Procédure de mise en place de la partie réseau dans le cadre de la SAE 24

PICHOT Owen,
NOACCO Lilian,
BOURGER Pierre

1. Adressages IP & organisation du réseau interne

Pour ce projet, quatres sous-réseaux seront requis, il nous faudra donc diviser l'adresse fournie par notre école [10.252.9.0]. Etant donné que nous aurons besoin de quatres sous-réseaux, nous allons prendre sur le dernier octet de notre adresses deux bits de sous-réseaux (car $2^2 = 4$) ce qui nous donne au final le tableau d'adressage suivant :

VLAN	Adresse sous-réseau	Plage d'adresses	Adresse de broadcast	Sous-interface associée
10	10.252.9.0	10.252.9.1-62	10.252.9.63	0.1 | 10.252.9.1
20	10.252.9.64	10.252.9.65-126	10.252.9.127	0.2 | 10.252.9.65
30	10.252.9.128	10.252.9.129-190	10.252.9.191	0.3 | 10.252.9.129
40	10.252.9.192	10.252.9.193-254	10.252.9.255	0.4 | 10.252.9.193

MASQUE : 255.255.255.192

Pour des raisons pratiques, nous avons également décidé de placer certains de nos périphériques sur des adresse ip fixes pour toujours pouvoir les joindre sur la même addresse.

ADRESSAGE STATIQUE	
Serveur WEB	10.252.9.131
Serveur DNS	10.252.9.132
Serveur FTP	10.252.9.133

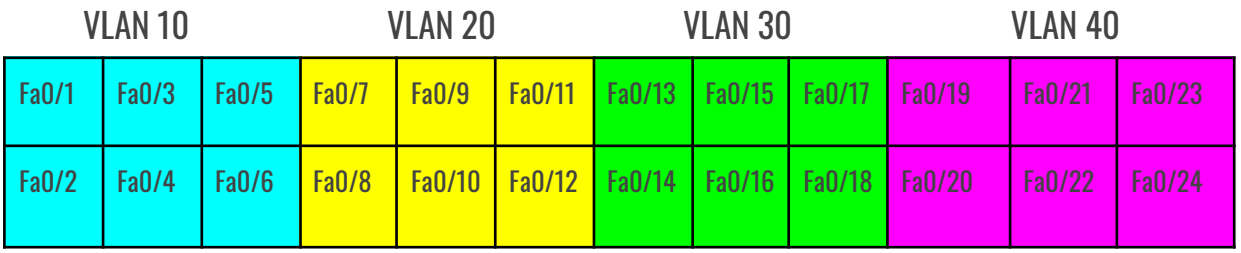
2. Configuration des équipements actifs (Routeur / switch)

a. Configuration du switch :

Il faut tout d'abord commencer par créer les vlans qui seront nécessaires à notre installation.

Action réalisée	Commandes associées
Création des Vlans	*Switch# conf t* *Switch(config)# vlan 10* *Switch(config-vlan)# name Voix* *Switch(config-vlan)# exit* *Switch(config)# vlan 20* *Switch(config-vlan)# name Users* *Switch(config-vlan)# exit* *Switch(config)# vlan 30* *Switch(config-vlan)# name Server* *Switch(config-vlan)# exit* *Switch(config)# vlan 40* *Switch(config-vlan)# name Administrateur*
Assignation des ports du switch à leurs vlans respectifs *Ajout du trunk pour lier les vlans*	*Switch(config)# interface range fa0/1-6* *Switch(config-if-range)# switchport mode access* *Switch(config-if-range)# switchport access vlan 10* *Switch(config-if-range)# no shutdown* *Switch(config)# interface range fa0/7-12* *Switch(config-if-range)# switchport mode access* *Switch(config-if-range)# switchport access vlan 20* *Switch(config-if-range)# no shutdown* *Switch(config)# interface range fa0/13-18* *Switch(config-if-range)# switchport mode access* *Switch(config-if-range)# switchport access vlan 30* *Switch(config-if-range)# no shutdown* *Switch(config)# interface range fa0/19-24* *Switch(config-if-range)# switchport mode access* *Switch(config-if-range)# switchport access vlan 40* *Switch(config-if-range)# no shutdown* *Switch(config)# interface gigabitEthernet0/1* *Switch(config-if)# switchport mode trunk*

Ci dessous une illustration de la disposition des vlans sur chaque port du switch à notre disposition (celui-ci possède 24 ports).



