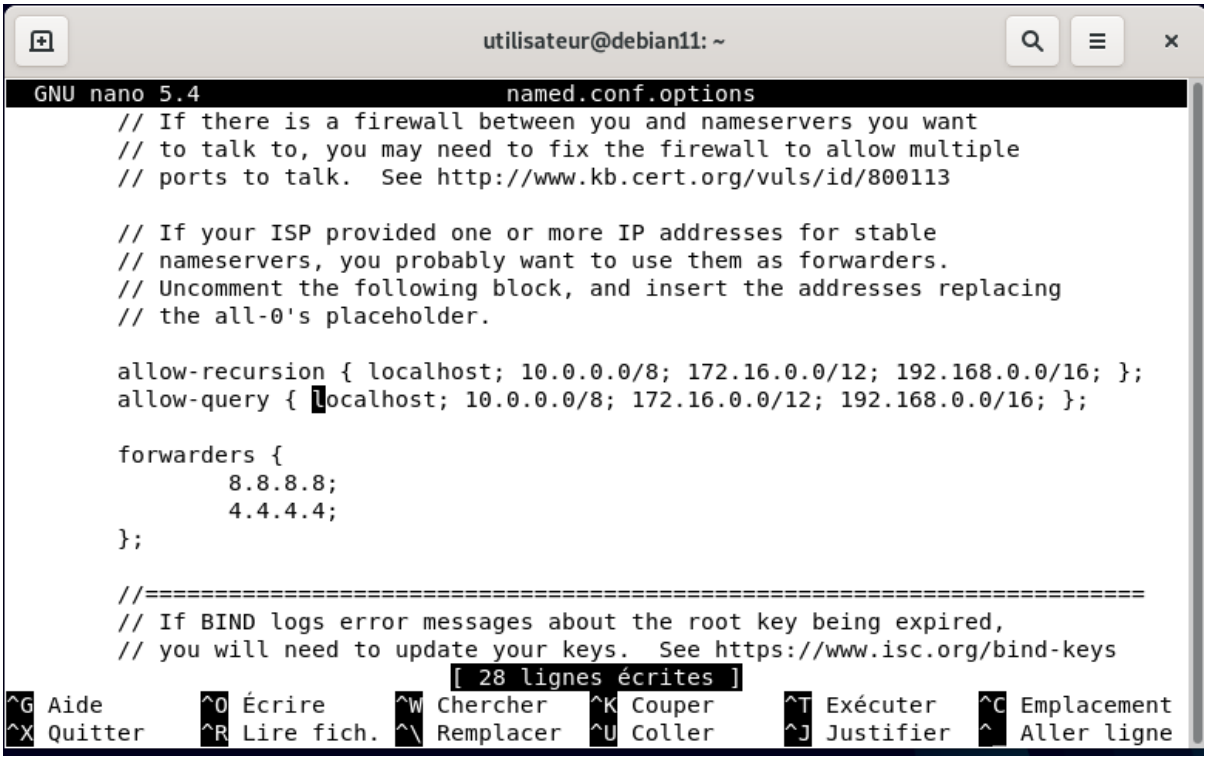
On doit aussi configurer Bind9 pour accepter toutes les requêtes provenant d'adresses privées définies par les standards RFC1918 (qui inclut les réseaux LAN).

Voici les plages d'adresses locales selon RFC1918 :

- 10.0.0.0/8 : Pour les réseaux de classe A.
- 172.16.0.0/12 : Pour les réseaux de classe B.
- 192.168.0.0/16 : Pour les réseaux de classe C.

Dans notre fichier named.conf.options, remplacez la directive allow-recursion par une liste incluant ces plages avec :

- allow-query : qui contrôle qui peut poser n'importe quelle question DNS (locale ou externe).
- allow-recursion : qui contrôle qui peut poser des questions récursives (requêtes nécessitant que le serveur DNS interroge d'autres serveurs).



On redémarre Bind9 afin que les modifications prennent effet : sudo systemctl restart bind9

Dans nos machines Debian on va devoir leur indiquer le nouveau DNS dans le fichier

Nano /etc/resolv.conf :

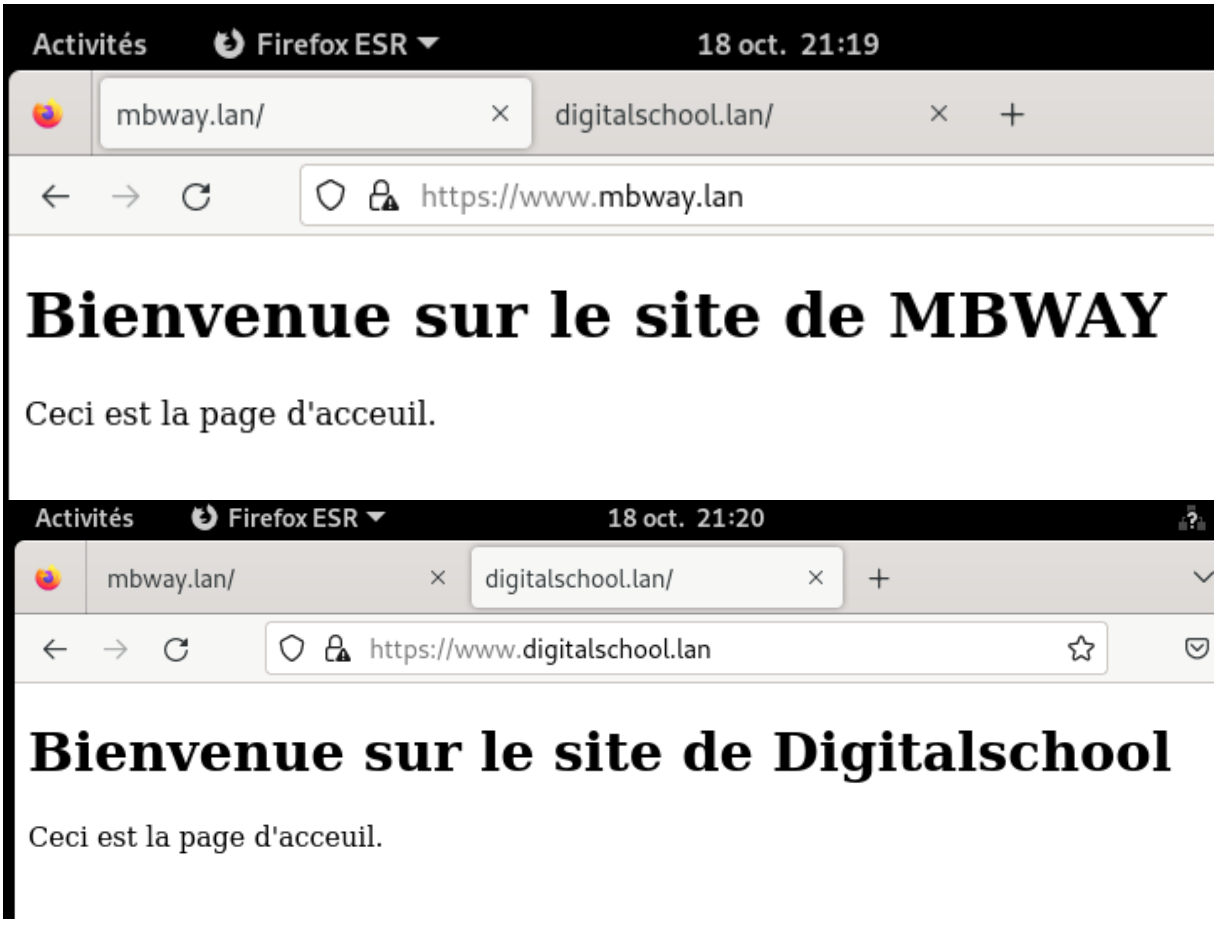
Nameserver : 192.168.30.10

On peut vérifier notre configuration grâce à la commande nslookup :

```
utilisateur@debianClient2:~$ nslookup www.mbway.lan
Server:         192.168.30.10
Address:        192.168.30.10#53

Name:   www.mbway.lan
Address: 192.168.30.10
```

On peut donc maintenant taper https://www.mbway.lan/

On teste la connexion :



Configuration du serveur DHCP :

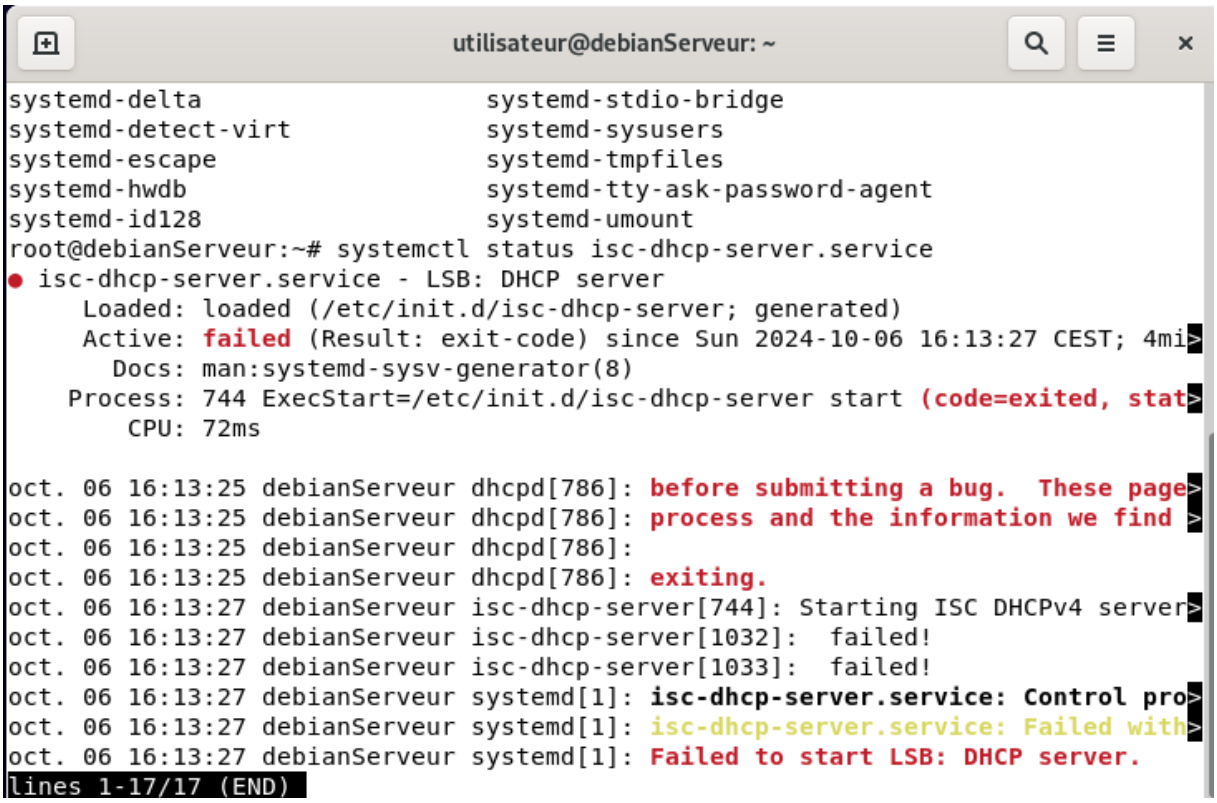
Afin de pouvoir attribuer des adresses IP automatiquement à tous nos VLAN, il faut que nous mettions en place un serveur DHCP.

Installation de ISC-DHCP-SERVER :

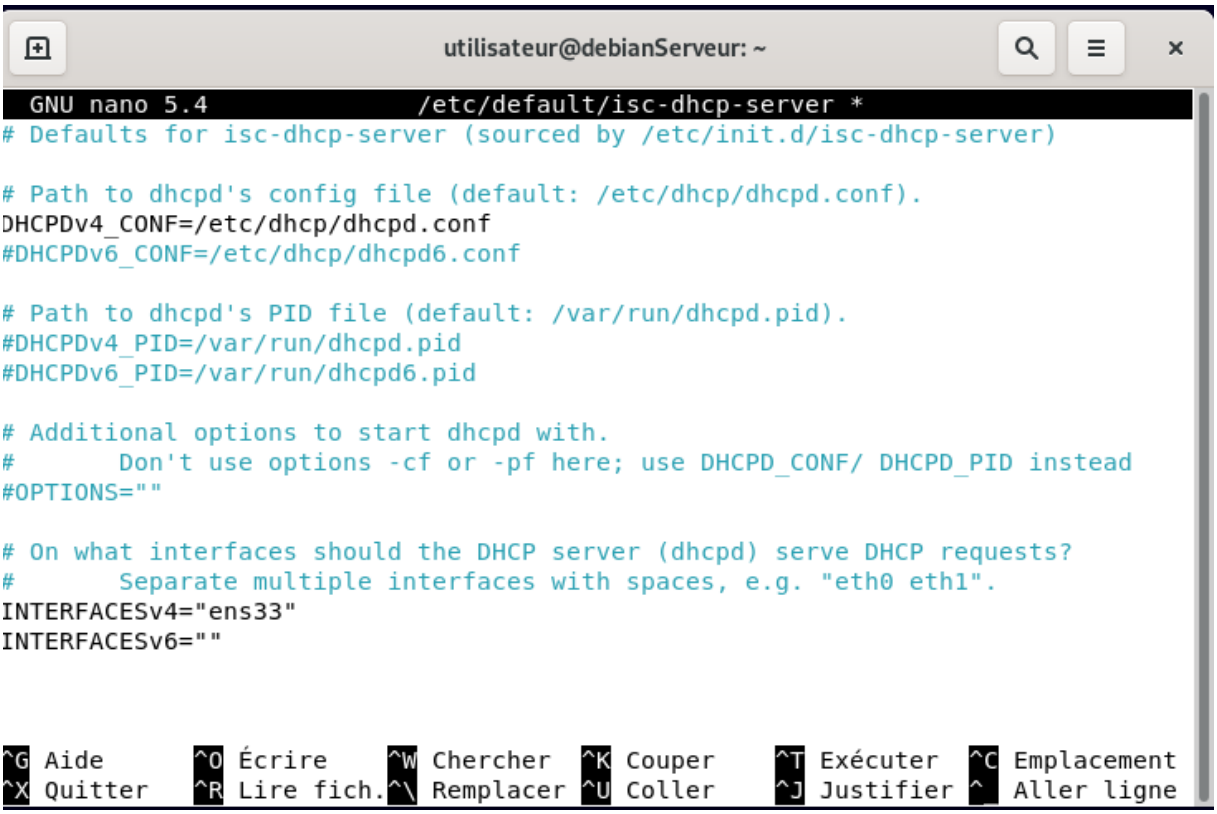
Pour cela il va falloir télécharger le paquet suivant sur notre machine Debian

- apt install isc-dhcp-server.

