

## SAE – Concevoir un réseau multisite

### RAPPORT – Partie FAI

Valentin Long - Julien Losser - Morgan Bois

#### Objectif

Dans cette partie de la SAE, nous devons reproduire la topologie d'un fournisseur d'accès à internet (FAI). Le but ici est de créer plusieurs systèmes autonomes (AS) qui devront être interconnectés entre eux, et pour cela, nous utiliserons le protocole BGP qui permet de faire du routage dynamique inter-AS. Nous cherchons également à faire communiquer nos sites entre eux à la manière d'un VPN, mais du côté FAI. Pour cela, MPLS-VPN sera mis en œuvre.

#### Topologie

La topologie du FAI étant imposée, il nous faut mettre en œuvre les différents protocoles imposés pour que nos sites puissent communiquer entre eux.

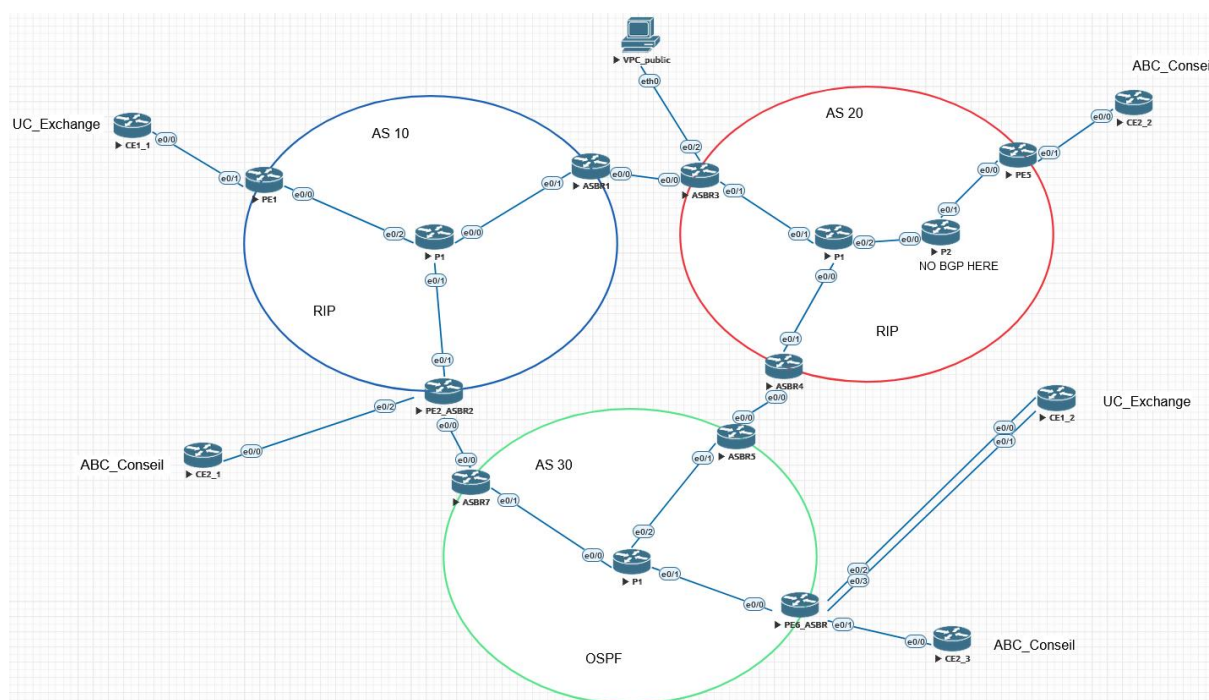


Figure 1 : Topologie du FAI

Nous disposons de 3 systèmes autonomes ainsi que 5 sites distants qui sont ABC\_Conseil et UC\_Exchange. Les sites d'un même nom doivent pouvoir échanger comme s'ils étaient dans le même réseau local.

Les routeurs de bordure inter AS, qui comportent ASBR dans leurs noms, excepté le PE6\_ASBR6, doivent être configurés de manière qu'ils puissent faire transiter le Trafic de nos réseaux locaux. Nous utiliserons pour cela le protocole eBGP. Aucune des adresses assignées à l'intérieur d'un AS, comme celle des loopback ou des ports n'est atteignable par un autre AS. Seuls les routeurs de bordures peuvent effectuer des ping inter AS à l'aide de leurs adresses en 172.16.X.X mais ils ne sont limités qu'à leurs voisins en eBGP.

A l'intérieur de l'AS, nous utiliserons des protocoles de routage dynamiques classiques comme OSPF ou RIP qui représenteront la partie IGP. Cela permet aux routeurs de faire une cartographie de leur AS. Il y aura également des sessions IBGP présentes à l'intérieur des systèmes. Ces sessions permettent de créer des liens de « peering » avec les autres routeurs de l'AS. On possède ainsi une connexion directe avec chaque nœud quel que soit la topologie. On pourra combiner cela avec un routeur qui jouera le rôle de route-reflector (réflecteur de route), ce qui facilitera encore plus la création des AS.

## Mise en place d'un AS

Pour créer un AS, nous avons besoin de routeurs. En théorie, un seul routeur suffirait pour créer un AS mais cela limitera le réseau concernant les pannes, la disponibilité, ... et surtout ce sera inutile. Dans notre cas, dans chaque AS, nous avons au minimum 4 routeurs. Dans ces derniers il y a les P, qui sont les cœurs de chaque AS, les PE, qui sont les routeurs de bordures avec les routeurs clients (ce sont les CE voir figure 1) et les ASBR qui sont les routeurs communiquant avec l'AS voisin. Chaque AS possède un numéro entre 0 et 65535 qui l'identifie. Nous créerons donc les 3 systèmes imposés (10, 20 et 30). La figure ci-contre représente la topologie de chaque AS à quelques exceptions près. Chaque routeur possède des adresses aux ports et une adresse loopback. Ces adresses seront redistribuées à l'ensemble de l'AS via RIP ou OSPF (sauf les adresses en sortie l'AS). Une fois cela fait, il est temps de monter les sessions IBGP. Pour ce faire, nous allons désigner le routeur P1 comme route-reflector, ce qui va permettre de minimiser grandement le nombre de sessions BGP présente. Avec ceci, chaque routeur monte une session BGP avec P1. Ce dernier s'occupera de « refléter » les routes aux autres routeurs. Cela évite de créer des sessions entre chaque routeur de l'AS. Ci-dessous un exemple :

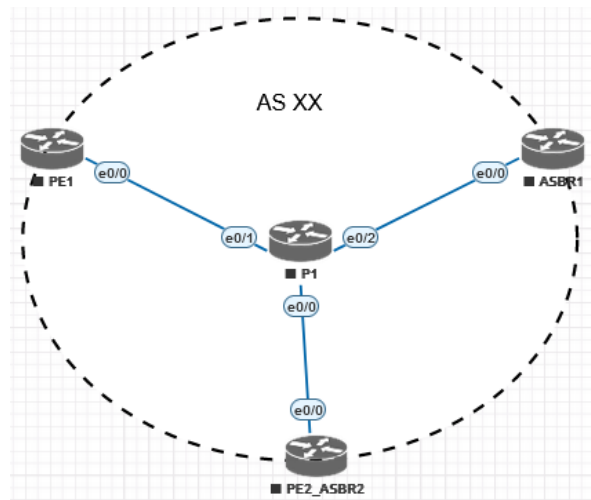


Figure 2 : Exemple d'AS

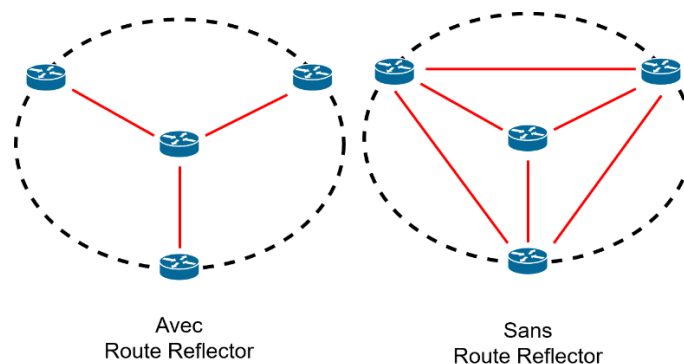


Figure 3 : Route-Reflector (en rouge les sessions IBGP)

## Routage IGP

---

Nous supposons ici que toutes les adresses IP ont été assignées et commencerons par la configuration des protocoles de routage IGP.

Pour OSPF, chaque réseau devra déclarer dans la configuration ses adresses et réseau auquel il est connecté y compris le réseau de la loopback. Attention, les réseaux en 172.16.X.X ne sont pas concernés dans l'IGP comme cité plus haut.

Exemple de configuration OSPF :

```
router ospf 1
 network 10.10.8.8 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.3 area 0
 network 10.10.10.4 0.0.0.3 area 0
 network 10.10.10.8 0.0.0.3 area 0
```

Dans cet exemple, c'est le routeur P de l'AS 30 qui déclare ses réseaux. Le premier réseau entré est sa loopback suivi du masque inversé. Le reste des réseaux sont ceux qui lui sont directement connectés aux interfaces physiques.

Pour RIP, il suffit juste de déclarer ceci :

```
router rip
 version 2
 network 10.0.0.0
 no auto-summary
```

La version 2 de rip doit être activée car elle prend en compte le VLSM. C'est utile car on n'est pas limité par des masques de sous-réseau en classfull. La commande « no auto-summary » permet justement de diffuser des masques classless et d'éviter de les résumer. Avec ces commandes, il nous faut juste déclarer la partie commune à tous nos réseaux connectés au routeur qui est ici 10.0.0.0.

