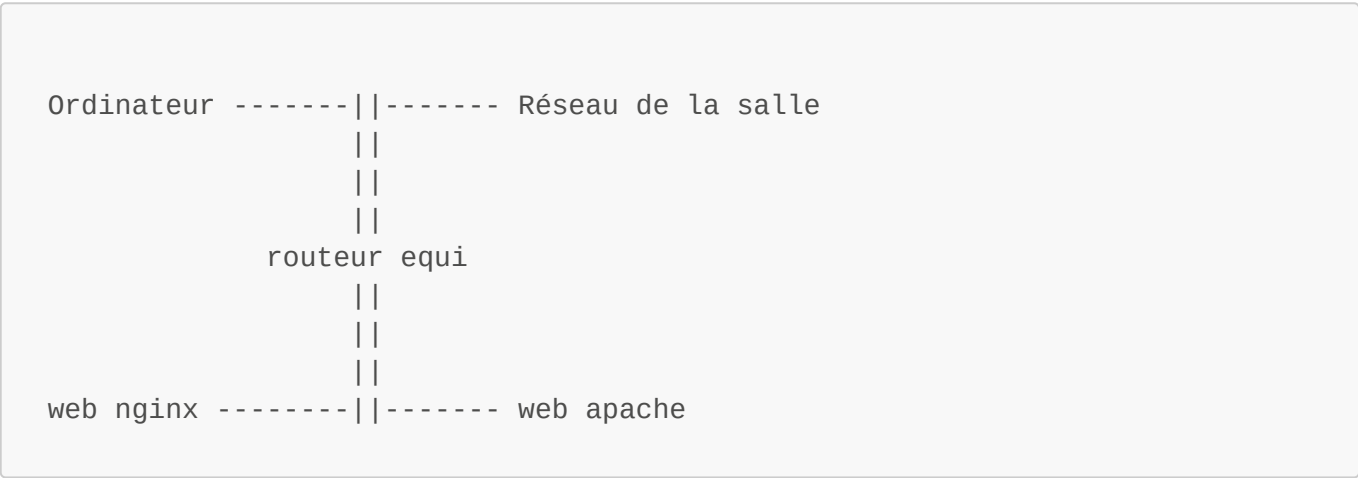


# Equilibreur de charge :

27/05/2024  
Helec Bastien  
Etzweiler Randall

## Préparatifs

La configuration physique sera la suivante :



- Réseau externe : 10.202.0.0/16
- Réseau interne : 192.168.0.0/24

routeur equi :

- pate externe : 10.202.6.1
- pate interne :192.168.0.254

Autres :

- hôte web : 192.168.0.1
- nginx : 192.168.0.80
- apache :192.168.0.81

*NB : Pour les serveurs web on utilisera une machine virtuelle chacune.*

*NB2 : On utilise un dungle usb pour la connexion internet.*

## Commandes :

Les commandes pour que notre PC Linux puisse fonctionner comme un routeur LVS, redirigeant le trafic HTTP entrant vers les serveurs réels sur les VMs sont les suivantes :

Activation du routage IP pour permettre le routage entre les interfaces réseau:

On ouvre le fichier `/etc/sysctl.conf` et ajoutons la ligne `net.ipv4.ip_forward = 1` puis on applique les modifications avec `sudo sysctl -p`.

## Installation des paquets nécessaires pour LVS :

On install `ipvsadm` pour configurer LVS avec :

```
sudo apt-get update
sudo apt-get install ipvsadm
```

## Configuration des interfaces réseau :

On fait `nano etc/network/interfaces` :

```
# Interface externe
auto lo
iface lo inet static
    address 10.202.6.1
    netmask 255.255.0.0

# Interface interne
auto enx00e04cdb0fc6
iface enx00e04cdb0fc6 inet static
    address 192.168.0.254
    netmask 255.255.255.0
```

## Configuration d'LVS avec ipvsadm :

On configure LVS pour rediriger le trafic entrant vers nos serveurs réels avec l'ajout d'un script pour configurer `ipvsadm` nommé `lvs-config.s` tel que :

```
#!/bin/bash

# Activation du forwarding IP
echo 1 > /proc/sys/net/ipv4/ip_forward

# Configuration de l'interface réseau
ifconfig lo:0 10.202.6.1 netmask 255.255.255.0 up

# Nettoyage des configurations précédentes
ipvsadm -C

# Ajout d'une nouvelle table LVS pour la VIP 10.202.6.1
ipvsadm -A -t 10.202.6.1:80 -s rr

# Ajoute des serveurs réels (RIP)
ipvsadm -a -t 10.202.6.1:80 -r 192.168.0.80:80 -m
ipvsadm -a -t 10.202.6.1:80 -r 192.168.0.81:80 -m
```

```
# Configuration iptables
iptables -A INPUT -d 10.202.6.1 -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i enx00e04cdb0fc6 -o lo -j ACCEPT
iptables -A FORWARD -i lo -o enx00e04cdb0fc6 -j ACCEPT
iptables -t nat -A POSTROUTING -o enx00e04cdb0fc6 -j MASQUERADE
iptables -t nat -A POSTROUTING -o lo -j MASQUERADE
iptables -t nat -A PREROUTING -d 10.202.6.1 -p tcp --dport 80 -j DNAT --to-destination 192.168.0.80-192.168.0.81
```

On le rend executable avec `chmod +x lvs-config.sh` et l'exécutons pour appliquer la configuration avec `sudo ./lvs-config.sh`.