TRUNK

A ce stade de l'installation, il est possible de tester l'installation en plaçant deux périphériques au sein du même vlan et d'essayer de ping les deux (sous VM debian).
Il ne sera pas possible cependant de communiquer avec un périphérique présent dans un autre vlan car il nous manque la prochaine section de la procédure, à savoir :

b. La configuration du routeur :

Action réalisée	Commandes associées
Création de chaque interface & assignation de son IP + masque. SPécification de l'encapsulation pour l'associatio chaque VLAN /!\ Ne pas oublier d'effectuer un 'no sh' sur chaqu sous-interface /!\	*Router# conf t* *Router(config)# int gi0/0.1* *Router(config-subif)# encapsulation dot1q 10* *Router(config-subif)#ip address 10.252.9.1 255.255.255.192* *Router(config-subif)#exit* *Router(config)# int gi0/0.2* *Router(config-subif)# encapsulation dot1q 20* *Router(config-subif)#ip address 10.252.9.65 255.255.255.192* *Router(config-subif)#exit* *Router(config)# int gi0/0.3* *Router(config-subif)# encapsulation dot1q 30* *Router(config-subif)#ip address 10.252.9.129 255.255.255.192* *Router(config-subif)#exit* *Router(config)# int gi0/0.4* *Router(config-subif)# encapsulation dot1q 40* *Router(config-subif)#ip address 10.252.9.193 255.255.255.192* *Router(config-subif)#exit*

A ce stade, il serait de bonne pratique d'essayer de ping un ordinateur d'un vlan depuis un autre vlan. Si le routeur est bien configuré, tous les vlans devraient être en mesure de communiquer.
N.B : Il est très préférable de n'utiliser que des machines basées LINUX pour les tests, Windows pouvant potentiellement bloquer les requêtes ICMP.

Il nous faut maintenant assigner à chacun de nos utilisateurs des adresses IP et pour éviter à nos utilisateurs de les entrer à la main, nous allons mettre en place le service DHCP sur le routeur.

Action réalisée	Commandes associées
Préciser les adresses qui ne pourront pas être données aux utilisateurs déjà utilisées de base notamment mais également les 10 premiers hôtes de chaque sous-réseau)	*Router# conf t* *Routeur(config)# ip dhcp excluded-address 10.252.9.1 10.252.9.10* *Routeur(config)# ip dhcp excluded-address 10.252.9.65 10.252.9.75* *Routeur(config)# ip dhcp excluded-address 10.252.9.129 10.252.9.139* *Routeur(config)# ip dhcp excluded-address 10.252.9.193 10.252.9.203*
Créer des 'pools' DHCP précisant la range des adresses à attribuer, la passerelle par défaut pour chaque sous-réseau et (optionnellement) le serveur DNS qui devra être utilisé	*Routeur(config)# ip dhcp pool V10* *Routeur(dhcp-config)# network 10.252.9.0 255.255.255.192* *Routeur(dhcp-config)# default-router 10.252.9.1* *Routeur(dhcp-config)#exit* *Routeur(config)# ip dhcp pool V20* *Routeur(dhcp-config)# network 10.252.9.64 255.255.255.192* *Routeur(dhcp-config)# default-router 10.252.9.65* *Routeur(dhcp-config)# dns-server 10.252.9.132* *Routeur(dhcp-config)#exit* *Routeur(config)# ip dhcp pool V30* *Routeur(dhcp-config)# network 10.252.9.128 255.255.255.192* *Routeur(dhcp-config)# default-router 10.252.9.129* *Routeur(dhcp-config)#exit* *Routeur(config)# ip dhcp pool V40* *Routeur(dhcp-config)# network 10.252.9.192 255.255.255.192* *Routeur(dhcp-config)# default-router 10.252.9.193* *Routeur(dhcp-config)#exit*

Cette étape n'est pas obligatoire pour le bon fonctionnement de votre intranet mais est tout de même indispensable pour ces utilisateurs : Permettre un accès à internet via votre routeur

Pour ce faire, nous allons déjà commencer par brancher un câble ethernet de port Gi0/1 du routeur vers votre accès internet (UHA dans notre cas).