## Sessions IBGP

---

Le routage effectué, il nous faut maintenant monter nos liens BGP entre les routeurs. Nous allons donc désigner notre routeur P1 comme route-reflector.

Dans la configuration des cœurs, il nous faut insérer ceci :

```
router bgp 20
 neighbor ALL_20 peer-group
 neighbor ALL_20 update-source Loopback0
 neighbor ALL_20 route-reflector-client
 neighbor 10.10.9.9 remote-as 20
 neighbor 10.10.9.9 peer-group ALL_20
 neighbor 10.10.11.11 remote-as 20
 neighbor 10.10.11.11 peer-group ALL_20
 neighbor 10.10.12.12 remote-as 20
 neighbor 10.10.12.12 peer-group ALL_20
```

Dans cet exemple c'est le router P1 de l'AS 20 qui est utilisé. Les voisins ont été mis dans des groupes pour pouvoir leurs assigner des commandes groupées.

On déclare nos voisins :

```
neighbor [loopback du voisin] remote-as [Numéro de l'AS du routeur]
```

On insère les voisins dans un groupe pour pouvoir effectuer des commandes groupées :

```
neighbor [loopback du voisin] peer-group [nom du groupe]
```

On indique que ce sont des loopback :

```
neighbor [groupe] update-source Loopback0
```

On nomme le routeur comme route-reflector pour tout le monde :

```
neighbor [groupe] route-reflector-client
```

Une fois les cœurs configurés, il ne nous reste plus qu'à configurer tous les routeurs de bordures, ASBR comme PE, pour cibler le routeur P1 dans chaque AS. Exemple :

```
router bgp 30
 neighbor 10.10.8.8 remote-as 30
 neighbor 10.10.8.8 update-source Loopback0
```

Nous pouvons voir nos sessions BGP avec la commande « show ip bgp summary ». Exemple avec les sessions du routeur P1 de l'AS :

```
Switch#sh ip bgp summary
BGP router identifier 10.10.8.8, local AS number 30
BGP table version is 1, main routing table version 1

Neighbor      V      AS MsgRcvd MsgSent   TblVer  InQ  OutQ Up/Down  State/PfxRcd
10.10.5.5      4      30     56     62        1    0    0 00:47:46      0
10.10.6.6      4      30     59     62        1    0    0 00:47:57      0
10.10.7.7      4      30     56     62        1    0    0 00:47:37      0
```

Nos sessions sont bien établies. (Champ UP/DOWN)

Etant donné que nous possédons 3 AS, nous devons mettre en œuvre une communication entre eux. Pour cela nous devons positionner des sessions eBGP. C'est à peu près le même principe que l'IBGP, à l'exception que nous n'utilisons plus d'interface loopback, mais des interfaces physiques.

Dans tous les routeurs inter AS, il nous faudra insérer ceci :

```
router bgp [numéro de l'AS]
 neighbor [@IP de l'interface du voisin] remote-as [numéro de l'AS voisin]
```

## MPLS-VPN

Nos sessions BGP étant montées, nous avons un FAI basique à peu près fonctionnel. Nous devons maintenant nous occuper de la partie communication entre plusieurs sites. Comme indiqué dans l'objectif, nous souhaitons que les sites d'un même nom se relient entre eux et forme un même réseau.

Pour ce faire, MPLS-VPN sera déployé. Ce protocole offre divers avantages pour les entreprises ayant divers sites qui souhaitent être interconnectés. Il est « scalable », c'est-à-dire que l'on peut facilement ajouter des sites aux réseaux déjà interconnectés. Il offre une certaine qualité de service grâce aux étiquettes MPLS qui permettent de prioriser certains paquets. Il est économique en bande passante car il permet d'éviter de désencapsuler l'entête IP grâce aux labels. MPLS-VPN permet également de partager les coûts liés à la construction et à la maintenance d'un réseau privé, ce qui réduit les dépenses pour les entreprises.

La figure 4 représente les liens virtuels que nous souhaitons réaliser en MPLS-VPN.

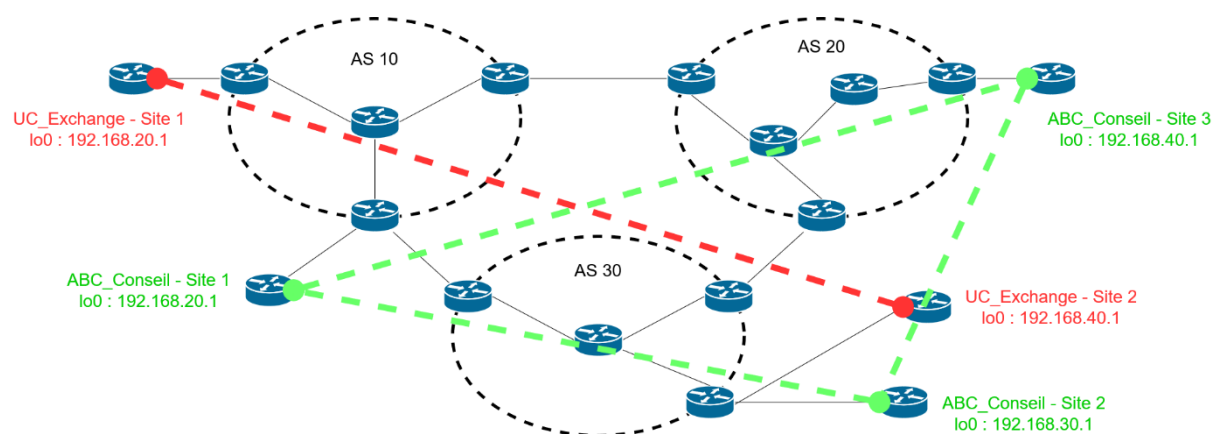


Figure 4 : Tunnels virtuels MPLS-VPN

Nous placerons sur les routeurs de bordure (tous les routeurs avec PE dans leur nom voir figure 1) ce qu'on appelle des VRF, pour Virtual Routing and Forwarding. Elles serviront à déclarer quel réseau doit faire partie du protocole et permettront de créer une instance de table de routage qui est associée à un client particulier.

## Mise en place d'MPLS-VPN

---

Pour mettre en place MPLS, nous devons tout d'abord l'activer sur les interfaces physique des routeurs. Dans les ASBR, les P et les PE mais pas dans ou vers les CE. La commande suivante permet cela :

```
mpls ip
```

MPLS activé nous pouvons positionner nos VRF selon les sites. Elles ne sont placées que lors d'une jonction entre un PE et un CE, et nulle part ailleurs.

Comme exemple le routeur PE6\_ASBR :

```
ip vrf ABC_Conseil
  rd 10:1
  route-target export 10:1
  route-target import 10:1

ip vrf UC_Exchange
  rd 10:2
  route-target export 10:2
  route-target import 10:2
```

Deux VRF sont ici présentes, car le routeur est connecté à deux sites. Le RD, pour route distinguisher, permet d'identifier à quel VPN appartient les paquets transmis. Le route-target est ensuite utilisé pour cibler la destination finale. Ces éléments sont contenus dans l'entête MPLS. Pour qu'ils puissent transiter entre les routeurs de chaque AS, nous devons déclarer des routes vpnv4. Ces routes seront échangées à l'aide du protocole MP-BGP, qui est une amélioration de BGP qui intègre le partage de ce type de routes.

Les route vpnv4 doivent être déclarées dans chaque routeur de chaque AS, excepté le P2 de l'AS 20. (Expliqué plus bas)

Voici un exemple de configuration de l'ASBR7 de l'AS 30 :

```
router bgp 30
  no bgp default route-target filter
  neighbor 10.10.8.8 remote-as 30
  neighbor 10.10.8.8 update-source Loopback0
  neighbor 172.16.10.1 remote-as 10

  address-family ipv4
    neighbor 10.10.8.8 activate
    neighbor 10.10.8.8 next-hop-self
    neighbor 172.16.10.1 activate

  address-family vpnv4
    neighbor 10.10.8.8 activate
    neighbor 10.10.8.8 next-hop-self
    neighbor 172.16.10.1 activate
```

Pour faire transiter nos routes vpnv4, il nous suffit de déclarer nos voisins comme étant actifs avec la commande :

```
neighbor [IP loopback ou physique si IBGP ou eBGP] activate
```

Ne pas oublier également de préciser le routeur qui joue le rôle de route-reflector comme prochain saut :

```
neighbor [IP loopback] next-hop-self
```

Il est nécessaire de préciser cela, sinon les routes transmises seront faussées, puisqu'elles posséderont l'adresse IP du saut précédent dans leurs prochain saut.

Cette procédure doit être répétée dans la partie « address-family ipv4 » car elle permet de configurer et de gérer les paramètres de routage associés à la famille d'adresses IPv4 dans le réseau MPLS-VPN, ce qui est essentiel pour assurer une communication efficace entre les réseaux de chaque site.