Une fois installé, il va tout d'abord falloir activer le serveur DHCP puis sélectionner notre interface réseau d'écoute. Cette sélection déterminera l'interface sur laquelle notre serveur recevra des requêtes DHCP et du coup enverra nos futurs pools :



On va donc configurer le fichier : nano /etc/default/isc-dhcp-server

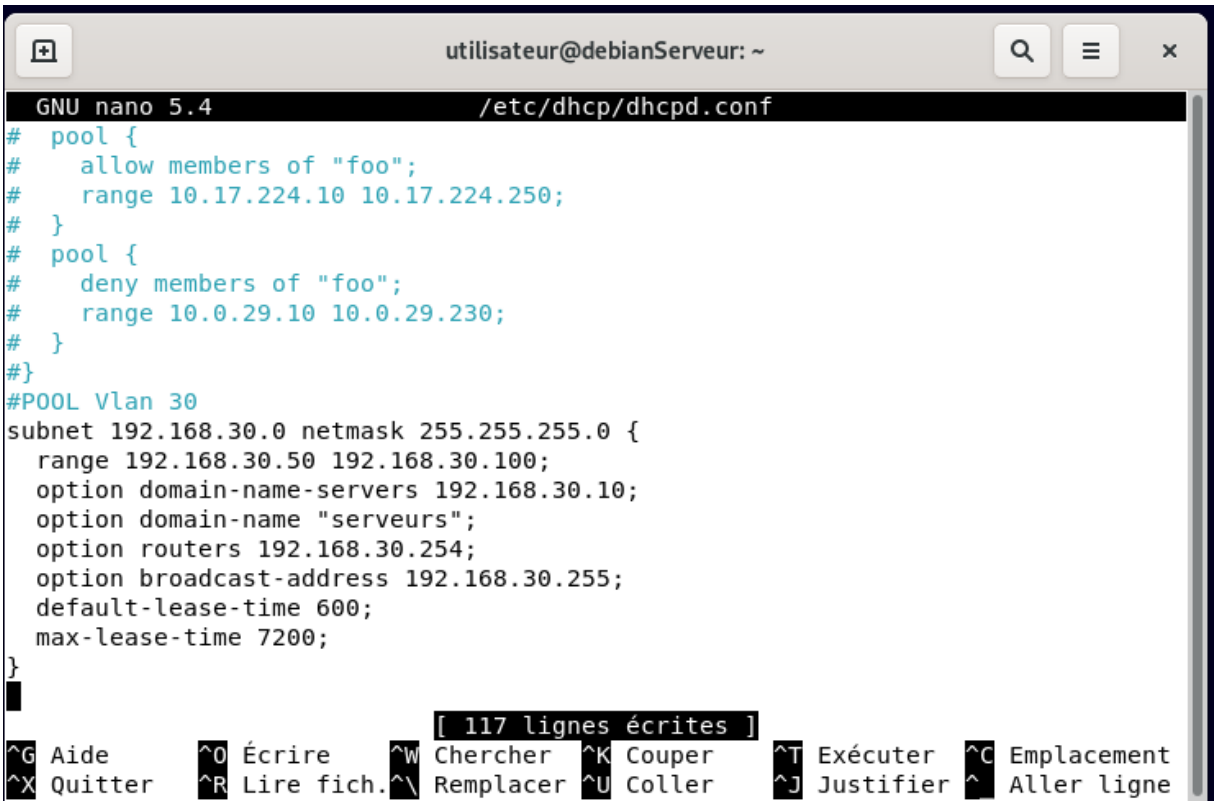


Notre serveur DHCP se réfèrera donc au fichier de configuration « dhcpd.conf ». De plus notre interface d'écoute sera la « ens33 ».

On redémarre le service : service isc-dhcp-server restart

Définitions des paramètres :

Maintenant, modifions le fichier dhcpd.conf qui se trouve dans le répertoire « isc-dhcp-sever » :



Définition des paramètres :

- subnet : le réseau

- range : la plage des adresses dynamiques

- option domain-name-servers : l'adresse du serveur DNS

- option domain-name ''serveurs'' : nom de notre domaine

- option routers : l'adresse de la passerelle

- option broadcast-address : l'adresse broadcaste du réseau

- default-lease-time : duré du bail en secondes

- max-lease-time : duré maximal du bail en secondes

En suivant la configuration le pool du VLAN 10 sera :

```
subnet 192.168.10.0 netmask 255.255.255.0 {
        range 192.168.10.50 192.168.10.100;
        option domain-name-servers 192.168.30.10 ;
        option domain-name « admnistration » ;
        option routers 192.168.10.254 ;
        option broadcast-address 192.168.10.255 ;
```

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

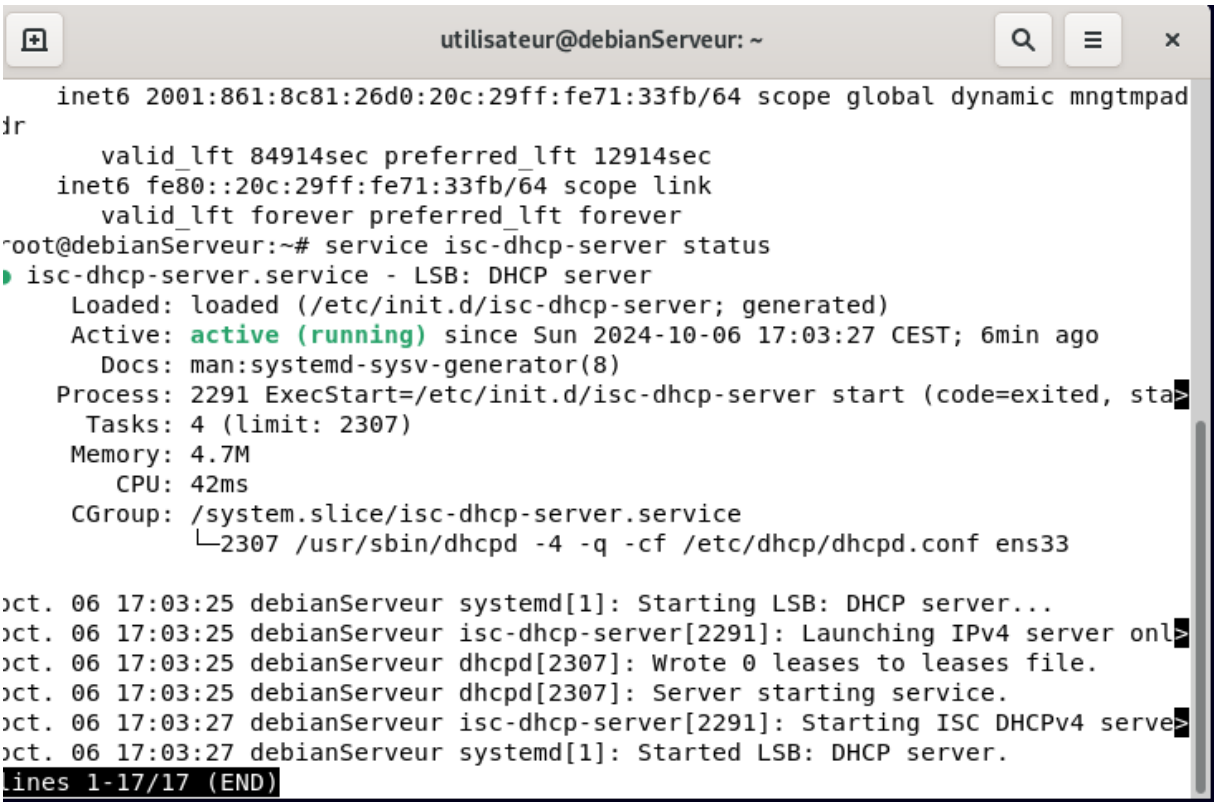
En suivant la configuration le pool du VLAN 20 sera :

```
subnet 192.168.20.0 netmask 255.255.255.0 {
        range 192.168.20.50 192.168.20.100;
        option domain-name-servers 192.168.30.10 ;
        option domain-name « formation » ;
        option routers 192.168.20.254 ;
        option broadcast-address 192.168.20.255 ;
        default-lease-time 600 ;
        max-lease-time 7200 ;
}
```

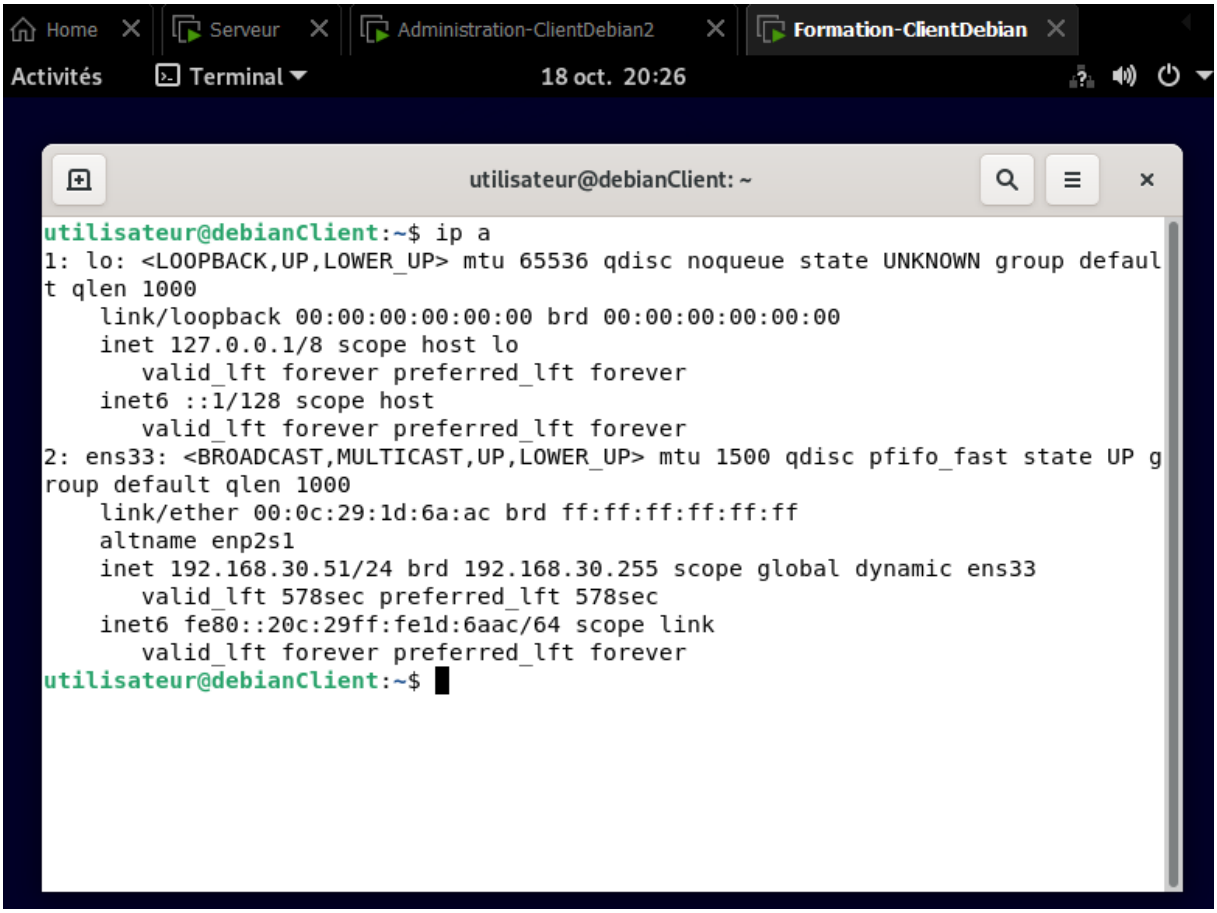
Et enfin en suivant la configuration, le pool du VLAN 30 sera :

```
subnet 192.168.30.0 netmask 255.255.255.0 {
          range 192.168.30.50 192.168.30.100;
          option domain-name-servers 192.168.30.10 ;
          option domain-name « serveur » ;
          option routers 192.168.30.254 ;
          option broadcast-address 192.168.30.255 ;
          default-lease-time 600 ;
          max-lease-time 7200 ;
}
```

On regarde si le service est maintenant actif :



On allume une nouvelle VM Debian et on regarde l'adresse qui lui est affecter :



Configuration du switch de niveau 2 (DELL 5548) :

Un switch manageable de niveau 2 ou 3 est indispensable pour pouvoir scinder notre parc informatique en plusieurs réseaux virtuels (Vlans).

Nous devons donc lui implémenter notre configuration en créant nos VLANS sur celui-ci et y relier chaque port à son VLAN de destination.

Pour cela nous allons nous connecter au Switch avec un câble console et lancer MobaXterm notre logiciel qui permettra d'obtenir une interface graphique de l'interface du switch sur notre PC hôte.

