Table des matières :

Avant-Propos

Compétences :

- A1.1.1 Analyse du cahier des charges d'un service à produire
- A1.2.4 Déterminer des tests nécessaires à la validation d'un service
- A1.3.2 Définition des éléments nécessaires à la continuité d'un service
- A2.2.3 Réponse à une interruption de service
- A4.1.9 Rédaction d'une documentation technique

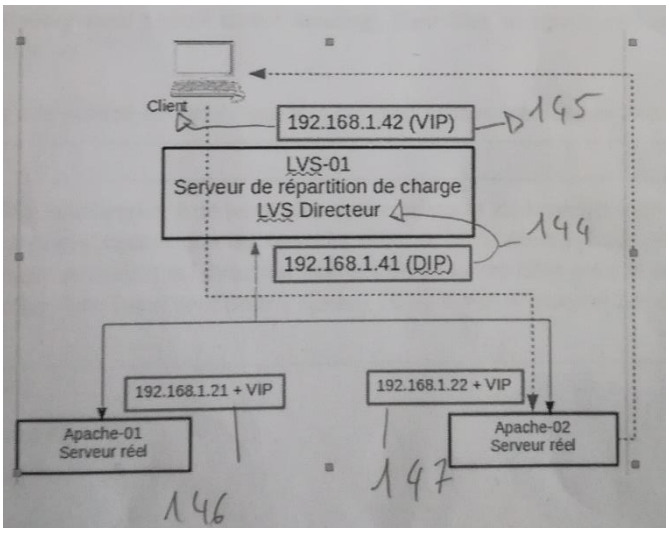
Il s'agit d'un équilibreur de charge permettant non seulement de redonner un service (comme un serveur Apache, un serveur LDAP), mais également d'équilibrer la charge induite des réquètes générées par les clients.

Le serveur LVS est généralement appelé directeur.

Keepalived répond entièrement aux besoins de haute disponibilité en équilibrage de charge, keepalived fournit le nécessaire à la mise en place des services LVS et au contrôle des services en vérifiant la santé des services sur les serveurs réels.

Keepalived utilise vrrp. Cela a deux avantages. Le premier est l'utilisation de plusieurs directeurs si tous utilisent keepalived. Si le maître tombe en panne, un backup prend sa place. Le second avantage est que la configuration de keepalived avec vrrp, y compris sur un directeur seul, met en place l'ensemble des VIP automatiquement sans aucune intervention. Le routeur maitre va automatiquement configurer les IP des LVS et de la passerelle virtuelle au niveau des différentes interfaces réseau sans aucune configuration manuelle de l'administrateur (ifconfig, ifup, ifdown, etc..).

L'utilisation de keepalived permet de se dispense r de l'utilisation des autres commandes de gestion du réseau et du LVS. La configuration du LVS est centralisée au sein d'un fichier à la syntaxe simple et claire. Les modifications des LVS ne nécessitent qu'un rechargement de ce fichier par keepalived sans interruption de service.



Objectif :

Dans cette procédure, nous allons montrer comment installer et configurer un serveur équilibrage de charges au moyen du service Keepalived sous Debian.

OS	Distribution	Version
Debian	Linux	8.5

Configuration directeur

On installe ipvsadm

```
root@mariette:~# apt-get update_
```

« apt-get install ipvsadm »

On saisit ensuite une adresse up virtuelle, il faut la déclarer au niveau de l'interface réseau du directeur.

.144 pour le directeur et .145 pour le client virtuel

```
  GNU nano 2.2.6        Fichier : /etc/network/interfaces

iface eth0 inet static
 address 192.196.1.144
 netmask 255.255.255.0
 gateway 192.168.1.254
# This is an autoconfigured IPv6 interface
iface eth0 inet6 auto

auto bond0
iface bond0 inet static
        address 192.168.1.145
        netmask 255.255.255.255
        broadcast 192.168.1.255
        netwirk 192.168.1.0
```

On ajoute ensuite la ligne suivante dans le fichier :

```
root@directeur:/etc/modprobe.d# nano bond.conf
```

```
  GNU nano 2.2.6              Fichier : bond.conf

alias bond0 bonding
```

On démarre l'interface :

```
root@directeur:/etc/modprobe.d# ifup bond0
```

On ajoute ensuite la commande ci-dessous pour ajouter un service, préciser le protocole tcp, suivi de son adresse ip du service et du port à équilibrer, Enfin l'option –s spécifie la politique de répartition. Dans le cas présent, il s'agit du mode Round Robin (notée rr)

```
root@directeur:/etc/modprobe.d# ipvsadm -A -t 192.168.1.145:80 -s rr
```

Après cela, il faut ajouter les serveurs réels proposant le service web

```
root@directeur:/etc/modprobe.d# ipvsadm -a -t 192.168.1.145:80 -r 192.168.1.146:
80 -g -w 1
```

```
root@directeur:/etc/modprobe.d# ipvsadm -a -t 192.168.1.145:80 -r 192.168.1.147:
80 -g -w 1
```

On peut lister et vérifier la déclaration prise avant :

```
root@directeur:/etc/modprobe.d# ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  192.168.1.145:80 rr
  -> 192.168.1.146:80             Route   1      0          0
  -> 192.168.1.147:80             Route   1      0          0
```

Il ne reste plus qu'à configurer les serveurs réels pour y déclarer l'adresse virtuelle VIP.

Tout d'abord il faut s's'assurer que les serveurs réels ne répondent pas aux requêtes ARP qui leur sont adressées, ça seules les requêtes venant du directeur doivent être interprétées pour le sous-réseau virtuel utilisé.