Ci-dessous la configuration exemple du cœur de l'AS 10 :

```
router bgp 10
  neighbor ALL_10 peer-group
  neighbor ALL_10 update-source Loopback0
  neighbor ALL_10 route-reflector-client
  neighbor 10.10.1.1 remote-as 10
  neighbor 10.10.1.1 peer-group ALL_10
  neighbor 10.10.2.2 remote-as 10
  neighbor 10.10.2.2 peer-group ALL_10
  neighbor 10.10.3.3 remote-as 10
  neighbor 10.10.3.3 peer-group ALL_10

  address-family vpnv4
    neighbor ALL_10 route-reflector-client
    neighbor 10.10.1.1 activate
    neighbor 10.10.2.2 activate
    neighbor 10.10.3.3 activate
```

Ici, pas besoin de déclarer de prochain saut puisque le routeur est le route-reflector. Il utilisera son @IP. La configuration IPv4, n'est pas non plus nécessaire ici, car le routeur qui joue le rôle de réflecteur l'utilise implicitement.

Dans les routeurs PE en lien avec les CE, il faut préciser la redistribution du réseau local en fonction du type de routage utilisé et du nom de la VRF pour partager ce même réseau avec les autres sites, exemple :

```
address-family ipv4 vrf ABC_Conseil
  redistribute static

address-family ipv4 vrf UC_Exchange
  redistribute rip
```

Voici la configuration à mettre en œuvre dans les PE voulant utiliser du RIP avec les CE :

```
router rip
version 2
network 10.0.0.0
no auto-summary

address-family ipv4 vrf UC_Exchange
redistribute bgp 30 metric transparent
network 172.16.0.0
no auto-summary
version 2
```

Il faut déclarer dans le processus RIP la famille d'adresse qu'utilise la ou les VRF présentes sur le routeur. Ainsi, les tables publiques et privées ne sont pas mélangées entre elles. La commande « redistribute bgp » permet de transmettre les route des autres réseaux de la VRF au routeur CE directement connecté qui est dans le réseau 192.16.0.0.

Pour les chemins statiques, il suffit de créer une route avec la commande suivante :

```
ip route vrf ABC_Conseil 192.168.30.0 255.255.255.0 172.16.20.21
```

Il faut bien préciser la VRF de cette route pour ne pas mélanger les tables. Au niveau des routeurs clients, une simple route par défaut suffit :

```
ip route 0.0.0.0 0.0.0.0 172.16.20.22
```

La route pointe vers l'adresse de l'interface du routeur connectée au PE. RIP peut également être mis en œuvre comme ceci :

```
router rip
version 2
network 172.16.0.0
network 192.168.40.0
no auto-summary
```

On précise le réseau de la loopback, qui, pour rappel émule un réseau local ainsi que le réseau de PE et CE.

Pour tester notre configuration, nous devons effectuer un ping depuis une adresse du réseau local (ici la loopback) vers un des sites dans la même VRF, exemple :

```
ping 192.168.30.1 source lo0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.30.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.20.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/12/14 ms
```



## Cas particulier

La routeur P2 présent dans l'AS 20, à une configuration différente de celle des autres routeurs présents dans les autres AS, comme l'illustre la figure ci-dessous :

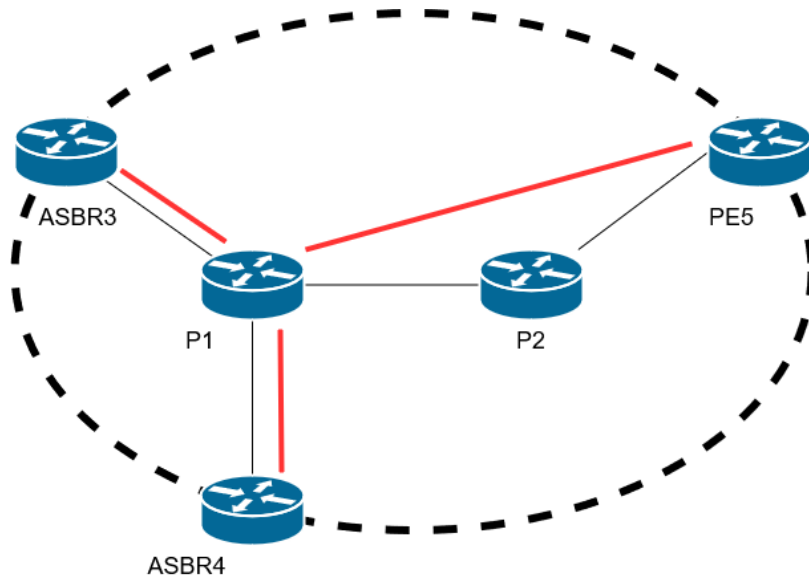


Figure 5 : Exception (En rouge les sessions BGP et MPLS)

Ce routeur n'étant ni un routeur de bordure, ni un cœur, il n'a pas besoin d'établir de sessions BGP et MPLS-VPN. Les routeurs de bordures « l'ignorent » et ciblent directement le routeur cœur P1. Il est quand même nécessaire de lui assigner des adresses IP (loopback et port) pour que le protocole RIP puisse fonctionner, car il se situe entre P1 et un routeur de bordure. L'activation de MPLS-IP sur les interfaces est néanmoins nécessaire pour autoriser le Traffic des paquets inter-sites.

## SAE – Concevoir un réseau multisite

### RAPPORT – Partie LAN

Valentin Long - Julien Losser - Morgan Bois

#### Objectif

L'objectif est de réaliser un réseau local comprenant plusieurs sous réseaux et devant être redondant pour limiter les pannes et augmenter la disponibilité. Pour mettre cela en œuvre plusieurs protocoles devront être utilisés. Nous déploierons le Multiple Spanning Tree (MST), qui est une amélioration du protocole STP et va nous permettre d'éviter les boucles de diffusion. VRRP sera également positionné, ceci dans le but d'avoir un accès au FAI redondant, ici une connexion Dual Homed. Nous devons également disposer d'un NAT, qui permettra une translation des adresses privées en une adresse publique.

#### Topologie

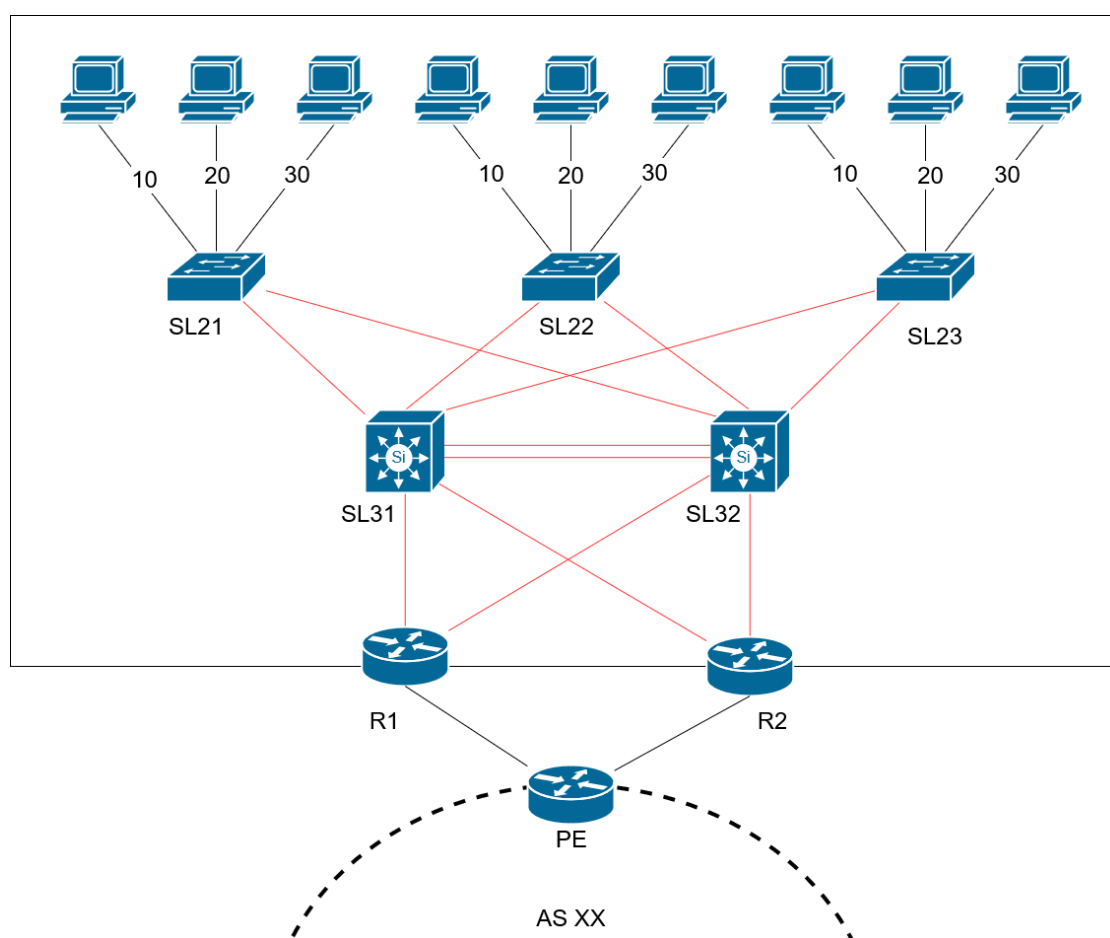


Figure 1 : Topologie du réseau

La figure ci-dessus représente la topologie que nous avons choisie d'utiliser pour nos réseaux locaux. En rouge les liens trunks, les nombres représentent les numéros des VLAN dans lequel se situent les liens. Nous avons opté pour une architecture 2 tiers, car cela nous semble plus adapté pour le nombre de VLAN que nous avons implémenté et c'est bien assez complexe à configurer. Comme la topologie le démontre, nous avons une redondance de tous les équipements réseaux présents dans le réseau local. Ainsi, si un routeur ou switch tombe en panne, tout le réseau ne sera pas inutilisable. Une agrégation entre 2 liens est également présente, ce qui permet d'obtenir de meilleures performances au niveau de la transmission des données car cela allège la charge des nœuds individuels.