Une fois connecté nous allons dans un premier temps activer le SSH avec la commande suivante en mode « configure terminal » :

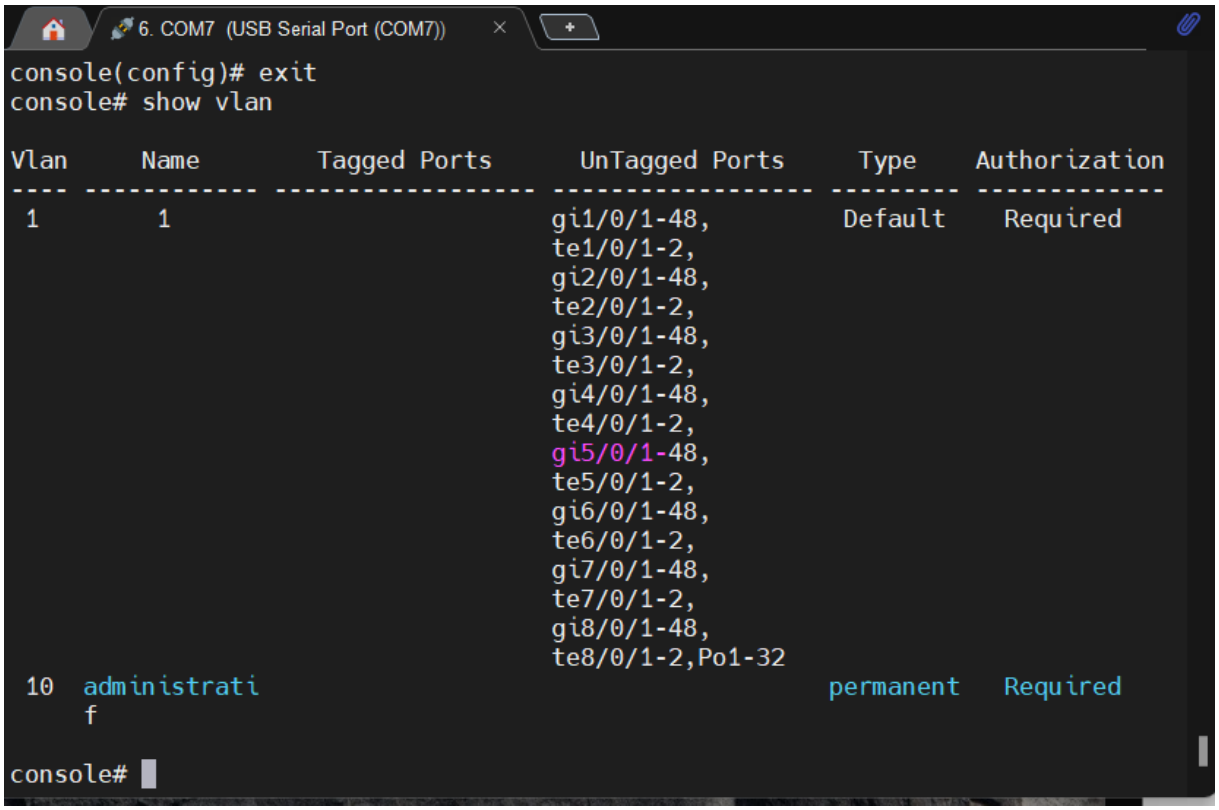
- crypto key generate rsa
- ip ssh server enable.

Une fois activé nous allons pouvoir affecter une adresse IP sur l'interface principal du switch : ici 192.168.254.254 255.255.255.0 et en Gateway : 192.168.254.253

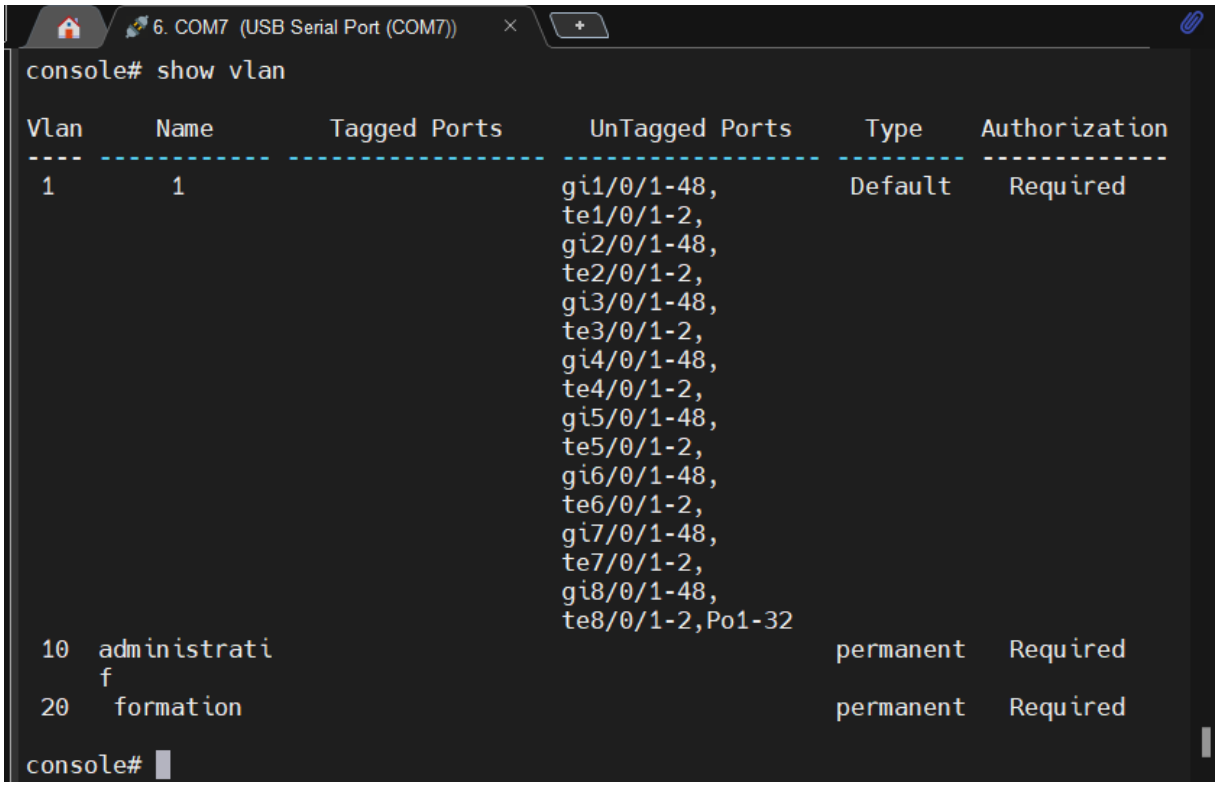
Maintenant nous pouvons commencer notre configuration.

Création de nos VLANs :

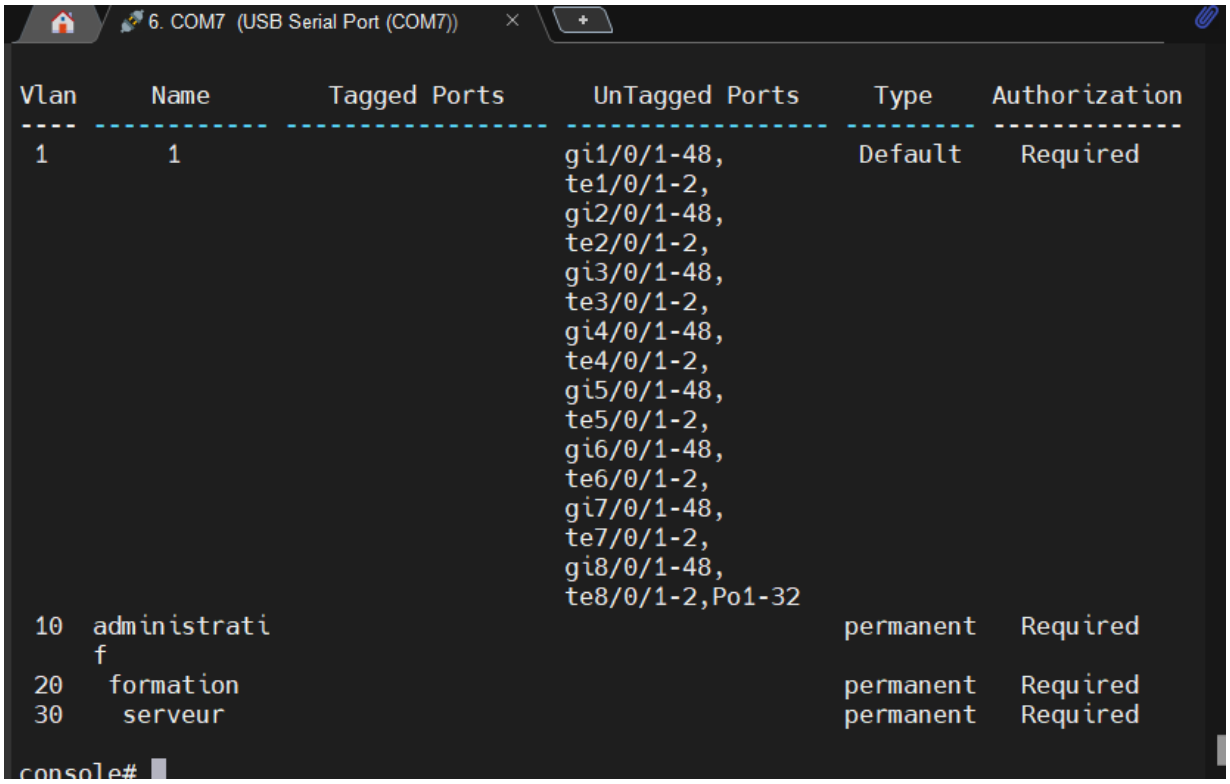
- enable
- configure terminal
- vlan10
- le vlan et créé on peut le voir avec show vlan
- conf
- interface vlan10
- name ADMINISTRATIF
- exit



- enable
- configure terminal
- vlan20
- le vlan et créé on peut le voir avec show vlan
- conf
- interface vlan20
- name Formation
- exit



- enable
- configure terminal
- vlan30
- le vlan et créé on peut le voir avec show vlan
- conf
- interface vlan30
- name Serveur
- exit



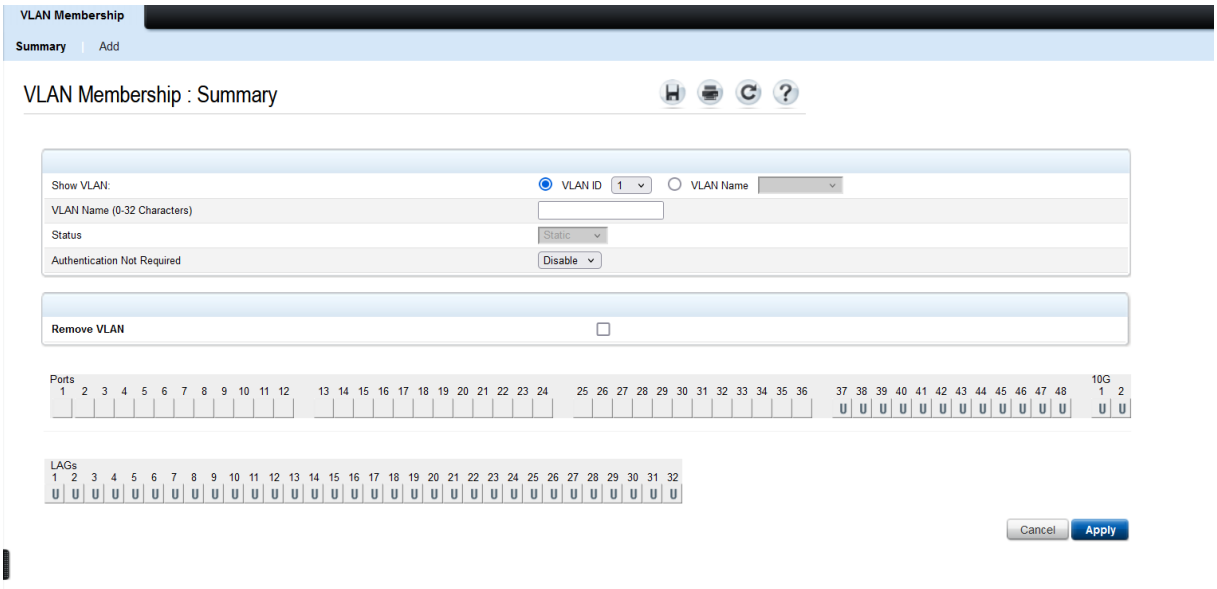
Attribution des ports à chaque VLAN (en CLI) :

- enable
- configure terminal
- interface range gi1/0/1-12
- switchport mode access
- switchport access vlan10
- exit
- interface range gi1/0/13-24
- switchport mode access
- switchport access vlan20
- exit
- interface range gi1/0/25-36
- switchport mode access
- switchport access vlan30
- exit
- interface gi1/0/48
- switchport mode trunk