Ensuite, il ne nous reste plus qu'à configurer notre routeur et ses sous-interfaces pour permettre aux utilisateurs d'utiliser internet comme accès externe.

Action réalisée	Commandes associées
Sur l'interface du routeur que l'on a connecté au reste de l'uha, on la configure en dynamique afin que celle ci récupère une adresse de l'UHA	*Routeur(config)# int gi0/1* *Routeur(config-if)# ip address dhcp* *Routeur(config-if)# ip nat outside* *Routeur(config-if)# no shutdown*
On ajoute ensuite à chaque SI du routeur la commande suivante	*Routeur(config)# int gi0/0.X* *Routeur(config-if)# ip nat inside*
Il ne nous reste plus qu'à créer une ACL permettant l'accès vers l'extérieur.	*Routeur(config)# ip access-list standard local* *Routeur(config-std-nacl)# permit 10.252.9.0 0.0.0.255*
On assigne ensuite cet ACL à l'interface connectée à l'uha	*Routeur(config)# ip nat inside source list local int gi0/1 overload*
On prend ensuite la passerelle du réseau que l'uha nous a fourni et on termine la procédure avec la commande suivante	*Routeur(config)# ip route 0.0.0.0 0.0.0.0 10.129.6.1*

Désormais, il ne reste plus qu'à ajouter à la configuration du routeur les différentes limitations inter-vlan afin de limiter les déplacements au sein du réseau de chaque utilisateur.

Pour ce faire, nous allons passer par des ACL (Access Control List), celles-ci permettent de permettre ou refuser les accès d'un endroit X à un endroit Y.
Il est important de noter que les ACL sont assignés par ordre de priorité (de la première à la dernière) et que tout ce qui n'est pas spécifié sera refusé d'avance (d'où les 'ip any any').

Commande à entrer	Définition de la règle
access-list 100 permit ip any any	N'importe quelle adresse peut joindre n'importe quelle autre adresse
access-list 120 permit ip 10.252.9.64 0.0.0.63 10.252.9.128 0.0.0.6	VLAN 20 (utils) peut accéder au VLAN 30 (serveur)
access-list 120 deny ip 10.252.9.64 0.0.0.63 10.252.9.192 0.0.0.63	VLAN 20 ne peut pas accéder au VLAN 40 (Admin)

access-list 120 deny ip 10.252.1.64 0.0.0.63 host 91.211.165.100	VLAN 20 ne peut pas accéder au site web Materiel.net
access-list 120 deny icmp 10.252.9.64 0.0.0.63 host 8.8.8.8	VLAN 20 ne peut pas ping le DNS de Google
access-list 120 permit ip 10.252.1.64 0.0.0.63 any	VLAN 20 peut accéder à n'importe quelle autre adresse
access-list 120 permit ip any any	N'importe quelle adresse peut joindre n'importe quelle autre adresse
access-list 130 permit ip 10.252.9.0 0.0.0.63 10.252.9.128 0.0.0.63	Tout notre réseau peut accéder au VLAN 30 (serveur)
access-list 130 permit ip any any	N'importe quelle adresse peut joindre n'importe quelle autre adresse
access-list 150 permit tcp 10.252.9.192 0.0.0.63 host 10.252.9.133	VLAN 40 peut accéder en ftp au serveur ftp
access-list 150 deny tcp any host 10.252.9.133 eq ftp	Personne ne peut accéder au serveur ftp
access-list 150 permit ip any any	N'importe quelle adresse peut joindre n'importe quelle autre adresse

Il ne manque maintenant plus qu'à assigner les ACL créés à nos sous-interfaces respectives

Action réalisée	Commandes associées
Associer les règles 130 entrante au VLAN 10	*Routeur(config)# int G0/0.1* *Routeur(config-if)# ip access group 130 in*
Associer les règles 120 entrante au VLAN 20	*Routeur(config)# int G0/0.2* *Routeur(config-if)# ip access group 120 in*
Associer les règles 150 entrante au VLAN 30	*Routeur(config)# int G0/0.3* *Routeur(config-if)# ip access group 150 in*
Associer les règles 100 entrante au VLAN 40	*Routeur(config)# int G0/0.4* *Routeur(config-if)# ip access group 100 in*

3. Configuration & implémentation du service FTP

Création d'un serveur FTP avec vsftpd, recommandé par la documentation Debian pour la mettre en œuvre ce genre d'installation. Pour ceci nous aurons besoin d'une machine debian ou vous taperez les commande suivantes. Le résultat sera la réalisation d'un serveur FTP local, fonctionnel et sécurisé.