Pour des raisons pratiques, nous avons choisi d'utiliser des masques en 255.255.255.0 ou /24. Cela nous permettra de découper plus facilement notre réseau. Ci-dessous l'adressage final :

		<b>UC_Exchange</b>	<b>ABC_Conseil</b>
<b>Site 1</b>	<b>VLAN 10</b>	10.242.11.0/24	10.252.11.0/24
	<b>VLAN 20</b>	10.242.12.0/24	10.252.12.0/24
	<b>VLAN 30</b>	10.242.13.0/24	10.252.13.0/24
<b>Site 2</b>	<b>VLAN 10</b>	10.242.21.0/24	10.252.21.0/24
	<b>VLAN 20</b>	10.242.22.0/24	10.252.22.0/24
	<b>VLAN 30</b>	10.242.23.0/24	10.252.23.0/24
<b>Site 3</b>	<b>VLAN 10</b>	-	10.252.31.0/24
	<b>VLAN 20</b>	-	10.252.32.0/24
	<b>VLAN 30</b>	-	10.252.33.0/24

Il est à prévoir qu'un serveur DHCP sera également présent, mais ce service n'est pas établi dans cette partie. Nous configurerons donc un serveur DHCP temporaire sur un des switch L3.

## Mise en place des VLAN

Étant donné que nous souhaitons avoir une certaine redondance dans notre réseau, il nous faut plusieurs passerelles pour chaque VLAN, car si l'une d'entre elles tombe en panne, la seconde pourra prendre le relais.

Pour permettre ceci, les deux routeurs qui servent de passerelle doivent posséder des adresses IP dans chaque VLAN. Attention, elle ne doit pas être la même pour les deux. Le tableau suivant prend comme exemple le site 1 d'UC\_Exchange :

	<b>R1</b>	<b>R2</b>
<b>VLAN 10</b>	10.242.11.253/24	10.242.11.252/24
<b>VLAN 20</b>	10.242.12.253/24	10.242.12.252/24
<b>VLAN 30</b>	10.242.13.253/24	10.242.13.252/24

La commande suivante permet d'assigner une adresse IP à une interface virtuelle (SVI) :

```
interface Vlan10
ip address 10.242.11.252 255.255.255.0
```

Attention, cette commande à bien été appliqué aux éléments R1 et R2, car ceux sont des images de switch de niveau 3 qui ont été utilisées. Il est possible d'obtenir le même résultat sur de vrais routeurs grâce aux sous interfaces, exemple :

```
interface e0/0.1
 encapsulation dot1Q 10
 ip address 10.242.11.252 255.255.255.0
```

En rouge le vlan encapsulé, en vert la sous interface de l'interface physique e0/0.

Nous continuerons de les appeler des routeurs pour éviter de les confondre avec les switches L3 de la couche de distribution.

Par la suite, dans la configuration des interface physique de chaque équipement L2 ou L3 du réseau (switch et routeurs), il faut déclarer les liens trunks, en rouge voir figure 1 :

```
interface e0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
```

Ce type de lien sert à faire transiter tout les vlan qui seront crée sur le réseau, à la manière d'une autoroute auquel se recordent les routes départementales.

Le type d'encapsulation « dot1Q » fait référence ici à la norme IEEE 802.1Q qui permet de classer les trames dans des VLAN. Cette norme rajoute un champ « TAG » codé sur 4 octets qui contient :

- Le champ « TPID » qui identifie les trames IEEE 802.1Q
- Le champ « priority » qui permet de définir la priorité d'un vlan par rapport à un autre
- Le champ « CFI » qui est toujours à zéro sauf si la trame est dans un réseau token ring
- Le champ « VID » qui correspond à l'ID du vlan

Cela permet une allocation plus efficace des ressources réseau, en regroupant les utilisateurs en fonction de leurs besoins en matière de bande passante et de sécurité.

Pour déclarer un vlan sur une interface, la commande ci-dessous est utilisé. Cela se configure sur les switches d'accès (dans notre topologie en tout-cas) :

```
switchport access vlan XX
```

XX étant le numéro du vlan

Pour distribuer nos vlans à tous les switchs sans devoir tout réécrire, nous pouvons utiliser le protocole VTP (Vlan Trunk Protocol) qui permet à un switch de partager ses vlan avec ceux qui lui sont abonnés.

Sur chaque switch :

```
vtp domain rt
vtp mode [client / serveur]
```

On lui précise un domaine, ainsi que le mode souhaité. Il ne peut y avoir qu'un serveur par domaine. Le serveur s'occupera de transmettre ses vlan aux clients.

Les switchs et routeurs étant configurés au niveau des vlans, nos équipements ont besoins d'une adresse IP. Comme nous possédons 2 passerelles, nous ne pouvons pas encore mettre en place un service DHCP (nous pourrions le faire sans passerelle mais il n'y a pas d'utilité ici). Il nous faudra d'abord activer VRRP, un protocole qui permet de gérer les liens redondants.

## Virtual Router Redundancy Protocol

Comme cité précédemment, posséder deux passerelles nécessite la création de liens redondants. VRRP offre cette possibilité en combinant plusieurs routeurs physiques en un seul et unique routeur dit « virtuel ». Ce routeur virtuel possèdera une adresse IP qui sera notre passerelle vers le monde extérieur. C'est ici qu'est sa force. Si un des routeurs tombe en panne ou qu'un des liens est rompu, l'adresse virtuelle de la passerelle sera toujours la même, ce qui rend les pannes complètement transparentes pour les utilisateurs du réseau. VRRP fonctionne en désignant un routeur « master » (maître) et un routeur sauvegarde (backup). Si le routeur maître est défaillant, le routeur secondaire prendra le relais. A droite un cas d'utilisation du protocole VRRP. Les adresses virtuelles de chaque vlan sont les mêmes pour chaque routeur du réseau local. Il ne faut pas oublier que les routeurs ont toujours une adresse dans chaque vlan qui leur correspond. (ce sont les adresses qui finissent par .252 ou .253). Nous pourrions à présent mettre en place un service DHCP qui ciblera l'adresse virtuelle comme passerelle.

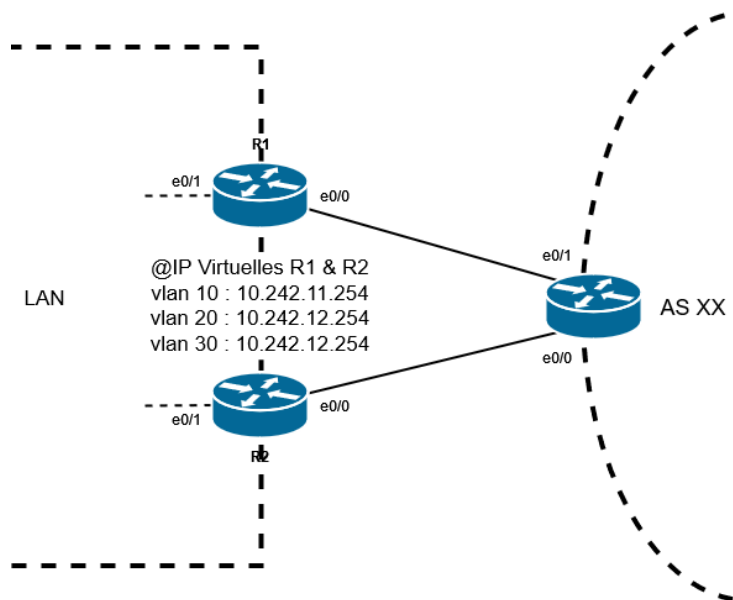


Figure 2 : VRRP

## VRRP - Mise en place

Dans chaque interface vlan des routeurs du réseau local, nous créerons une instance VRRP en précisant l'adresse virtuelle. Une priorité sera également définie, de sorte à pouvoir choisir quel routeur sera élevé au rang de « master ». Exemple de configuration pour R1 dans le vlan 10 :

```
interface Vlan10
ip address 10.242.11.253 255.255.255.0
vrrp 10 ip 10.242.11.254
vrrp 10 priority 150
```

En rouge l'instance VRRP, elle est différente pour chaque vlan. En vert l'adresse IP virtuelle et en bleu la priorité.

Si le voisin de R1, qui est R2 possède un nombre supérieur à 150, il deviendra « master ». Dans le cas contraire, il se contentera d'assurer la redondance de R1.

## DHCP

---

La mise en place du DHCP est ici anecdotique puisqu'une machine virtuelle dédiée s'en chargera. On l'utilisera uniquement pour des questions pratiques. Ce service sera placé temporairement sur un des switchs de la couche de distribution.

Pour chaque vlan, nous allons devoir créer un pool :

```
ip dhcp pool vlan10
 network 10.242.11.0 255.255.255.0
 default-router 10.242.11.254

ip dhcp pool vlan20
 network 10.242.12.0 255.255.255.0
 default-router 10.242.12.254

ip dhcp pool vlan30
 network 10.242.13.0 255.255.255.0
 default-router 10.242.13.254
```

La commande « network » précise le réseau dans lequel le DHCP doit être actif. Le « default-router » précise la passerelle. Nous précisons donc l'adresse de notre routeur virtuel créé précédemment.

Ne pas oublier d'exclure les adresses des équipements réseaux, exemple avec le vlan 10 :

```
ip dhcp excluded-address 10.242.11.250 10.242.11.254
```

On exclut ici les 5 dernières adresses pour laisser une marge au cas où nous en aurions besoin.