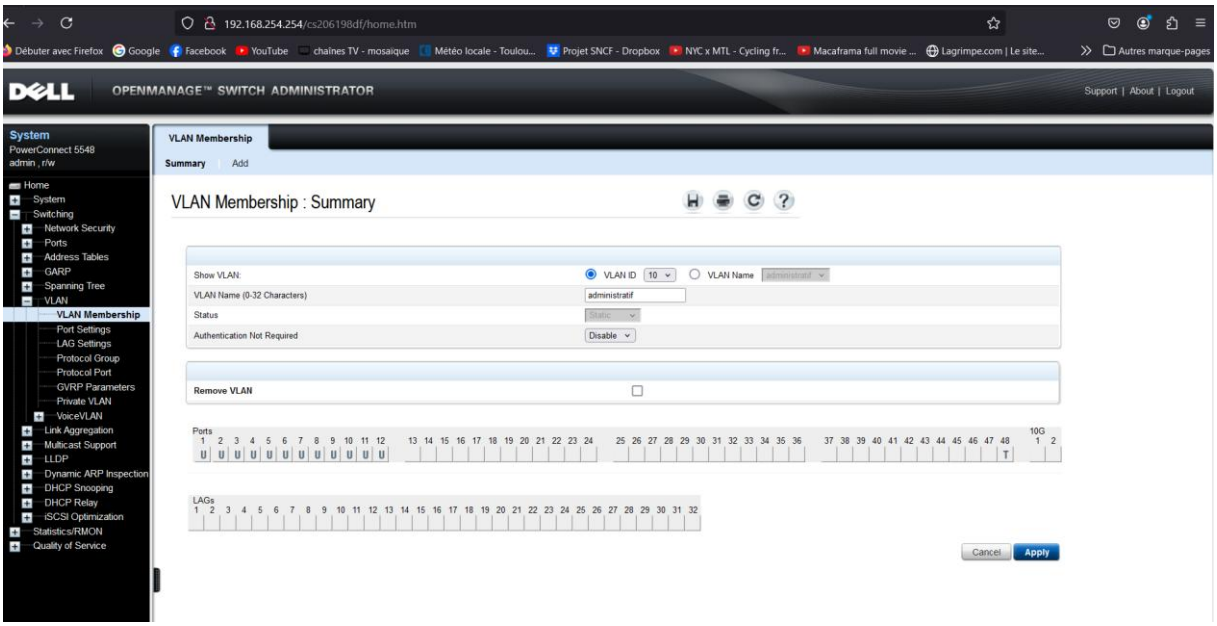
Voici ce que cela nous donne lorsqu'on fait un show running-config :

```
console# show vlan

Vlan     Name         Tagged Ports      UnTagged Ports     Type     Authorization
---- ------------ ------------------ ------------------ --------- -------------
 1        1                          gi1/0/37-48,        Default    Required
                                     te1/0/1-2,
                                     gi2/0/1-48,
                                     te2/0/1-2,
                                     gi3/0/1-48,
                                     te3/0/1-2,
                                     gi4/0/1-48,
                                     te4/0/1-2,
                                     gi5/0/1-48,
                                     te5/0/1-2,
                                     gi6/0/1-48,
                                     te6/0/1-2,
                                     gi7/0/1-48,
                                     te7/0/1-2,
                                     gi8/0/1-48,
                                     te8/0/1-2,Po1-32
 10   administrati      gi1/0/48          gi1/0/1-12     permanent   Required
      f
 20    formation        gi1/0/48         gi1/0/13-24     permanent   Required
 30     serveur         gi1/0/48         gi1/0/25-36     permanent   Required

console#
```

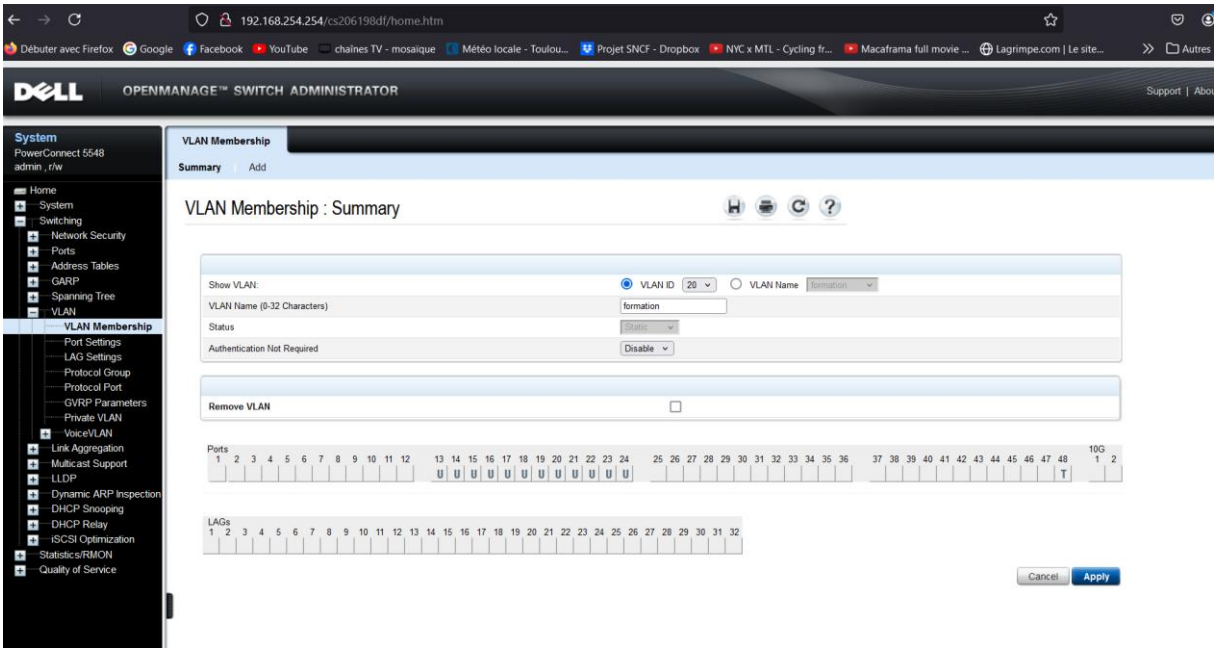
Connection avec l'interface HTTPS :



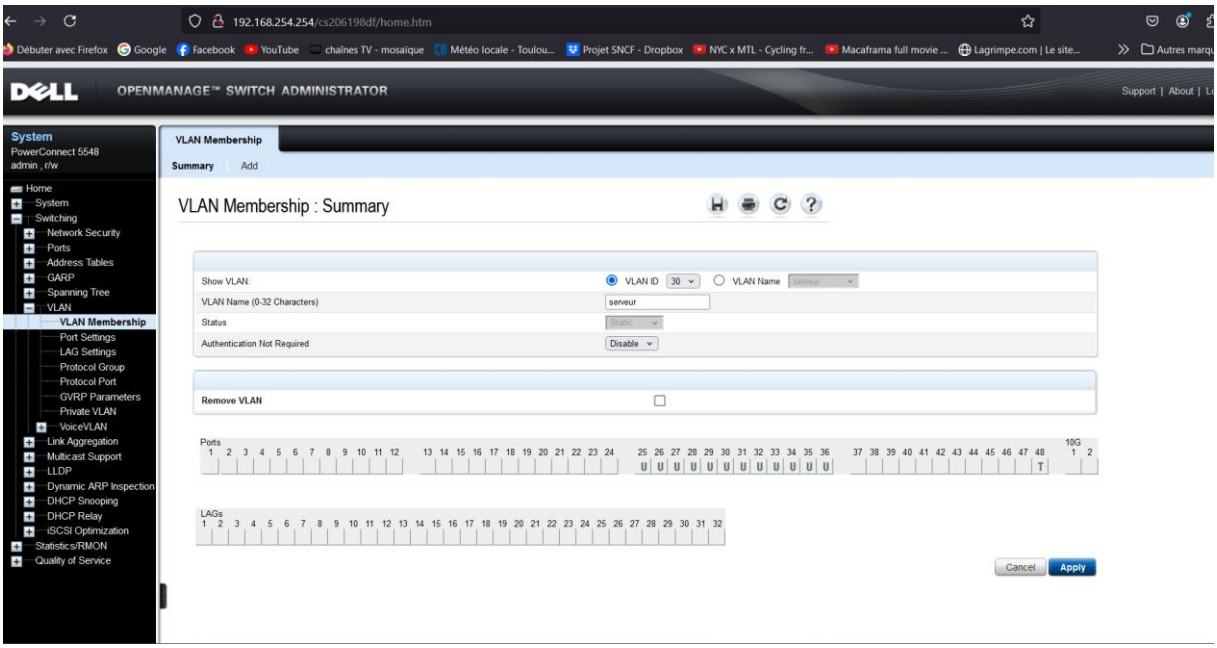
Pour le VLAN 10 :



Pour le VLAN 20 :



Pour le VLAN 30 :



Voilà, la configuration du switch est maintenant terminée, chaque port est attribué à un VLAN, la VLAN 1 étant la VLAN par défaut de notre switch, là où notre routeur va être branché.

Configuration du routeur Cisco (1900 séries) :

Notre routeur va jouer le rôle de passerelle entre les VLANS 10, 20 et 30.

Nous avons donc à notre disposition un routeur CISCO qui va être l'élément clé pour faire notre routage inter-vlan.

Comme pour le switch, il faut lancer MobaXterm pour pouvoir s'y connecter en Serial et ainsi le configurer.

On va dans un premier temps faire la configuration de base du routeur :

```
  Enter host name [Router]: routeurThomas

  The enable secret is a password used to protect access to
  privileged EXEC and configuration modes. This password, after
  entered, becomes encrypted in the configuration.
  Enter enable secret: root

  The enable password is used when you do not specify an
  enable secret password, with some older software versions, and
  some boot images.
  Enter enable password: root
% Please choose a password that is different from the enable secret
  Enter enable password: root

  The virtual terminal password is used to protect
  access to the router over a network interface.
  Enter virtual terminal password: root
  Configure SNMP Network Management? [yes]: yes
    Community string [public]:

Current interface summary


Any interface listed with OK? value "NO" does not have a valid configuration

Interface                  IP-Address      OK? Method Status                Proto
col
Embedded-Service-Engine0/0 unassigned      NO  unset  initializing          down

GigabitEthernet0/0         unassigned      NO  unset  down                  down

GigabitEthernet0/1         unassigned      NO  unset  down                  down


Enter interface name used to connect to the
management network from the above interface summary: utilisateur
Invalid interface


Enter interface name used to connect to the
management network from the above interface summary:
```

Voici la liste de mes commandes dans l'ordre chronologique pour configurer le routage inter-VLan :

```
enable
configure terminal
interface GigabitEthernet0/0.10
encapsulation Dot1Q 10
ip address 192.168.10.254 255.255.255.0
exit
interface GigabitEthernet0/0.20
encapsulation Dot1Q 20
ip address 192.168.20.254 255.255.255.0
exit
interface GigabitEthernet0/0.30
encapsulation Dot1Q 30
ip address 192.168.30.254 255.255.255.0
exit
```

Bien évidemment, pour que le routeur prenne en compte ma configuration il faut que j'active mes sous interfaces réseaux en activant l'interface réseau parente et il faut également brancher un câble au routeur :

- Ici, il s'agit de l'interface GigabitEthernet0/0

```
enable
configure terminal
interface GigabitEthernet0/0
no shutdown
```

Voici maintenant l'état de notre routeur après avoir effectué toutes mes commandes :

```
Router>en
Router#show ip int brief
Interface                  IP-Address      OK? Method Status                Proto
col
Embedded-Service-Engine0/0 unassigned      YES unset  administratively down down

GigabitEthernet0/0         unassigned      YES unset  up                    up

GigabitEthernet0/0.10      192.168.10.254  YES manual up                    up

GigabitEthernet0/0.20      192.168.20.254  YES manual up                    up

GigabitEthernet0/0.30      192.168.30.254  YES manual up                    up

GigabitEthernet0/1         unassigned      YES unset  administratively down down

Router#
```

On peut donc voir notre configuration :

```
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.10.0/24 is directly connected, GigabitEthernet0/0.10
L        192.168.10.254/32 is directly connected, GigabitEthernet0/0.10
      192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.20.0/24 is directly connected, GigabitEthernet0/0.20
L        192.168.20.254/32 is directly connected, GigabitEthernet0/0.20
      192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.30.0/24 is directly connected, GigabitEthernet0/0.30
L        192.168.30.254/32 is directly connected, GigabitEthernet0/0.30
Router#
```

Nous allons maintenant configurer les IP helper-address sur nos Vlan 10 et 20 afin qu'ils puissent accéder au DHCP :

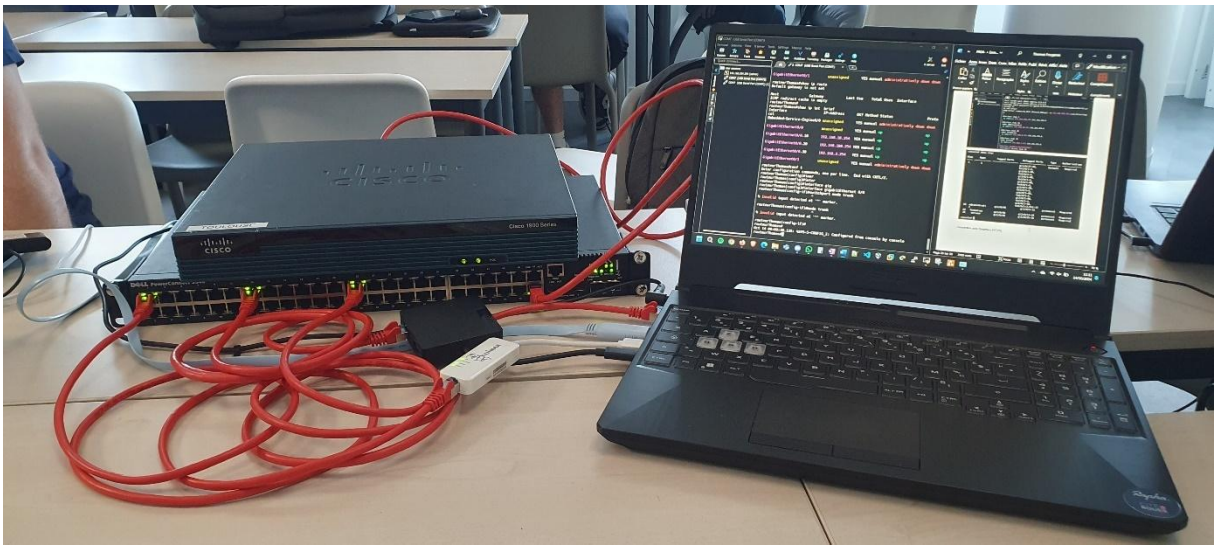
```
enable
configure terminal
interface GigabitEthernet0/0.10
ip helper-address 192.168.30.10
exit
interface GigabitEthernet0/0.20
ip helper-address 192.168.30.10
exit
```