Action réalisée	Commandes associées
Installation de VSFTPD	*apt install vsftpd* *vsftpd -versions* *nano /etc/vsftpd.conf*
Modifications du fichier configuration	*nano /etc/vsftpd.* *listen=NO* *listen_ipv6=NO* *anonymous_enable=NO* *local_enable=YES* *write_enable=YES* *userlist_enable=YES* *userlist_file=/etc/vsftpd.userlist* *userlist_deny=NO* *dirmessage_enable=YES* *use_localtime=YES* *xferlog_enable=YES* *connect_from_port_20=YES* *secure_chroot_dir=/var/run/vsftpd/empty* *pam_service_name=vsftpd* *rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem* *rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key* *ssl_enable=NO*
Création de l'utilisateur « admin » et redémarrage du s	*adduser admin* *echo « admin » | tee -a /etc/vsftpd.userlist* *service vsftpd restart*

4. Configuration & implémentation du service WEB

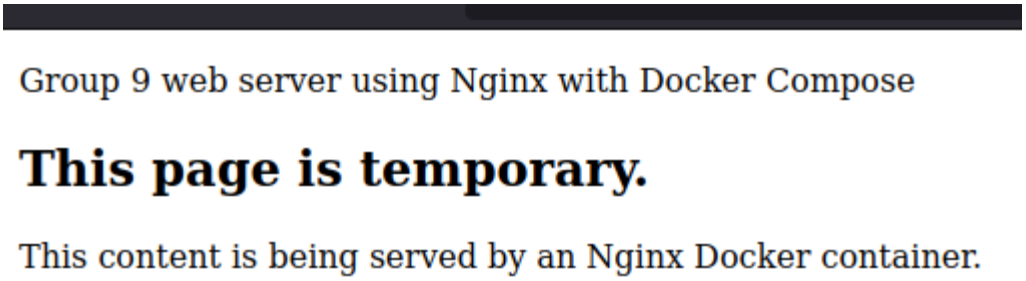
Nous avons choisi d'utiliser **nginx** avec **Docker** pour mettre en place notre serveur web. Pour ce faire, nous avons d'abord installé Docker et Docker **Compose** en exécutant la commande "***sudo apt install Docker -y ; sudo apt install docker-compose -y***". Une fois Docker et Docker Compose installés, nous avons créé le fichier **docker-compose.yml** pour construire **l'image** du conteneur. Voir la configuration que nous avons utilisée (Fig.1)

```
  GNU nano 6.2                          docker-compose.yaml
version: '3'
services:
  web:
    image: nginx:latest
    ports:
      - "80:80"
    volumes:
      - ./src:/usr/share/nginx/html
```

Nous avons choisi de mapper le port du docker sur le port externe 80 afin d'arriver directement sur la page web sans préciser de port spécifique.
Afin d'afficher une page web, il a fallu créer un dossier /src pour un fichier index.html pour créer la page web. Voir la configuration du fichier html (Fig.2)

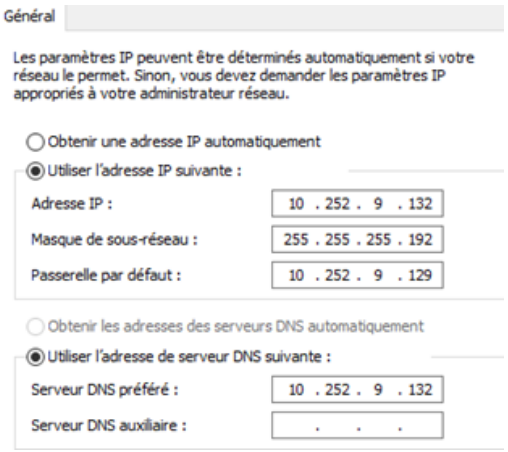
```
  GNU nano 6.2                                index.html
<!doctype html>
<html lang="en">
<head>

<p>Group 9 web server using Nginx with Docker Compose</p>


</head>
<body>

<h2>This page is temporary.</h2>
<p>This content is being served by an Nginx Docker container.</p>

</body>
</html>
```

Une fois que l’on accède à l’adresse du webserver, la page apparait comme il se doit:



Il ne reste plus qu'à mettre en place le DNS et le rediriger vers le serveur web.