## Multiple Spanning Tree Protocol

---

Le Multiple Spanning Tree Protocol est une amélioration du protocole RSTP qui est lui-même une amélioration du protocole STP. Comparé à son parent RSTP, MST permet à plusieurs vlan d'utiliser une même instance de spanning-tree, ce qui réduit l'utilisation des ressources cpu et réseau. On peut ainsi regrouper plusieurs vlan dans la même instance. Le temps de convergence est également diminué. Il passe de 50 secondes pour le protocole STP classique à moins de 6 secondes pour le MST.

La configuration d'MST est la même pour tous les routeurs et switchs possédant des liens trunk dans notre topologie :

```
spanning-tree mst configuration
 name MST
 revision 1
 instance 1 vlan 10-20
 instance 2 vlan 30
```

Ici, la première instance supporte deux vlans, tandis que la seconde n'en possède qu'une. On pourrait s'imaginer que la première contient les vlans utilisateur et administration, tandis que la seconde instance contiendrait les services voulant disposer d'une haute qualité de service comme la VOIP, par exemple.

## Agrégation

---

Une agrégation est présente entre les switches de distribution SL31 et SL32 (voir figure 1). Ce lien permet de mieux répartir la charge entre les switches et offre une meilleure tolérance aux pannes. Une agrégation n'est autre que deux liens que l'on fusionne (au sens logiciel) pour n'obtenir qu'un seul et unique lien.

Sur les switches concernés par l'agrégation :

```
interface Ethernet1/0
  switchport trunk encapsulation dot1q
  switchport mode trunk
  channel-protocol pagp
  channel-group 8 mode desirable

interface Ethernet1/1
  switchport trunk encapsulation dot1q
  switchport mode trunk
  channel-protocol pagp
  channel-group 8 mode desirable

interface Port-channel 8
  switchport trunk encapsulation dot1q
  switchport mode trunk
```

Pour effectuer cette agrégation nous utilisons ici le protocole PAGP qui est simple d'utilisation. Premièrement, il faut déclarer les interfaces concernées par l'agrégation avec la commande « channel-protocol pagp ». On lui assigne ensuite un numéro de canal utilisé pour reconnaître les liens d'une même agrégation « channel-group 8 mode desirable ». Le mode « desirable » dans le protocole PAGP signifie que l'interface est configurée pour être active et à la recherche d'autres interfaces pour les agréger. Ceci effectué, il faut basculer toutes nos interfaces en mode trunk, ainsi que le nouveau lien d'agrégation créée qui se nomme « Port-channel 8 ».

## NAT

---

Une connexion nattée doit être mise en œuvre pour pouvoir communiquer avec internet. Dans notre cas, nous utiliserons un Nat dynamique avec surcharge (PAT). Cela permettra de traduire toutes les adresses privées sortantes en une seule adresse publique, d'où le mot « surcharge ».

Il nous faut auparavant créer une ACL qui autorisera ou non les réseaux à sortir via le NAT. Toute la configuration qui va suivre s'effectue sur les 2 routeurs.

```
access-list 1 permit 10.242.0.0 0.0.255.255
```

Cet ACL permet à tous les réseaux de chaque vlan de pouvoir sortir via le Nat. Aucune restriction n'a été imposée.

Ensuite, nous précisons quelle interface doit être surchargée :

```
ip nat inside source list 1 interface Ethernet0/0 overload
```

En rouge le numéro de l'ACL que nous venons de créer. Nous indiquons que l'interface de sortie vers internet est e0/0 ainsi que le type de nat qui est « overload » ou surcharge en français.

Il ne faut pas oublier également de préciser quelle sont les interfaces privées (inside) et les interfaces publiques (outside) du NAT :

```
ip nat [inside / outside]
```



## SAE – Concevoir un réseau multisite

### RAPPORT – Partie Services Réseaux

Valentin Long – Julien Losser – Morgan Bois

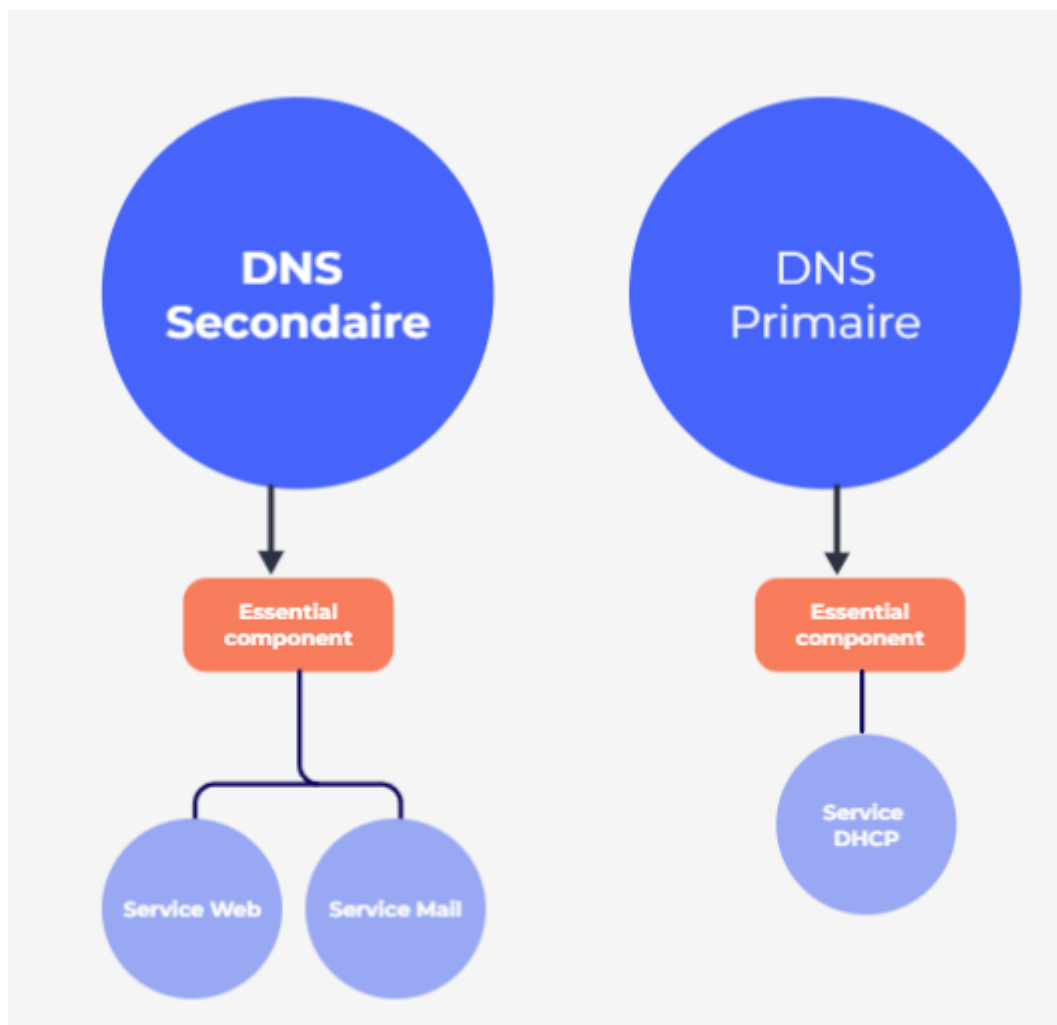
#### Objectif

---

Dans cette partie de la SAE, nous devons déployer un service DNS, serveur web Apache, serveur mail et DHCP pour le FAI. Les serveurs seront sous Linux, uniquement en ligne de commande. Un serveur DNS secondaire devra également être déployé, avec le transfert de zone chiffré avec DNSSEC. Un service web devra être déployé sur le même serveur que le DNS secondaire avec un site intranet et un site pour les machines extérieures. Un service mail avec un compte admin et un compte client sera déployé sur le serveur DNS secondaire/Web avec le trafic SMTP et IMAP autorisé.

#### Topologie

---



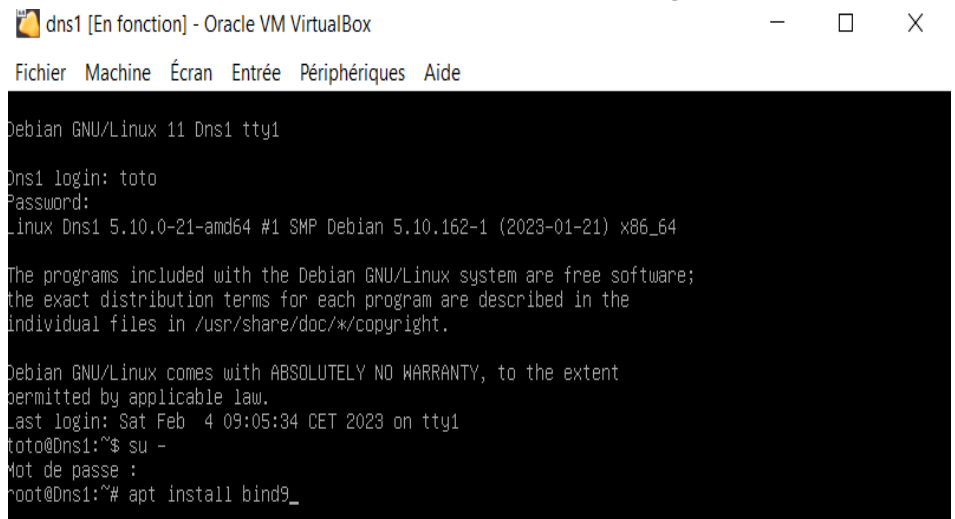
## Service DNS Primaire

Dans ce système, un serveur DNS primaire est un serveur qui héberge le fichier de zone primaire de notre site Web. Il s'agit d'un fichier de base de données texte qui contient toutes les informations faisant autorité pour un domaine, notamment son adresse IP, l'identité de l'administrateur du domaine et divers enregistrements de ressources.