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int
Router(config)#interface gi
Router(config)#interface gigabitEthernet 0/0.10
Router(config-subif)#ip he
Router(config-subif)#ip hell
Router(config-subif)#ip help
Router(config-subif)#ip helper-address 192.168.30.10
Router(config-subif)#exit
Router(config)#int
Router(config)#interface gi
Router(config)#interface gigabitEthernet 0/0.20
Router(config-subif)#ip help
Router(config-subif)#ip helper-address 192.168.30.10
Router(config-subif)#exit
Router(config)#exit
Router#ex
```

Pour l'instant nous ne configurons aucune Access-List donc par défaut tout le trafic est autorisé (ICMP / TCP / UDP etc…)

Configuration réseau des postes :

Voici donc la configuration physique finale de notre projet :

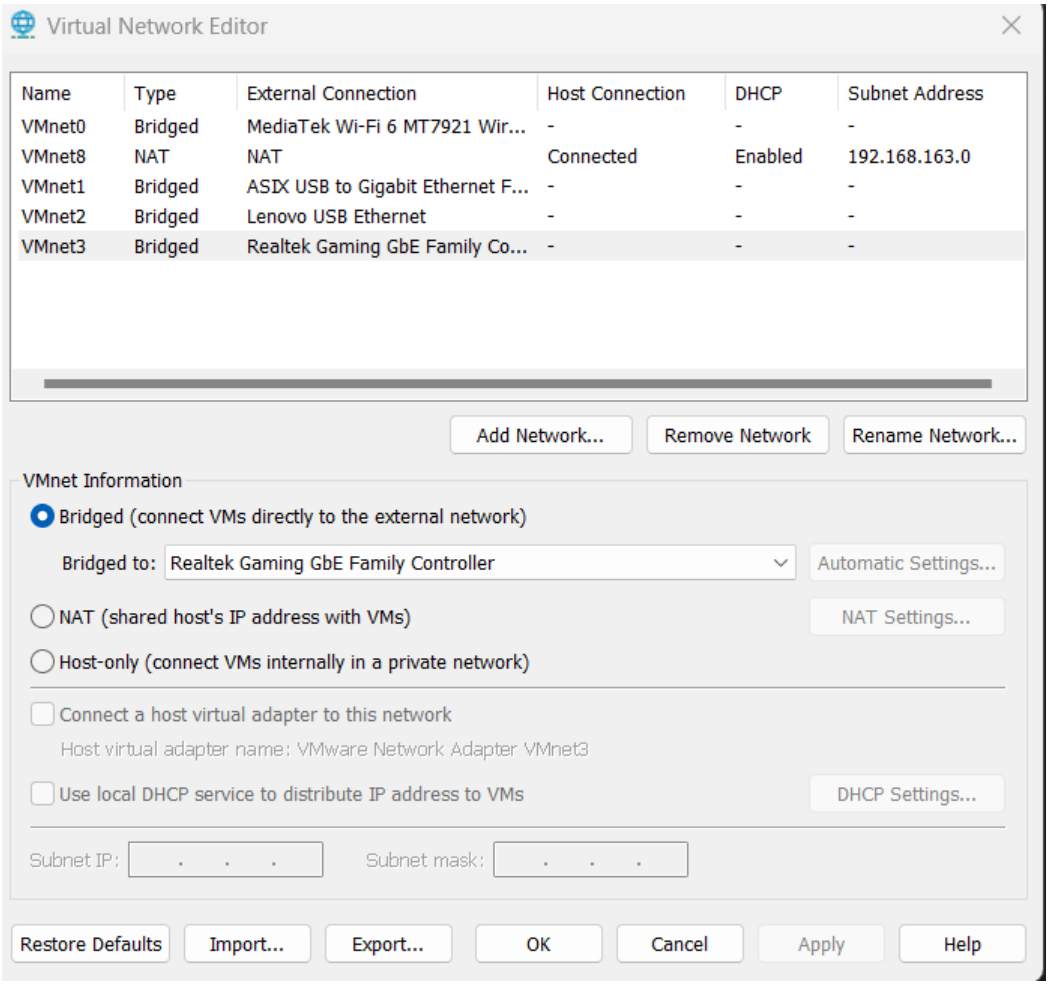


Dans un premier temps nous allons configurer nos cartes virtuelles pour chacune de nos VM sur VMware :

VMnet 1 : pour le Vlan 10 qui sera relier au port 10 du switch
VMnet 2 : pour le Vlan 20 qui sera relier au port 20 du switch
VMnet 3 : pour le Vlan 30 qui sera relier au port 30 du switch

Une autre possibilité est de :

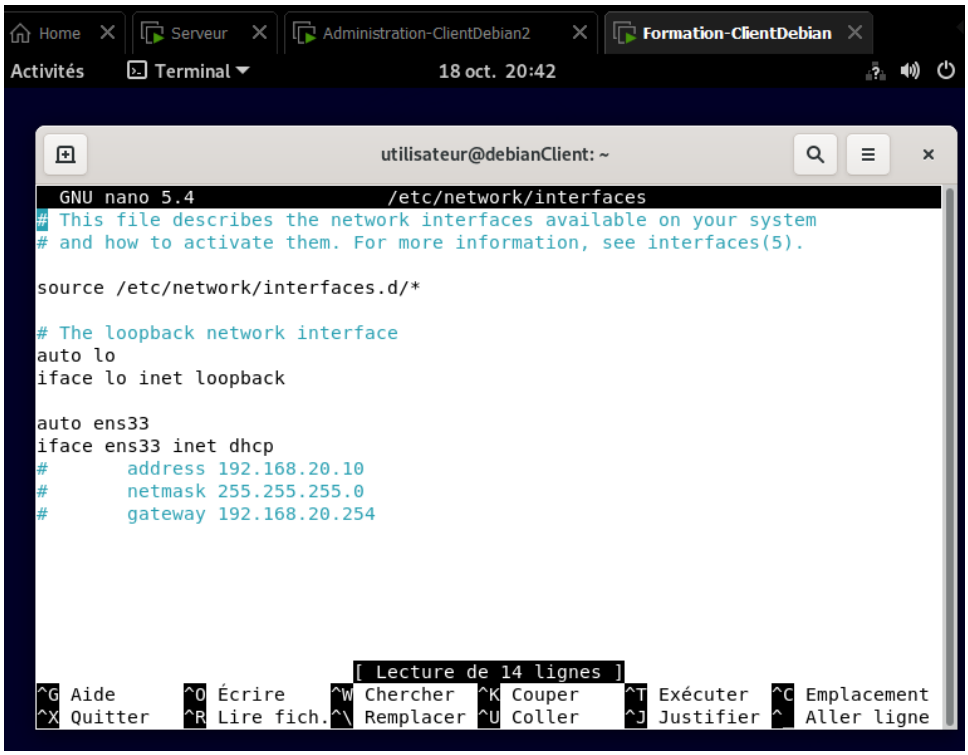
- Connecter une seule interface physique de l'ordinateur hôte sur un port en Trunk.
- Créer un commutateur virtuel basé sur cette interface et utiliser celui-ci comme support de connexion des cartes réseaux des VM
- Puis, indiquer dans les paramètres de carte réseau des différentes VM, le vlan d'appartenance.

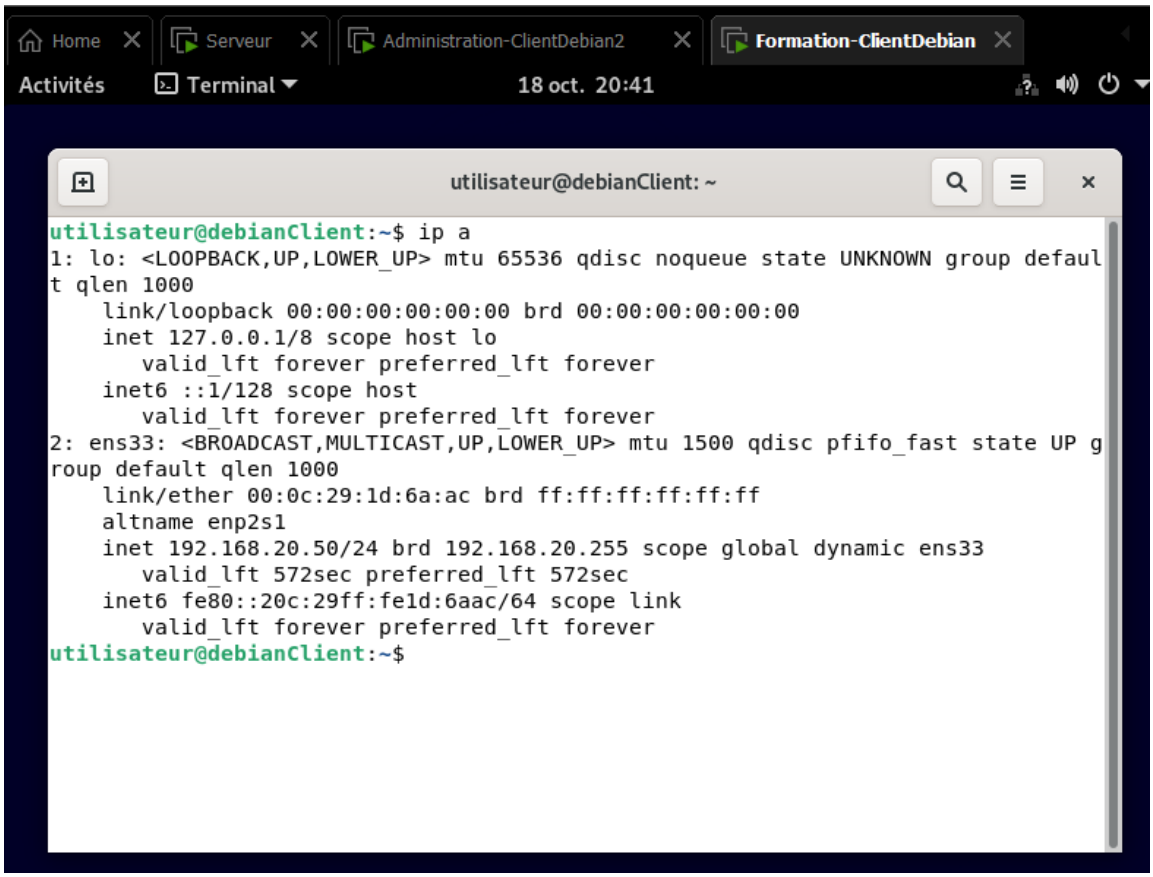


Nous allons maintenant effectuer la configuration réseaux du poste administrateur et formation :

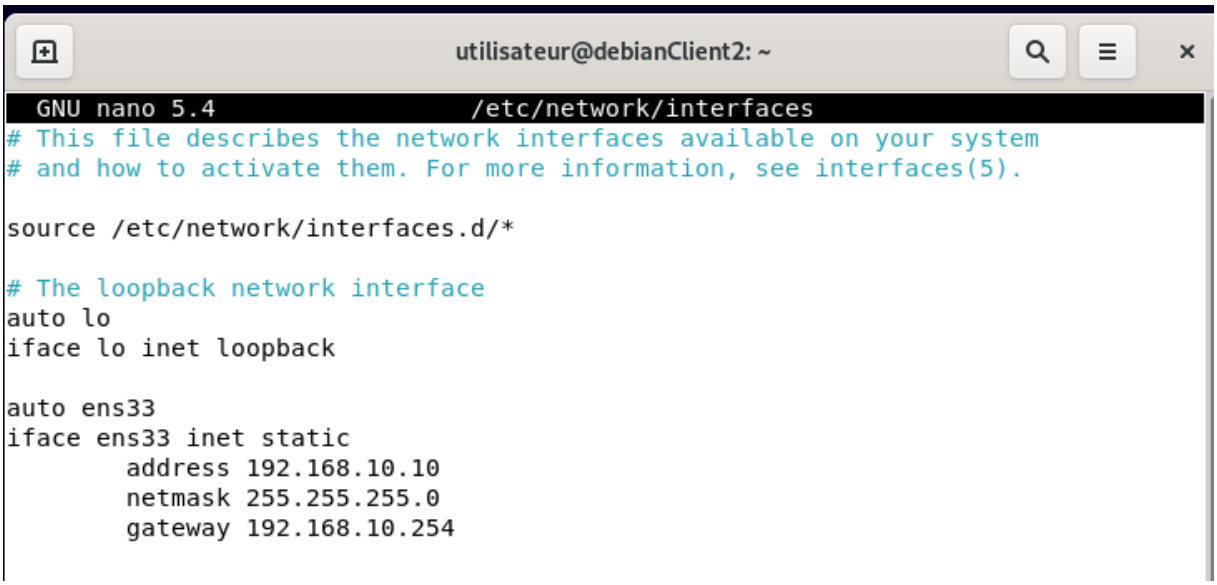
Poste présent dans le réseau formation :

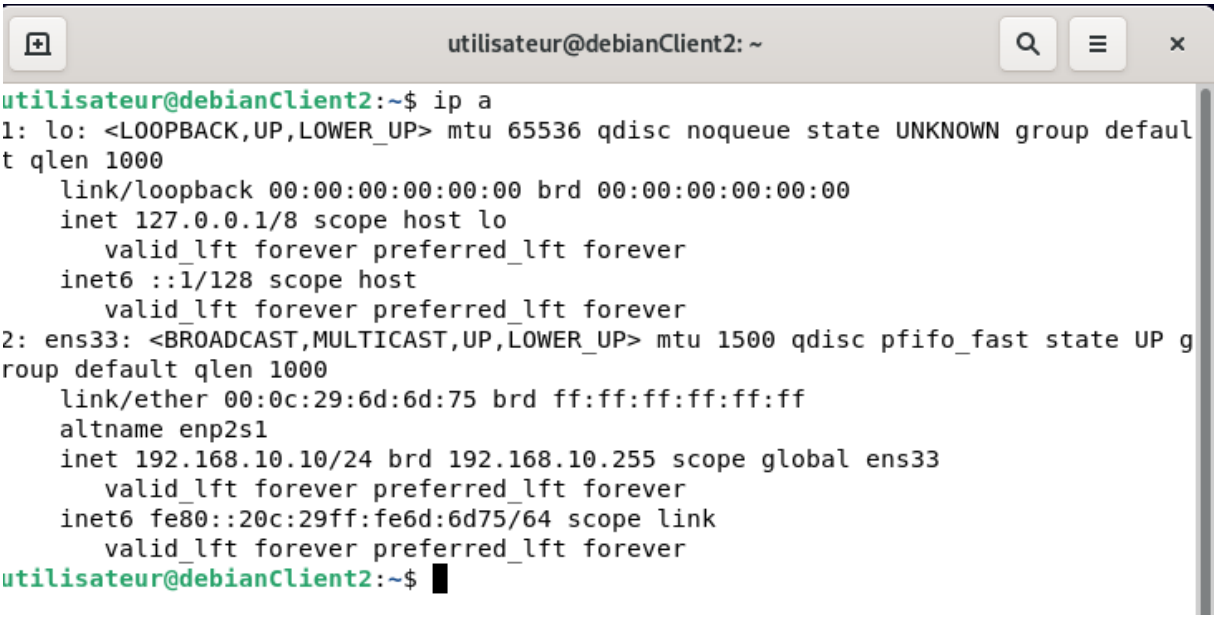
Nous allons configurer son interface en mode DHCP et l’on voit qu’il a bien reçu les informations de notre DHCP :


Home
Serveur
Administration-ClientDebian2
Formation-ClientDebian
Activités
Terminal
18 oct. 20:42
utilisateur@debianClient: ~
GNU nano 5.4 /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 ens33
iface ens33 inet dhcp
address 192.168.20.10
netmask 255.255.255.0
gateway 192.168.20.254
[Lecture de 14 lignes]
^G Aide ^O Écrire ^W Chercher ^K Couper ^T Exécuter ^C Emplacement
^X Quitter ^R Lire fich. ^\ Remplacer ^U Coller ^J Justifier ^_ Aller ligne


Home
Serveur
Administration-ClientDebian2
Formation-ClientDebian
Activités
Terminal
18 oct. 20:41
utilisateur@debianClient: ~
utilisateur@debianClient:~$ 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
link/ether 00:0c:29:1d:6a:ac brd ff:ff:ff:ff:ff:ff
altname enp2s1
inet 192.168.20.50/24 brd 192.168.20.255 scope global dynamic ens33
valid_lft 572sec preferred_lft 572sec
inet6 fe80::20c:29ff:fe1d:6aac/64 scope link
valid_lft forever preferred_lft forever
utilisateur@debianClient:~$

Poste de l'administrateur qui lui sera en IP STATIC afin de facilité les règles de filtrage :





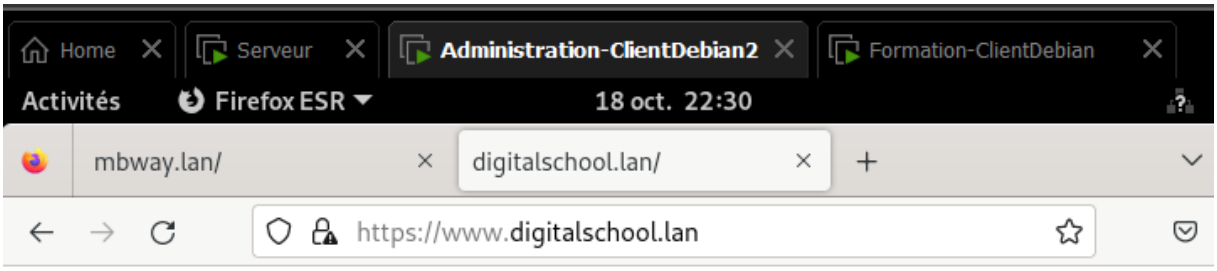
Test de connexion :

Depuis le Vlan 10 192.168.10.0/24 avec le poste administrateur 192.168.10.10 :

Ping vers le serveur et le poste formation :