Pour ce faire, on édite le fichier /etc/sysctl.conf pour y ajouter (si ligne n'existent pas déjà), les enregistrements suivants : A faire dans les 2 serveurs :

```
root@server2:~# nano /etc/sysctl.conf
root@server2:~# nano /etc/sysctl.conf
```

```
  GNU nano 2.2.6            Fichier : /etc/sysctl.conf

#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#

net.ipv4.conf.all.arp_ignore=1
net.ipv4.conf.all.arp_announce=2
net.ipv4.conf.lo.arp_ignore=1
net.ipv4.conf.lo.arp_announce=2
```

```
  GNU nano 2.2.6            Fichier : /etc/sy

#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#

net.ipv4.conf.all.arp_ignore=1
net.ipv4.conf.all.arp_announce=2
net.ipv4.conf.lo.arp_ignore=1
net.ipv4.conf.lo.arp_announce=2
```

On recharge ensuite la configuration du noyau :

```
root@server2:~# sysctl -p
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.ipv4.conf.lo.arp_ignore = 1
net.ipv4.conf.lo.arp_announce = 2
```

```
root@server2:~# sysctl -p
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.ipv4.conf.lo.arp_ignore = 1
net.ipv4.conf.lo.arp_announce = 2
```

On sauvegarde la configuration du directeur :

```
root@directeur:~# ipvsadm -Sn > /etc/ipvsadm_rules
```

On déclare ensuite l'adresse vip dans l'interface loopback

```
  GNU nano 2.2.6        Fichier : /etc/network/interfaces

# This file describes the network interfaces available on yo
# and how to activate them. For more information, see interf

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto lo:0
iface lo:0 inet static
        address 192.168.1.145
        netmask 255.255.255.255_
```

```
root@directeur:~# service networking restart
```

Utilisation de Keepalived

A faire dans les 2 serveurs :

On installe keepallived

```
  GNU nano 2.2.6        Fichier : /etc/network/interfaces

# This file describes the network interfaces available on your syst
# 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
allow-hotplug eth0
iface eth0 inet static
        address 192.168.1.146
        netmask 255.255.255.0
        gateway 192.168.1.254
# This is an autoconfigured IPv6 interface
iface eth0 inet6 auto
```

```
root@server2:~# apt-get update _
```

```
root@server2:~# apt-get install keepalived_
```

On fait pareil sur le 2 ème serveur

```
  GNU nano 2.2.6        Fichier : /etc/network/interfaces

# This file describes the network interfaces available on your syst
# 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
allow-hotplug eth0
iface eth0 inet static
        address 192.168.1.147
        netmask 255.255.255.0
        gateway 192.168.1.254_
```

```
root@server2:~# apt-get update_
```

```
root@server2:~# apt-get install keepalived_
```

Apres avoir installer keepllived sur les deux serveurs :

Il faut aller configurer le fichier keepalived.conf dans « /etc/keepalived »

```
root@server2:/etc/keepalived# nano keepalived.conf_
```

```
  GNU nano 2.2.6                Fichier : /etc/keepalived/keepalived.conf

global_defs {

notification_email { admin@dmn.org
}

notification_email_from admlvs@dmn.org

smtp_server smtp.sio.fr

smtp_connect_timeout 30

router_id LVS_WEB
}

vrrp_sync_group KAD_VRRP {

group{  KEEP1
}

notify /root/adm/notify_ka.sh
}

vrrp_instance KEEP1{

        state BACKUP

        interface eth0

        virtual_router_id 50

        authentication {
        auth_type uadmin
        auth_pass root
}

        nopreempt

        priority 100

vrrp_instance KEEP1{

        state BACKUP

        interface eth0

        virtual_router_id 50

        authentication {
        auth_type udamin
        auth_pass root
}

        nopreempt

        priority 50

 adver-int 1
```

La suite

```
 GNU nano 2.2.6                Fichier : /etc/keepalived/keepalived.conf

 virtual_ipaddres{
 192.168.1.145
 {
}

virtual_server 192.168.1.145 80{
 delay_loop 4
 lb_algo wlc
 lb_kind DR
 persistence_timeout 120
 protocol TCP

 real_server 192.168.1.146 80{

 weight 1

 HTTP_GET{
 url{
        path /service.txt
        digest
        }
        connect_port 80
        connect_timeout 2
        nb_get_retry 1
        delay_before_retry 1
        }
    }
}

real_server 192.168.1.147 80{
 weight 1

 HTTP_GET{
 url{
        path /service.txt
        digest
        }
        connect_port 80
        connect_timeout 2
        nb_get_retry 1
        delay_before_retry 1
        }
   }
}
```

Il faut ensuite aller génère la clé digest à rentrer ensuite dans le document keepalived.conf

```
root@server1:/etc/keepalived# genhash -s 192.168.1.146 -p 80 -u /service.txt -v
----------------------[ HTML hash final resulting ]----------------------
MD5SUM = cb5eccac031b438da3e2558da2751303

Global response time for [/service.txt] =86102
```

Il faut le rentrer à l'emplacement de digest

```
 GNU nano 2.2.6      Fichier : /etc/keepalived/keepalived.conf

        digest cb5eccac031b438da3e2558da2751303_
        }
        connect_port 80
        connect_timeout 2
        nb_get_retry 1
        delay_before_retry 1
        }
    }


real_server 192.168.1.147 80{
 weight 1

 HTTP_GET{
 url{
        path /service.txt
        digest cb5eccac031b438da3e2558da2751303
        }
```

Puisqu'il faut le faire sur les 2 serveurs,

On peut directement transférer le fichier vers une autre machine avec la commande suivante :

```
root@server2:~# scp /etc/keepalived/keepalived.conf root@192.168.1.146:/etc/keep
alived/keepalived.conf
```

On peut maintenant démarrer keepalived

```
root@server1:/etc/keepalived# service keepalived restart
```

```
root@server2:~# service keepalived restart
```