Dans un premier temps il faut crée un Machine virtuelle linux mais sans interface graphique !

Il faudra commencer par installer différent paquet pour réaliser le DNS primaire comme bind9 et aussi les service apache2 et DHCP

Car il faut ensuite que cette VM soit en réseaux Nat donc en IP fixe afin d'avoir une IP qui ne change pas pour commencer.



```

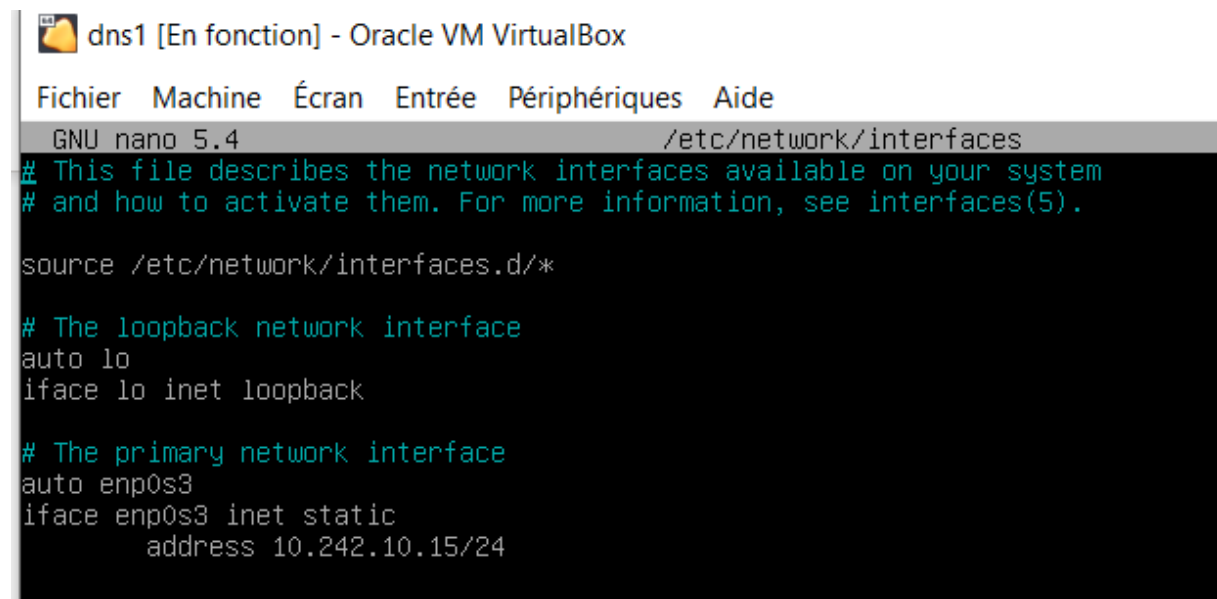
dns1 [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
Debian GNU/Linux 11 Dns1 tty1
Dns1 login: toto
Password:
Linux Dns1 5.10.0-21-amd64 #1 SMP Debian 5.10.162-1 (2023-01-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Feb  4 09:05:34 CET 2023 on tty1
toto@Dns1:~$ su -
Mot de passe :
root@Dns1:~# apt install bind9_

```

Il faudra taper la commande suivante « nano etc/network/interfaces » afin d'ouvrier le fichier de configuration puis on ajoute les paramétriser suivant « iface enp0s3 inet static », « address 10.242.10.15/24 ».



```

dns1 [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
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

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 10.242.10.15/24

```

Et pour finir redémarrer les services réseaux avec la commande « service networking restart ».

Dans Bind9 il y a plusieurs fichiers de configuration :


named.conf/named.conf.local / named.conf.options / named.conf.default-zones.

Le fichier named.conf permet de spécifier l'emplacement des fichiers de configuration. Le fichier named.conf.local nous permet de déclarer une zone ici par exemple ucexchange.com. Le fichier named.conf.options nous permet de spécifier des options sur la configuration du DNS.

Pour la configuration de named.conf dans le fichier named.conf nous allons lui spécifier l'emplacement des fichiers de configuration pour modifier le fichier named.conf.

Il faudra aller dans l'emplacement suivant « nano /etc/bind/named.conf »

Puis y ajouter les 3 includes suivants « include /etc/bind/named.conf.options » et « include /etc/bind/named.conf.local »

 dns1 [En fonction] - Oracle VM VirtualBox

Fichier Machine Écran Entrée Périphériques Aide

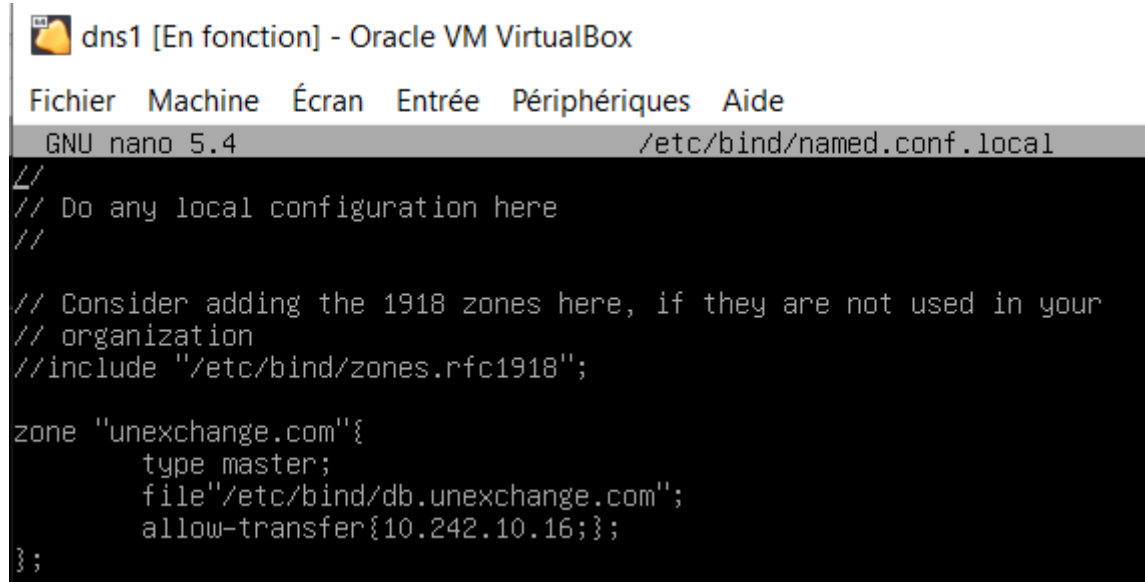
```
GNU nano 5.4 /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Pour la configuration de `named.conf.local`, nous allons lui identifier la zone.

Il faudra aller dans l'emplacement suivant « `nano /etc/bind/named.conf.local` ».

On définit notre zone ici `ucexchange.com`, le type en mode master, on indique l'emplacement du fichier `db.ucexchange.com` et on active `allow-transfer` sur l'ip de notre dns secondaire.