```
utilisateur@debianClient2:~$ ping 192.168.20.50
PING 192.168.20.50 (192.168.20.50) 56(84) bytes of data.
64 bytes from 192.168.20.50: icmp_seq=1 ttl=63 time=2.05 ms
^C
--- 192.168.20.50 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.048/2.048/2.048/0.000 ms
utilisateur@debianClient2:~$ ping 192.168.30.10
PING 192.168.30.10 (192.168.30.10) 56(84) bytes of data.
64 bytes from 192.168.30.10: icmp_seq=1 ttl=63 time=1.70 ms
^C
--- 192.168.30.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.704/1.704/1.704/0.000 ms
utilisateur@debianClient2:~$
```

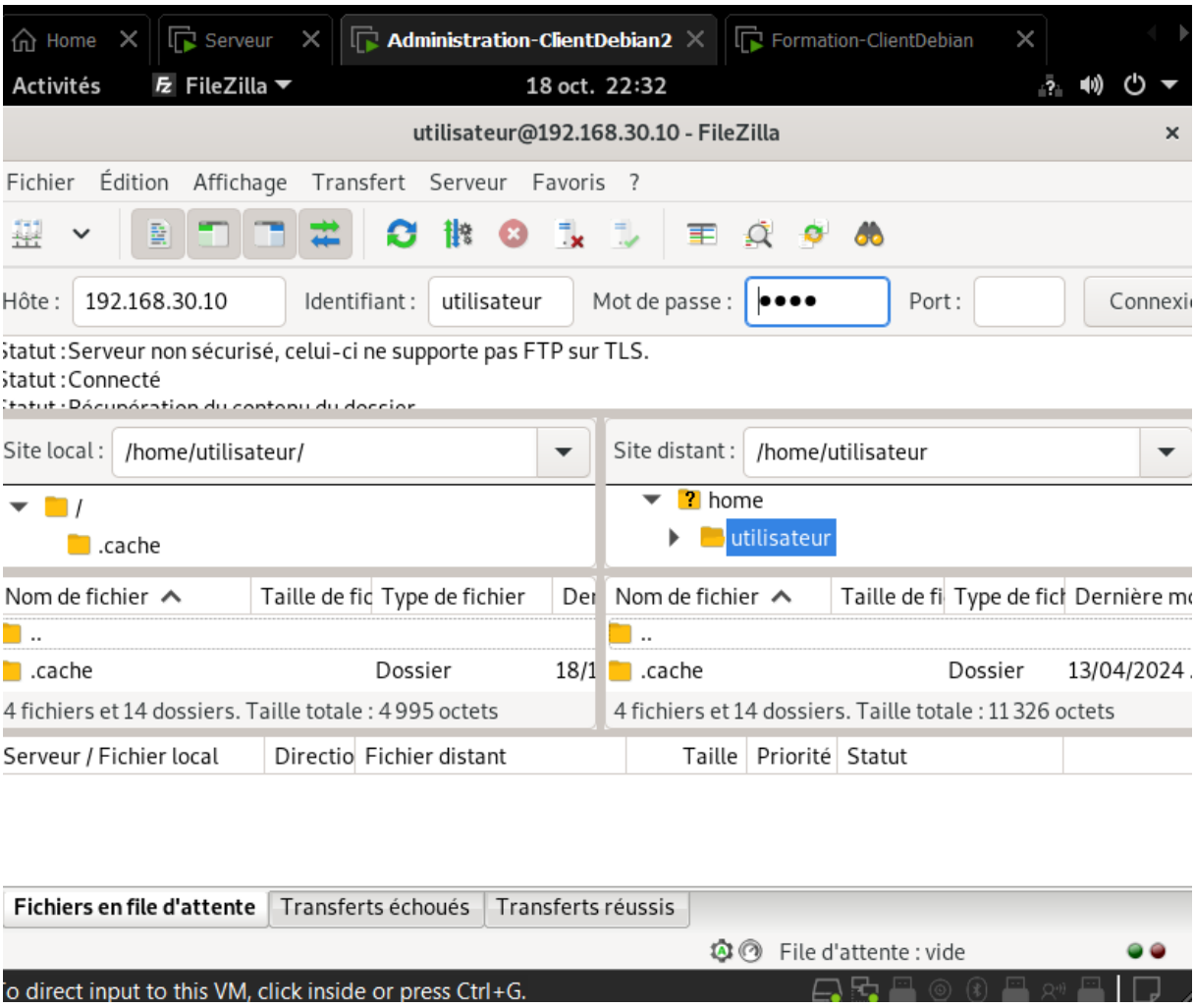
Accès HTTPS :



Bienvenue sur le site de Digitalschool

Ceci est la page d'acceuil.

Accès FTP :

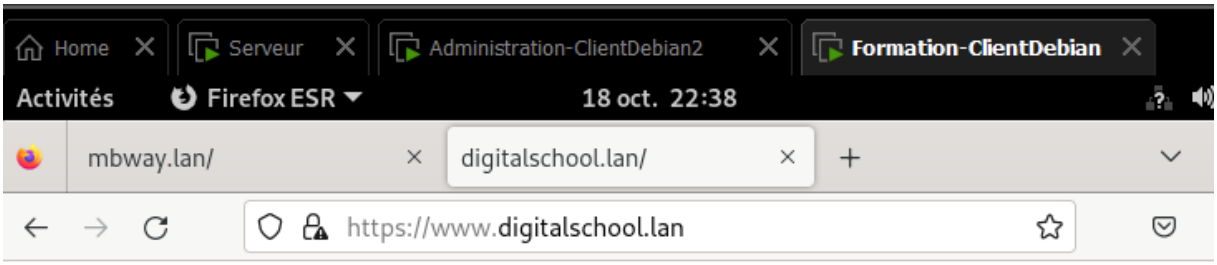


Depuis le Vlan 20 192.168.20.0/24 avec le poste formation 192.168.20.50 :

Ping vers le serveur et le poste administration :

```
utilisateur@debianClient:~$ ping 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
64 bytes from 192.168.10.10: icmp_seq=1 ttl=63 time=1.20 ms
^C
--- 192.168.10.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.200/1.200/1.200/0.000 ms
utilisateur@debianClient:~$ ping 192.168.30.10
PING 192.168.30.10 (192.168.30.10) 56(84) bytes of data.
64 bytes from 192.168.30.10: icmp_seq=1 ttl=63 time=2.18 ms
^C
--- 192.168.30.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.183/2.183/2.183/0.000 ms
utilisateur@debianClient:~$
```

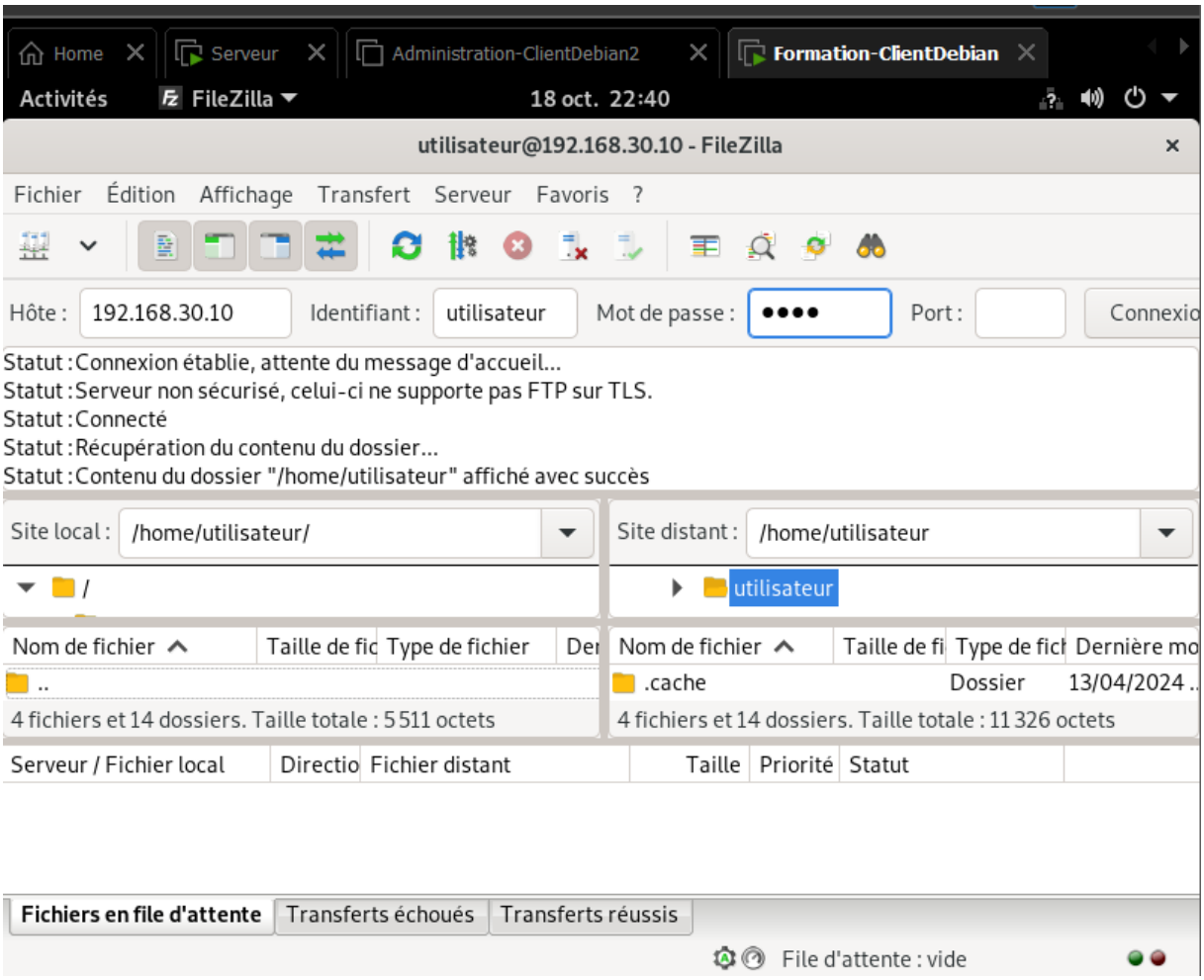
Acces HTTPS :



Bienvenue sur le site de Digitalschool

Ceci est la page d'acceuil.

Accès FTP :



Filtrage du trafic par ACL :

VLAN 10 Réseau administrateur : ens34 192.168.10.0/24 IP debian 192.168.10.10/24

VLAN 20 Réseau formation : ens36 192.168.20.0/24 IP debian 192.168.20.50/24

VLAN 30 Réseau serveur : ens35 192.168.30.0/24 IP serveur 192.168.30.10/24

Voici la liste de nos ACL présentes sur notre routeur :

Le serveur Web hébergera deux sites (www.mbway.lan et www.digitalschol.lan) accessibles à tous les utilisateurs :

Nous devons d'abord créer la règle ACL autorisant la connexion au DNS :

- ip access-list extended vlan10
- Permit udp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 eq 53

Ensuite :

- permit tcp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 eq 443

Idem pour le Vlan20 :

- ip access-list extended vlan20
- Permit udp 192.168.20.0 0.0.0.255 192.168.30.0 0.0.0.255 eq 53
- permit tcp 192.168.20.0 0.0.0.255 192.168.30.0 0.0.0.255 eq 443

Le serveur FTP sera accessible uniquement depuis le poste des développeurs, situé dans le réseau administratif :

- ip access-list extended vlan10
- permit tcp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 eq 21
- permit tcp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 gt 1021

Et pour SFTP:

- permit tcp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 eq 22

Les salles de formation ne peuvent pas accéder au sous-réseau administratif mais le poste de l'administrateur peut accéder aux postes et équipements des salles de formation :

- ip access-list extended vlan10
- permit icmp 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255 echo

- ip access-list extended vlan20
- permit icmp any any echo-reply

Tous les sous-réseaux reçoivent leurs paramètres TCP/IP d'un serveur DHCP commun situé dans le réseau dédié aux serveurs :

- ip access-list extended vlan10
- permit udp any any eq 67
- permit udp any any eq 68

- ip access-list extended vlan20
- permit udp any any eq 67
- permit udp any any eq 68

Le Vlan 30 étant notre Vlan Serveur aucune ACL n'est nécessaire.

Chaque ACL devra être placée sur son interface en « IN » pour que le trafic soit bel et bien analysé puis traité

Nous allons maintenant appliquer nos ACL sur les différentes interfaces :

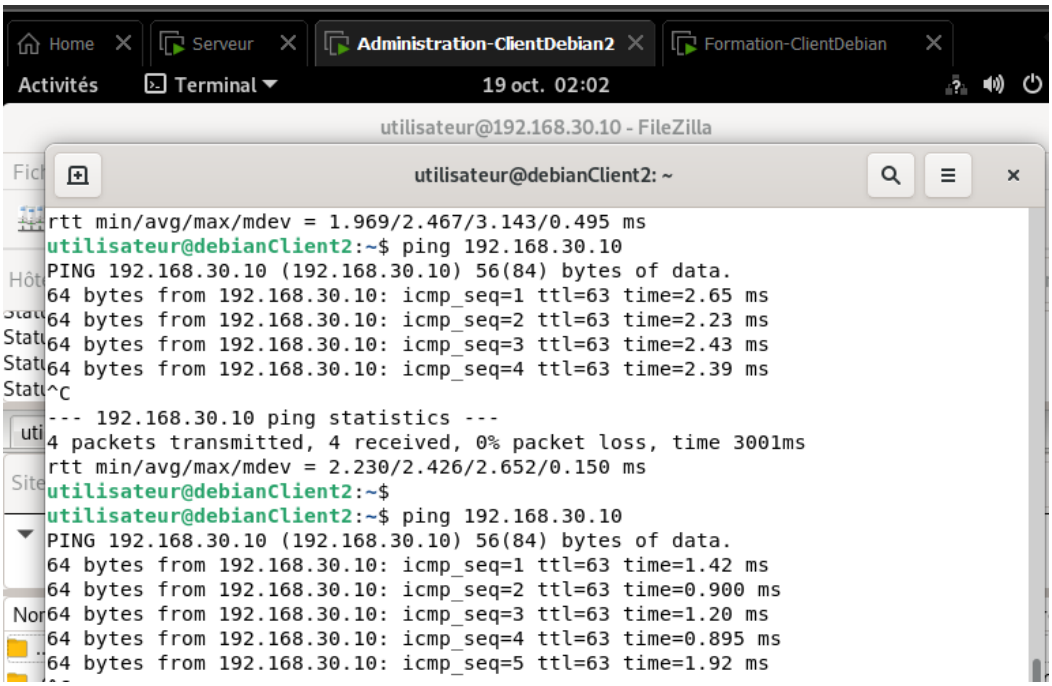
- interface GigabitEthernet0/0.10
- ip access-group vlan10 in
- exit
- interface GigabitEthernet0/0.20
- ip access-group vlan20 in