5. Configuration & implémentation du service DNS



Pour déployer un serveur DNS sous Windows Server nous allons, tout d'abord récupérer la VM fournis dans le dossier Master. (Sinon fournissez-vous une version de windows server entre 2016 et 2019 puis mettez la dans une Machine Virtuelle.) Lancez la VM.

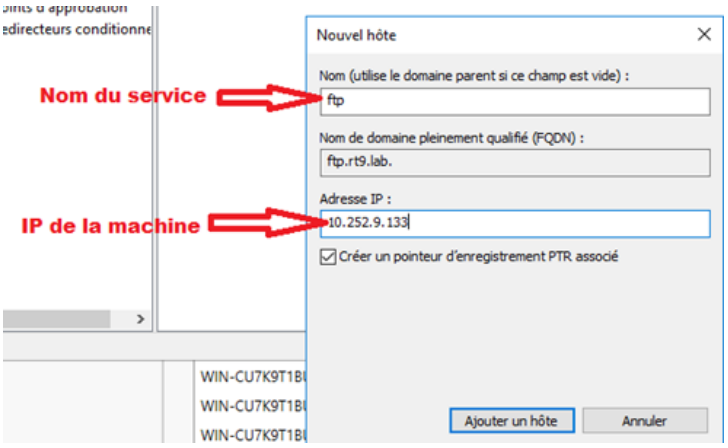
N'oubliez pas de paramétrer la carte réseau en bridge de la VM, et de configurer une IP statique correspondant à votre installation. (« ncpa.cpl » dans la VM).

Initialisation :

1	cliquez sur « Ajouter des rôles et des fonctionnalités »
2	Sélectionnez « Type d'installation » -> Installation basée sur un rôle ou une fonctionnalité
3	Cocher la case « Serveur DNS », décocher les autres cases. è Cliquez sur installer !
4	Après redémarrage cliquer sur l'onglet DNS puis clique droit sur votre nom de serveur -> gérez votre DNS
5	Clique droit sur « Nouvelle Zone » Suivant > Zone Principale > rt9.lab > Suivants > Ne pas autoriser les mise à jour.....> Terminer

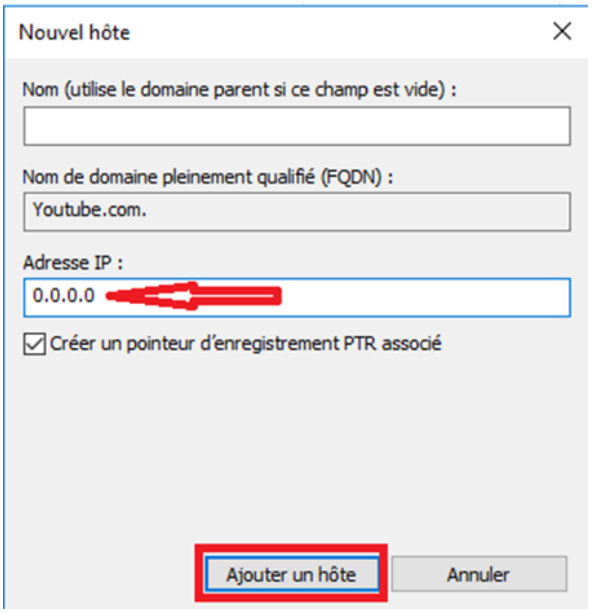
Redirection DNS vers le serveur web et ftp

Pour rediriger vers votre serveur web via un FQDN cliquer sur votre zone de recherche rt9.lab, puis clique droit -> Nouvelle Hôte -> Remplir de la sorte.



Bloquer un site sur le réseau local via le DNS.

Suivez la démarche d'initialisation, en remplaçant rt9.lab par le site que vous voulez bloquer (exemple « Youtube.com ») Puis comme précédemment, ajouté un nouvel hôte que vous remplirez de la sorte.



FIN

Merci de votre attention