```

dns1 [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 5.4 /etc/bind/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 "unexchange.com"{
    type master;
    file "/etc/bind/db.unexchange.com";
    allow-transfer {10.242.10.16;};
};

```

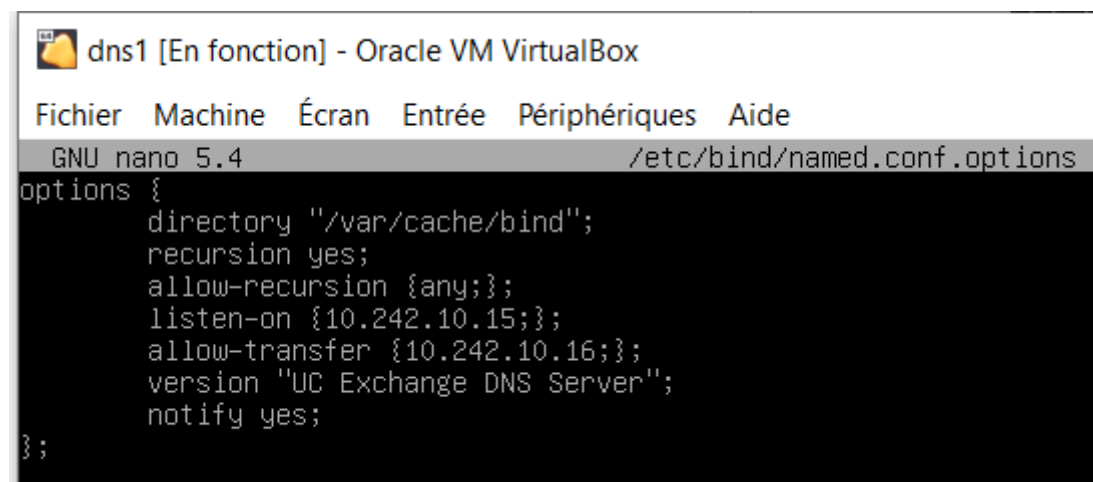
Pour la Configuration de `named.conf.options`, nous allons lui identifier les options de DNS.

Il faudra aller dans l'emplacement suivant « `nano /etc/bind/named.conf.options` ».

On ira définir l'emplacement du fichier de cache « `/var/cache/bind` ».

On activera les options suivantes : `recursion yes` / `allow-recursion {any;}` / `listen-on {sur l'ip du dns principal}` / `allow-transfer {sur l'ip du dns secondaire}`

Pour la prise en charge du DNS secondaire.



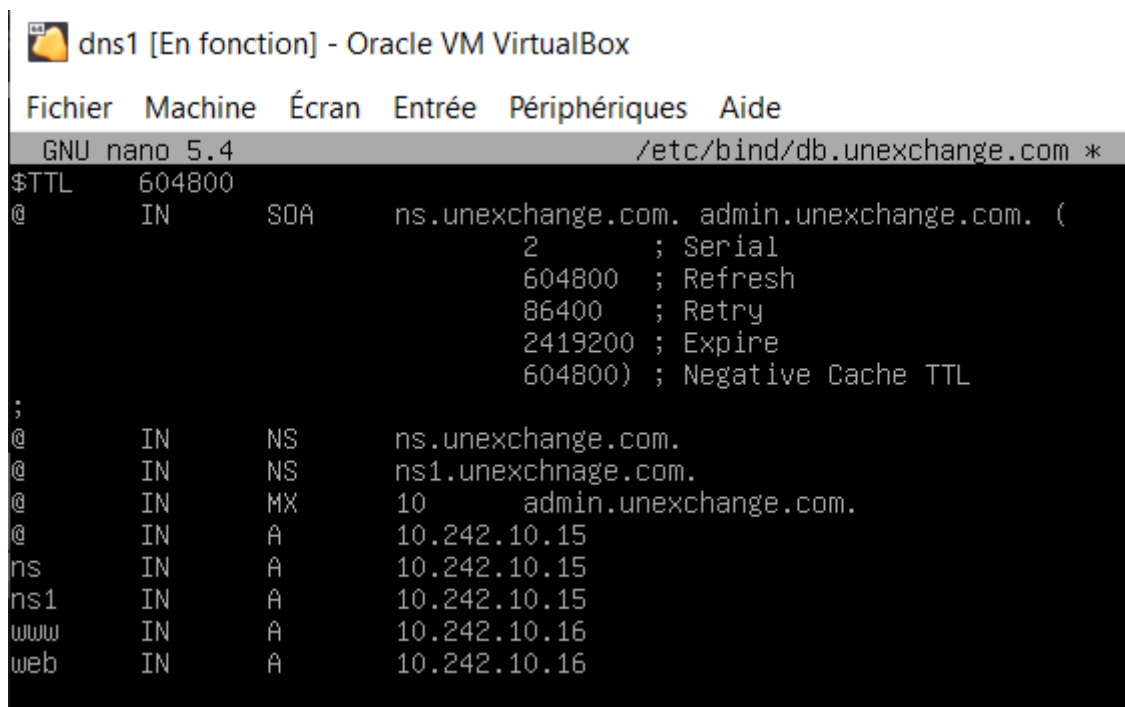
```

dns1 [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 5.4 /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";
    recursion yes;
    allow-recursion {any;};
    listen-on {10.242.10.15;};
    allow-transfer {10.242.10.16;};
    version "UC Exchange DNS Server";
    notify yes;
};

```

Pour la dernière configuration, c'est celle du fichier db.ucexchange.com dans le fichier named.conf.local nous allons ajouter un champ file avec un fichier nommer db.ucexchange.com ce fichier correspond à l'enregistrement du DNS ce fichier n'est pas créé, il va donc falloir penser à le créer et à le configurer.

Pour configurer le fichier db.ucexchange.com il faut taper la commande suivant afin de créer le fichier db.ucexchange.com «nano /etc/bind/db.ucexchange.com»



The screenshot shows a terminal window titled "dns1 [En fonction] - Oracle VM VirtualBox". The terminal is running the GNU nano 5.4 editor, editing the file /etc/bind/db.unexchange.com. The content of the file is as follows:

```
$TTL      604800
@         IN      SOA      ns.unexchange.com. admin.unexchange.com. (
                                2           ; Serial
                                604800      ; Refresh
                                86400       ; Retry
                                2419200    ; Expire
                                604800)    ; Negative Cache TTL
;
@         IN      NS       ns.unexchange.com.
@         IN      NS       ns1.unexchange.com.
@         IN      MX       10      admin.unexchange.com.
@         IN      A        10.242.10.15
ns        IN      A        10.242.10.15
ns1       IN      A        10.242.10.15
www       IN      A        10.242.10.16
web       IN      A        10.242.10.16
```

NS sont les enregistrements du nom du serveurs.

MX pour le mail


web pour le serveur web

www pour le serveur web

## Service DNS Secondaire

Pour configurer le DNS secondaire il faut avoir installer bind9 et appliqué une IP fixe

« 10.242.10.16/24 » le fichier named.conf est exactement le même que pour le DNS primaire.

 dns2 [En fonction] - Oracle VM VirtualBox


Fichier Machine Écran Entrée Périphériques Aide

```
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

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 10.242.10.16/24
```

 dns2 [En fonction] - Oracle VM VirtualBox

Fichier Machine Écran Entrée Périphériques Aide

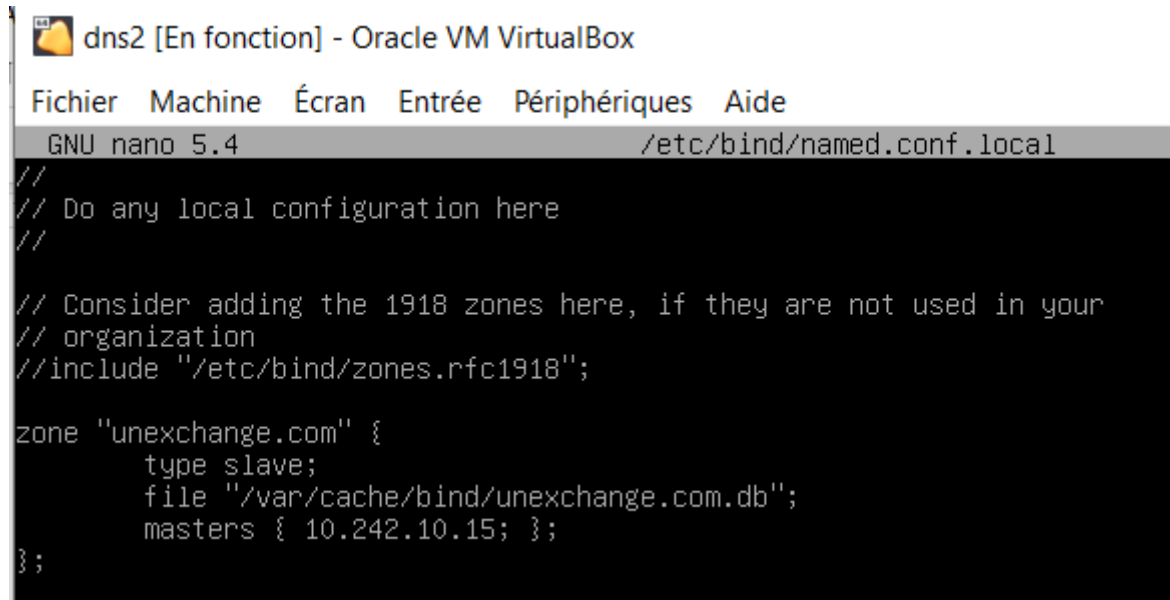
```
GNU nano 5.4 /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Pour la configuration de named.conf.local dans le fichier named.conf.local nous allons lui identifier la zone.

Il faudra aller dans l'emplacement suivant « nano /etc/bind/named.conf.local ».

On définit notre zone ici ucexchange.com, le type en mode slave, on indique l'emplacement du fichier ucexchange.com.db et on active master sur l'IP de notre DNS principal.



```

dns2 [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 5.4 /etc/bind/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";

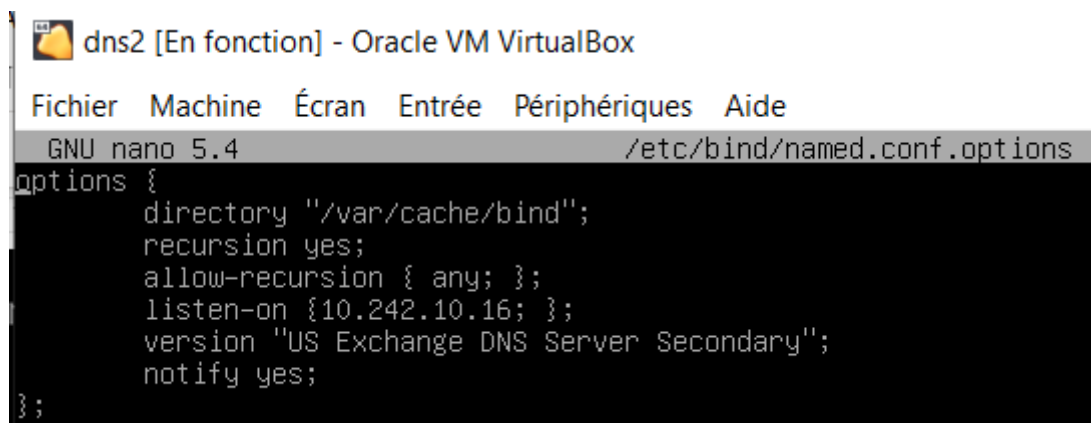
zone "unexchange.com" {
    type slave;
    file "/var/cache/bind/unexchange.com.db";
    masters { 10.242.10.15; };
};

```

Pour configuration de named.conf.options dans le fichier named.conf.options nous allons lui identifier les options de DNS.