Configuration iptables pour permettre le forwarding et le NAT :

## Load balancing

Pour mettre en place un équilibrage de charge avec HAProxy entre nos serveurs Nginx et Apache, on suit ces étapes :

### Installation d'HAProxy

```
sudo apt-get update
sudo apt-get install haproxy
```

### Configuration d'HAProxy

On modifie le fichier de configuration HAProxy, situé à `/etc/haproxy/haproxy.cfg` tel que :

```
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd
listeners
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # tune.ssl.default-dh-param 2048

defaults
    log      global
    mode     http
    option   httplog
    option   dontlognull
    timeout  connect 5000
```

```
timeout client 50000
timeout server 50000
errorfile 400 /etc/haproxy/errors/400.http
errorfile 403 /etc/haproxy/errors/403.http
errorfile 408 /etc/haproxy/errors/408.http
errorfile 500 /etc/haproxy/errors/500.http
errorfile 502 /etc/haproxy/errors/502.http
errorfile 503 /etc/haproxy/errors/503.http
errorfile 504 /etc/haproxy/errors/504.http

frontend http_front
    bind *:80
    default_backend http_back

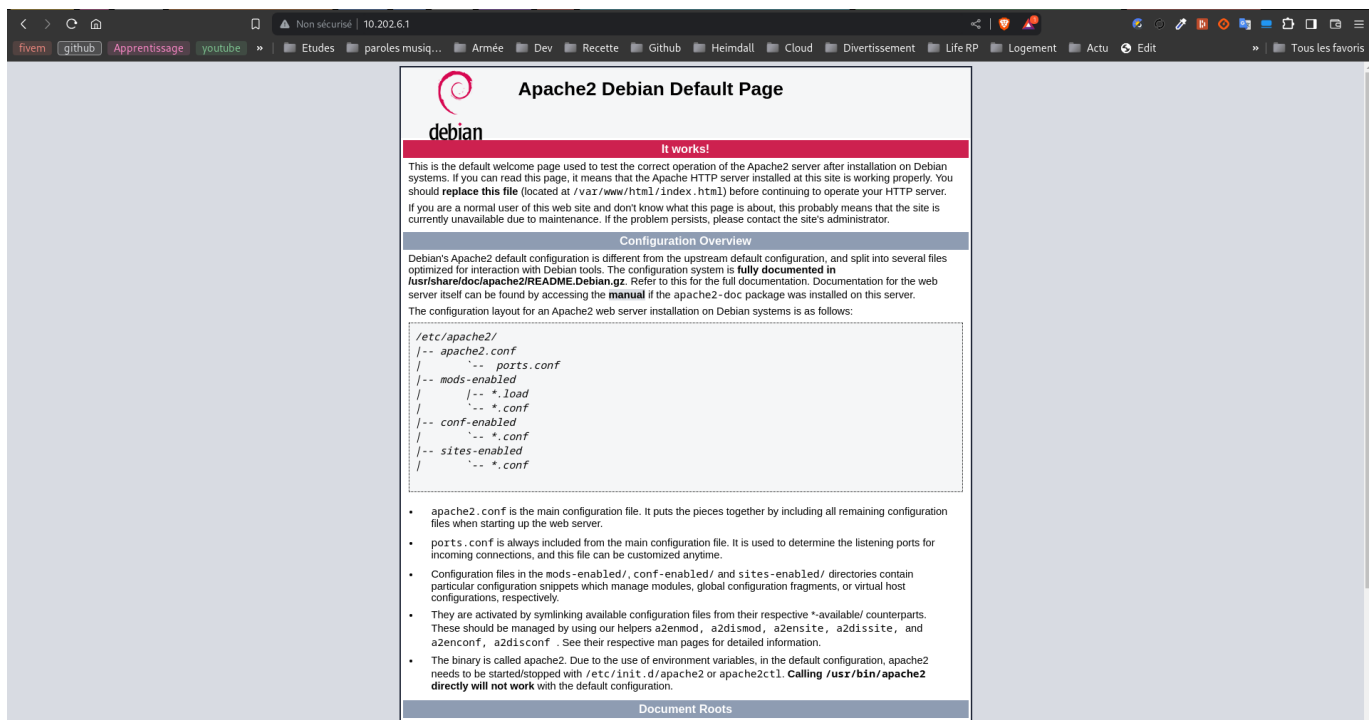
backend http_back
    balance roundrobin
    server nginx 192.168.0.80:80 check
    server apache 192.168.0.81:80 check
```

Vérification et application des changements :

```
sudo haproxy -f /etc/haproxy/haproxy.cfg -c
sudo systemctl restart haproxy
```

## Test final :

En rentrant l'adresse : <http://10.202.6.1:80>



Puis on utilise le load balancing avec HAProxy en rentrant l'adresse : <http://10.202.6.1:80>

On trouve également la meme chose constatement (en inversant les deux dernières lignes du fichier),  
probablement de part la charge qui n'est pas assez importante pour que l'équilibrage soit visible.