```
Router#sh access-lists
Extended IP access list vlan10
    10 permit udp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 eq domain (187 matches)
    20 permit tcp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 eq 443
    30 permit tcp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 eq ftp
    40 permit tcp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 gt 1021
    50 permit icmp 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255 echo
    60 permit udp any any eq bootps (62 matches)
    70 permit udp any any eq bootpc
    80 permit icmp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 (12 matches)
    90 permit tcp 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 eq 22 (25 matches)
Extended IP access list vlan20
    10 permit udp 192.168.20.0 0.0.0.255 192.168.30.0 0.0.0.255 eq domain (106 matches)
    20 permit tcp 192.168.20.0 0.0.0.255 192.168.30.0 0.0.0.255 eq 443
    30 permit icmp any any echo-reply
    40 permit udp any any eq bootps (61 matches)
    50 permit udp any any eq bootpc
```

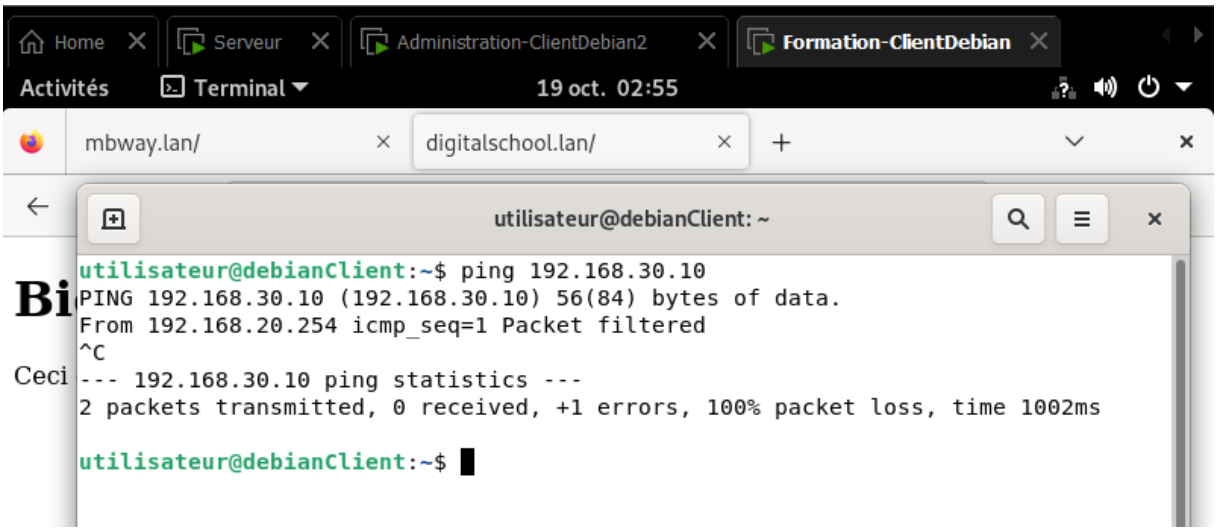
Phase d’expérimentation et de test du parc informatique :