Il faudra aller dans l'emplacement suivant « nano /etc/bind/named.conf.options ».

On activera les options suivantes : « /var/cache/bind » / recursion yes / allow-recursion {any} / listen-on {sur l'ip du dns secondaire}.



```

dns2 [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 5.4 /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";
    recursion yes;
    allow-recursion { any; };
    listen-on {10.242.10.16; };
    version "US Exchange DNS Server Secondary";
    notify yes;
};

```

## Service Apache2

---

L'objectif est de configurer un serveur web apache2 avec 2 sites internet. Pour configurer un serveur web, il faut installer apache2 avec la commande suivante « apt install apache2 ».

Pour la première configuration du virtual host 1:

Il faudra aller dans l'emplacement suivant « cd /etc/apache2/sites-enabled »

Une fois dans le répertoire on entre dans le fichier « nano 000-default.conf » ce fichier permet de spécifier la configuration du premier serveur web. Nous allons modifier les paramètres servers alias pour ajouter « web. » devant ucexchange.com et l'emplacement du fichier.



dns2 [En fonction] - Oracle VM VirtualBox

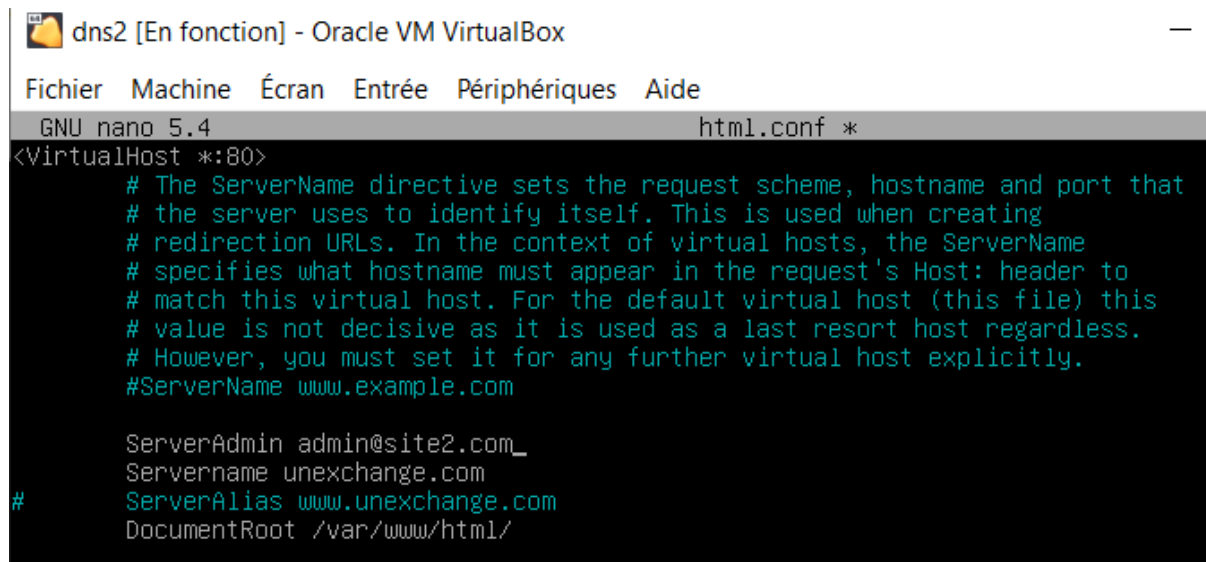
```
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 5.4                                000-default.conf *
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In 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.example.com

ServerAdmin webmaster@localhost
ServerName ucexchange.com
ServerAlias web.ucexchange.com_
DocumentRoot /var/www/html
```



Et pour la configuration du virtual host 2:

Une fois le premier virtual host configuré il faut en créer un deuxième pour le deuxième serveur web pour ça, nous allons créer un deuxième fichier pour plus de lisibilité « site2.com.conf » pensé à copier la configuration de l'autre fichier. Une fois dans le fichier, il faut modifier serveradmin, servername, serveralias et documentroot.



```

dns2 [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 5.4                                html.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.example.com

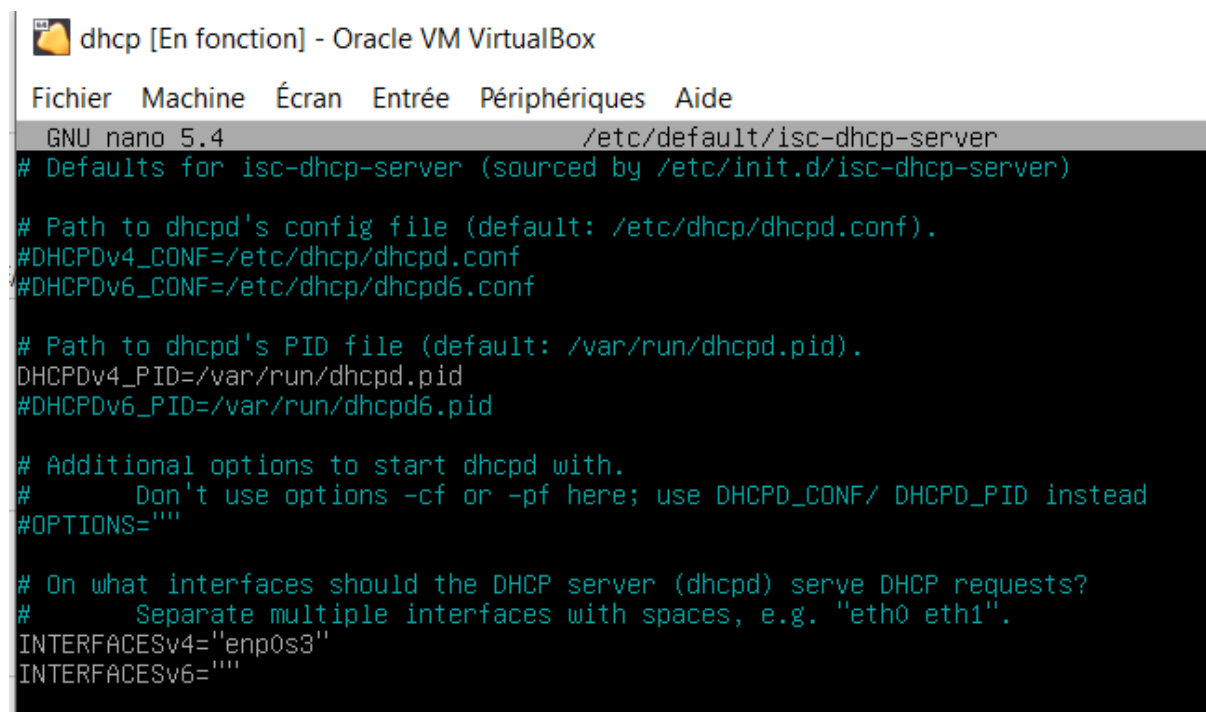
ServerAdmin admin@site2.com_
ServerName unexchange.com
# ServerAlias www.unexchange.com
DocumentRoot /var/www/html/

```

## Service serveurs DHCP

Il faut installer isc-dhcp-server pour ça il faut taper « apt install isc-dhcp-server ».

Une fois le dhcp installé il faut configurer le fichier « nano /etc/default/isc-dhcp-server » il faut décommenter la ligne « dhcpd4\_conf » et ajouter interfaces réseaux sur votre carte.



```

dhcp [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 5.4                                /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

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


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

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

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s3"
INTERFACESv6=""

```

Il faut ensuite configurer le fichier « nano /etc/dhcp/dhcpd.conf » ici on définit la plage adresses et la durée du bail. Il faut ensuite restart le service DHCP.

 dhcp [En fonction] - Oracle VM VirtualBox

Fichier Machine Écran Entrée Périphériques Aide

GNU nano 5.4 /etc/dhcp/dhcpd.conf

```
#class "foo" {
#  match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
#}

#shared-network 224-29 {
#  subnet 10.17.224.0 netmask 255.255.255.0 {
#    option routers rtr-224.example.org;
#  }
#  subnet 10.0.29.0 netmask 255.255.255.0 {
#    option routers rtr-29.example.org;
#  }
#  pool {
#    allow members of "foo";
#    range 10.17.224.10 10.17.224.250;
#  }
#  pool {
#    deny members of "foo";
#    range 10.0.29.10 10.0.29.230;
#  }
#}

default-lease-time 86400 ;
max-lease-time 172800;

subnet 10.242.10.0 netmask 255.255.255.0 {
  range 10.242.10.15 10.242.10.199;
  option domain-name-servers      unexchange.com;
  option routers 10.242.10.0;
}
```

## Service serveur mail Postfix

Pour configurer un serveur mail, il faut installer postfix « apt-get install postfix » une fois installer on tape la commande « dpkg-reconfigure postfix » voici la liste des paramètres à configurer :

Il faudra entrer les paramètre dans l'ordre suivant : « Site internet » / « gmail.ucexchange.com » / « postmaster » / « dns2, smtp.ucexchange.com, dns2, localhost.localdomain », / « localhost » / « non » / « 127.0.0.0/8 » / « 50000 » / « @ » / « IPV4 »

On pourra retrouver ces configurations dans le fichier de configuration /etc/postfix/main.cf