Test de connexion avec notre Serveur :

Depuis le poste administrateur :

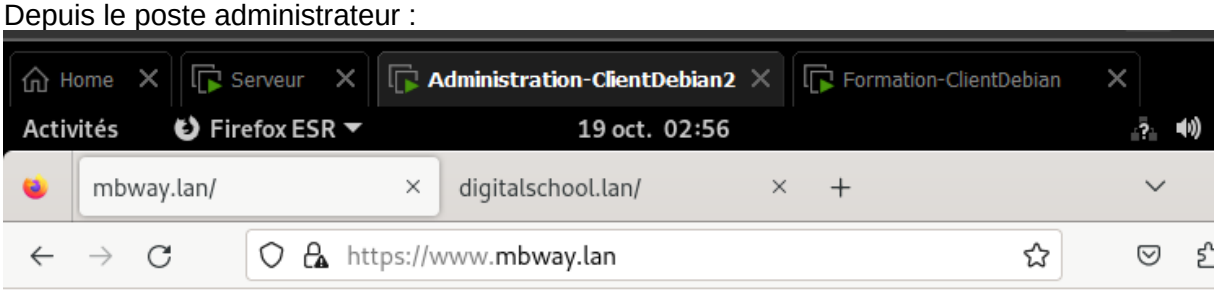


Depuis le poste formation :



Accès Intranet :

Depuis le poste administrateur :



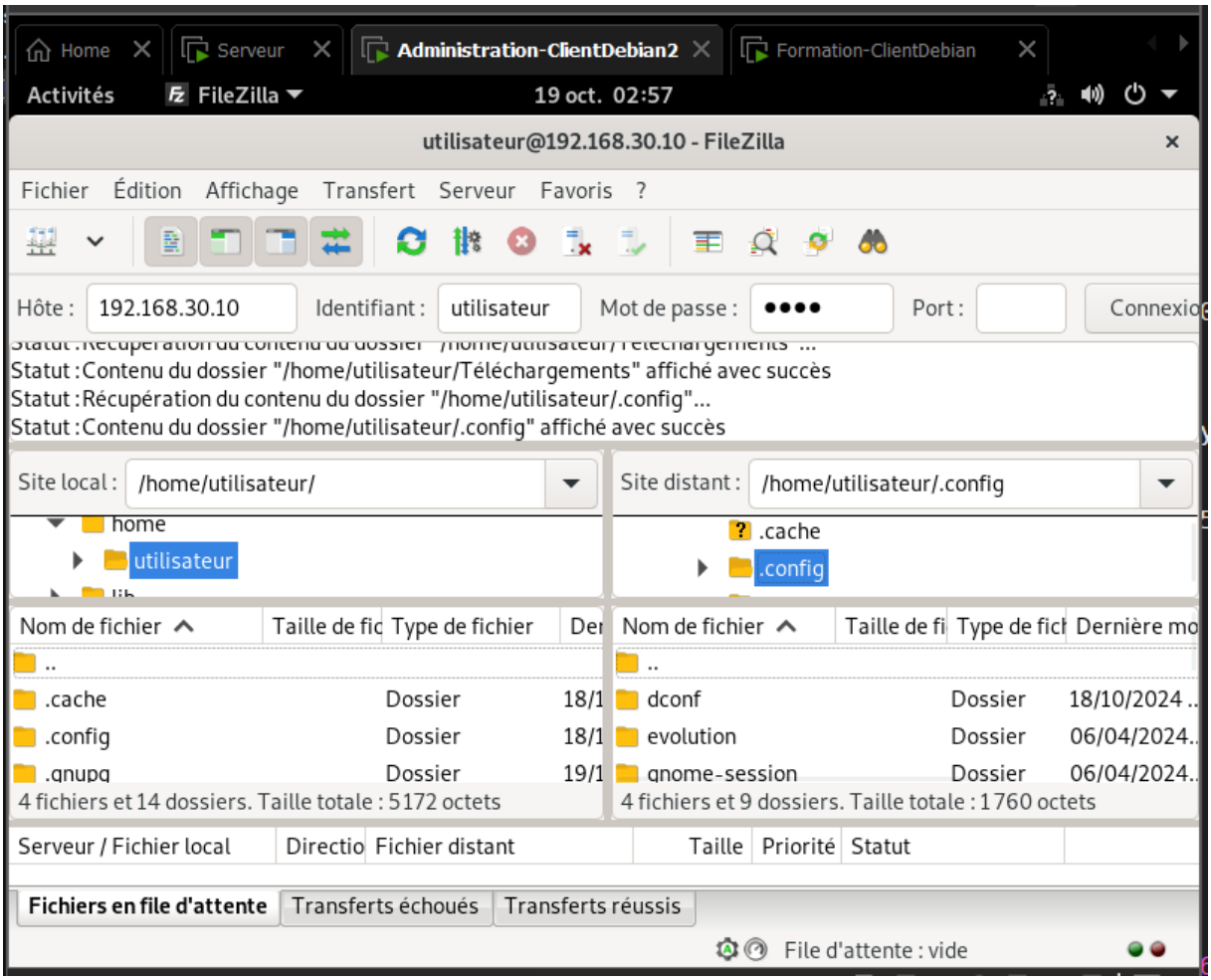
Bienvenue sur le site de MBWAY

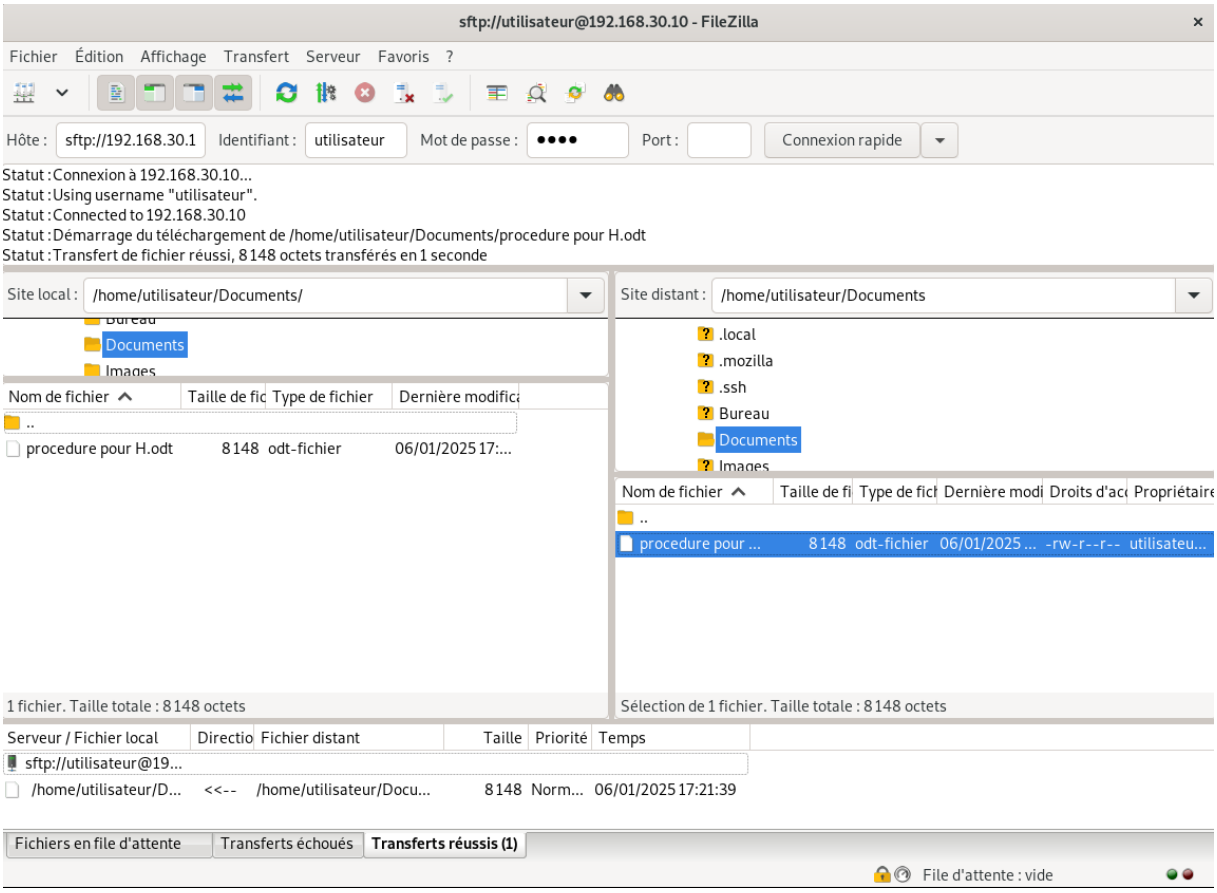
Ceci est la page d'acceuil.

Depuis le poste formation :

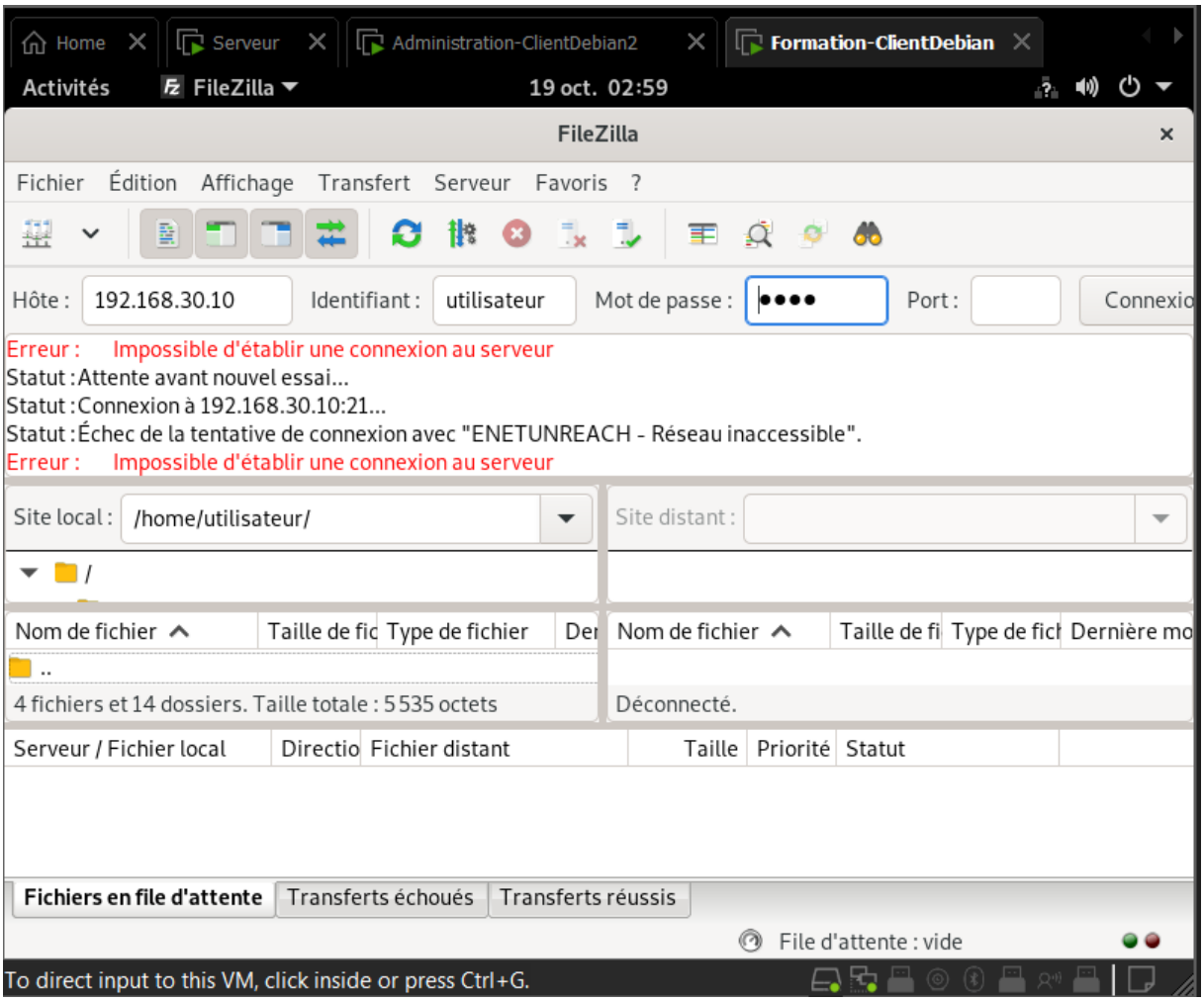
Accès FTP :

Depuis le poste administrateur :



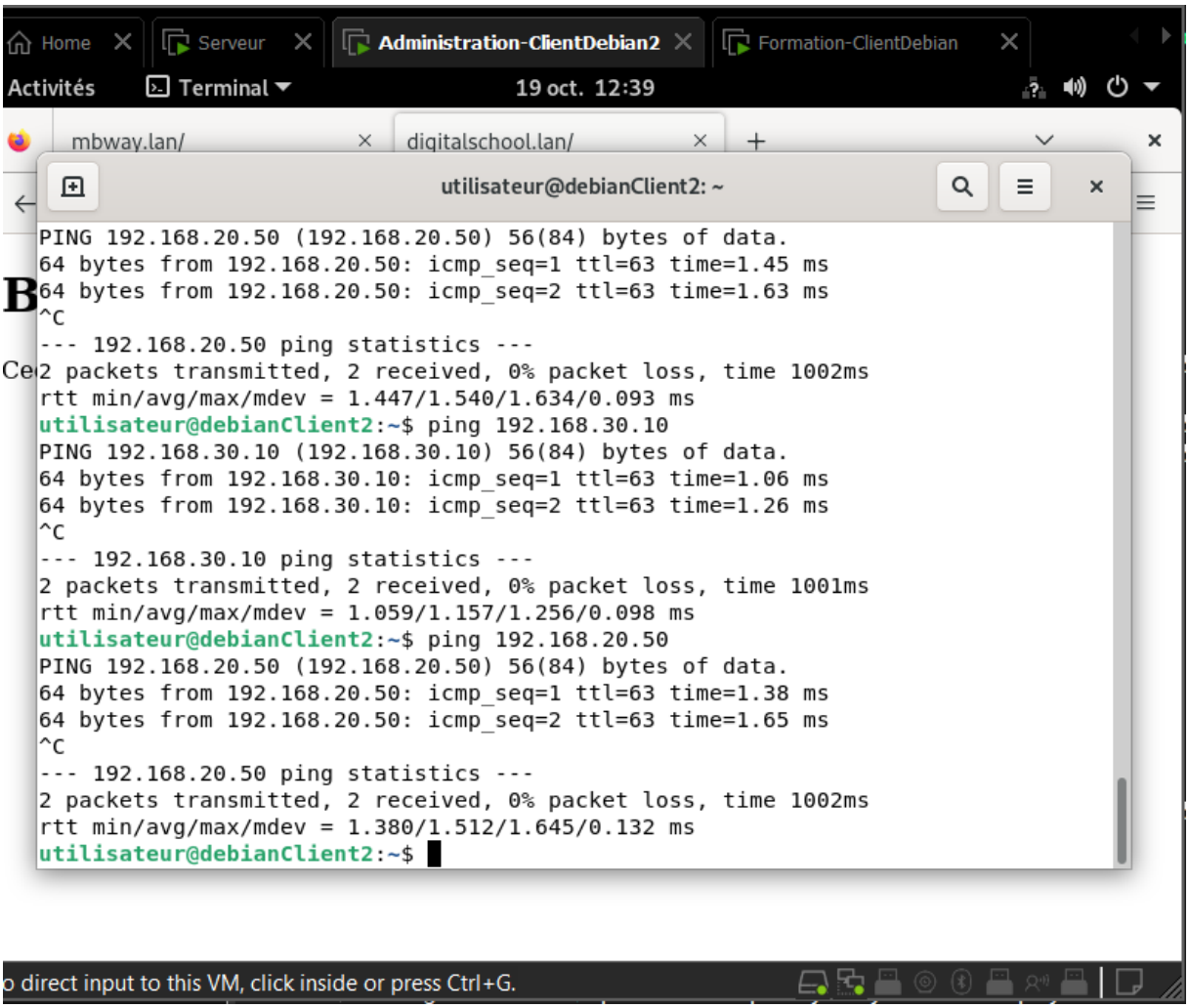


Depuis le poste formation :

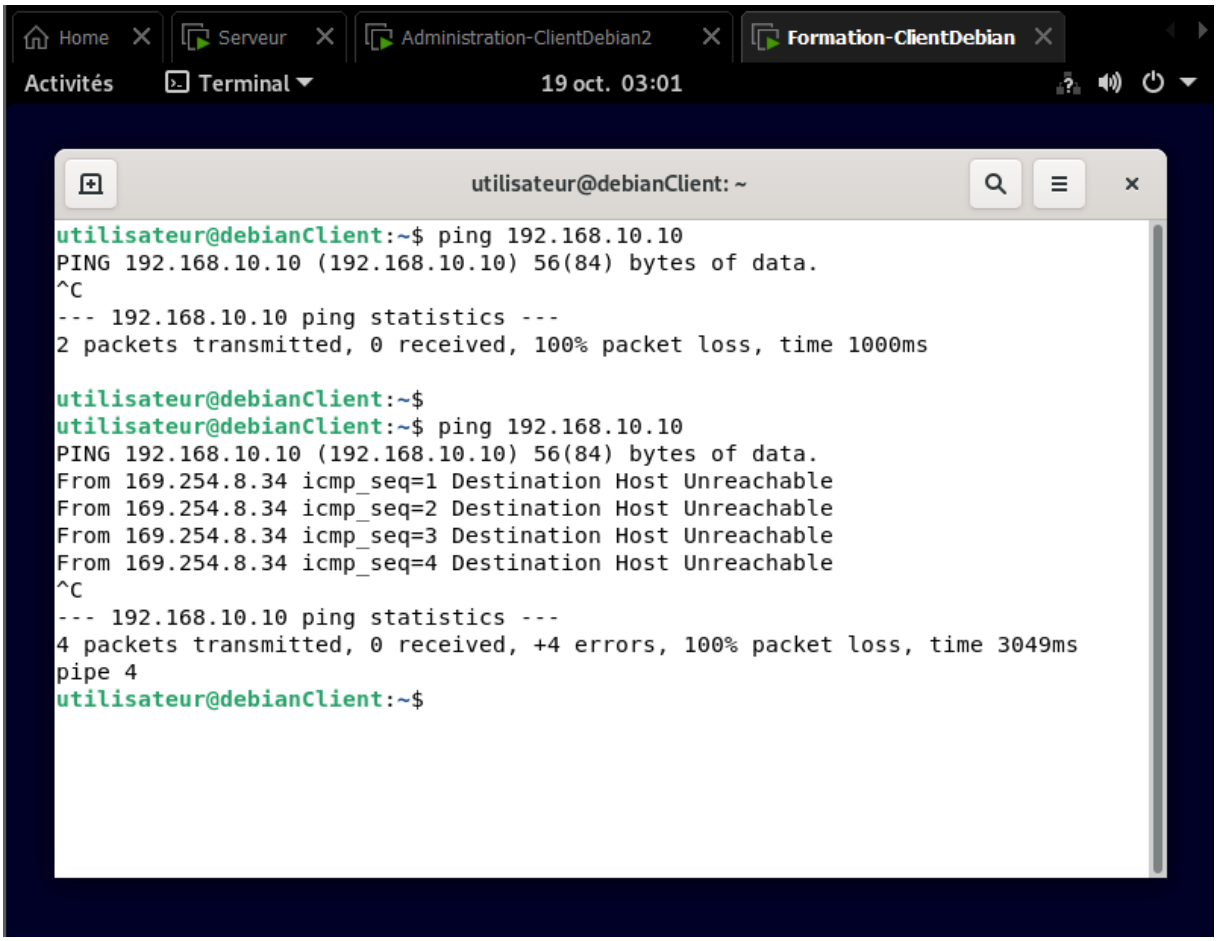


Accès VLAN :

Depuis le poste administrateur :



Depuis le poste formation :



Conclusion :

En conclusion, tous les points du cahier des charges ont été respectés. Les serveurs Debian sont regroupés dans un réseau unique, avec un serveur web, FTP, DNS et DHCP.

Tous les sous-réseaux obtiennent leurs paramètres TCP/IP du serveur DHCP commun.

Le serveur web héberge deux sites, le serveur FTP est réservé aux administrateurs, et le réseau administratif est isolé du réseau formation.

L'administrateur a accès aux équipements des salles de formation