

# Mise en place d'un cluster NAT avec les outils Red Hat Linux

Matthieu Petite

11 décembre 2007

## Résumé

Ce document décrit la mise en place d'un cluster Nat à l'aide des outils fournis par Red Hat. La plate forme de test comporte :

- Un serveur RHEL5 qui jouera le rôle d'**active router**
- Un serveur RHEL5 qui jouera le rôle de **passive router**
- Un serveur RHEL5 qui jouera le rôle de **serveur HTTP 1**
- Un serveur RHEL5 qui jouera le rôle de **serveur HTTP 2**

Dans cette exemple, le service WEB port 80 sera mis sous haute disponibilité.

# 1 Cluster NAT : configuration basic

La configuration minimale d'un cluster consiste à mettre en place une architecture à deux niveaux. Sur le premier, se trouvent deux serveurs LVS, un actif, l'autre passif. Chacun de ces serveurs dispose d'au moins deux interfaces réseaux interconnectée dans le réseau public et dans le réseau privé. Ces deux machines jouent donc le rôle de routeur inter lan. Sur le second niveau se trouvent l'ensemble des serveurs qui offrent le service utile pour les clients du système haute disponibilité (HTTP, FTP, Mail,...).

Dans l'exemple de configuration décrit ci-dessous, la machine LVS active utilise la translation d'adresse (NAT) pour rediriger le trafic depuis la partie public vers les serveurs de services. Ces derniers possèdent donc au moins une interface interconnectée avec réseau privé et communique avec le réseau public à travers le serveur LVS actif.

Les requêtes qui arrivent sur le cluster LVS sont adressées à une **adresse IP virtuelle ou VIP**. Cette adresse est une IP public qui peut être associée à une entrée dans les tables DNS. C'est cette adresse qui migre du routeur actif vers le routeur passif en cas de de failover, garantissant ainsi dans le temps la disponibilité de cette resource IP.

Un seul serveur LVS est actif à un instant donné. Outre le rôle de routage du trafic réseau, ce dernier supervise l'état des serveurs de service et peut le cas échéant en retirer un de la liste des serveurs actifs.

## 2 Configuration IP

### 2.1 Définition des réseaux

Réseau/hote	Adresse IP	Netmask	Gateway	Commentaire
public	192.168.0.0	255.255.255.240	192.168.0.1	réseau public = internet
public	192.168.0.16	255.255.255.240	192.168.0.30	réseau privé
VIP	192.168.0.14	255.255.255.240	192.168.0.30	VIP pour service http
DIP	192.168.0.30	255.255.255.240	192.168.0.30	DIP : gateway pour réseau interne
LVS1 public	192.168.0.3	255.255.255.240	192.168.0.1	
LVS2 public	192.168.0.4	255.255.255.240	192.168.0.1	
LVS1 privée	192.168.0.19	255.255.255.240	192.168.0.30	
LVS2 privée	192.168.0.20	255.255.255.240	192.168.0.30	
Service 1 privée	192.168.0.21	255.255.255.240	192.168.0.30	
Service 2 privée	192.168.0.22	255.255.255.240	192.168.0.30	

Pour la plate forme de test, les adresses IP sont portées par des VLAN (2 pour le réseau public, 3 pour le réseau privé) . Cette particularité est due à l'architecture technique à disposition lors de la mise en place des serveurs mais n'est en aucun cas une obligation. Il faudra donc penser à modifier les interfaces à travers les différents fichiers de configuration.

## 2.2 Active Routeur

### Configuration eth0.2

modifier le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0.2` comme suit :

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0.2
BOOTPROTO=static
BROADCAST=192.168.0.15
HWADDR=00:0C:29:70:99:0C #set your mac address
IPADDR=192.168.0.3
NETMASK=255.255.255.240
NETWORK=192.168.0.0
ONBOOT=yes
```

### Configuration eth0.3

modifier le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0.3` comme suit :

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0.3
BOOTPROTO=static
BROADCAST=192.168.0.31
HWADDR=00:0C:29:70:99:0C #set your mac address
IPADDR=192.168.0.19
NETMASK=255.255.255.240
NETWORK=192.168.0.16
ONBOOT=yes
```

## Redirection de packet

modifier le fichier /etc/sysctl.conf pour activer l'ip forwarding :

```
# Controls IP packet forwarding
net.ipv4.ip_forward = 1

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality
# of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core
filename # Useful for debugging multi-threaded applications
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 1

#Controls the maximum size of a message, in bytes
kernel.msgmnb=65536

# Controls the default maximum size of a message queue
kernel.msgmax = 65536

# Controls the maximum shared segment size, in bytes
kernel.shmmax = 4294967295

# Controls the maximum number of shared memory segments,
# in pages
kernel.shmall = 268435456
```

## 2.3 Passive Routeur

### Configuration eth0.2

modifier le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0.2` comme suit :

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0.2
BOOTPROTO=static
BROADCAST=192.168.0.15
HWADDR=00:0C:29:70:99:0C #set your mac address
IPADDR=192.168.0.4
NETMASK=255.255.255.240
NETWORK=192.168.0.0
ONBOOT=yes
```

### Configuration eth0.3

modifier le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0.3` comme suit :

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0.3
BOOTPROTO=static
BROADCAST=192.168.0.31
HWADDR=00:0C:29:70:99:0C #set your mac address
IPADDR=192.168.0.20
NETMASK=255.255.255.240
NETWORK=192.168.0.16
ONBOOT=yes
```

### Redirection de packet

Faire la même modification que pour le serveur actif

## 2.4 Serveur HTTP 1

### Configuration eth0

Modifier le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0.3` comme suit :

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0.3
BOOTPROTO=static
BROADCAST=192.168.0.31
HWADDR=00:0C:29:70:99:0C #set your mac address
IPADDR=192.168.0.21
NETMASK=255.255.255.240
NETWORK=192.168.0.16
ONBOOT=yes
```

## 2.5 Serveur HTTP 2

### Configuration eth0

modifier le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0.3` comme suit :

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0.3
BOOTPROTO=static
BROADCAST=192.168.0.31
HWADDR=00:0C:29:70:99:0C #set your mac address
IPADDR=192.168.0.22
NETMASK=255.255.255.240
NETWORK=192.168.0.16
ONBOOT=yes
```

## 2.6 Prise en compte des modifications

Pour que la configuration réseau soit prise en compte redémarrer le service réseau sur l'ensemble des machines au moyen de la commande ci-dessous :

```
service network restart
```

## 3 Installation des logiciels

Le cluster fourni par Red Hat repose sur l'installation de deux packages ://

- piranha
- piranha-gui (uniquement sur le nœud actif)

Le package piranha doit installer le logiciel ipvsadm si ce dernier n'était pas présent auparavant.

Ces package se trouvent sur le DVD d'installation du système RHEL5 dans la rubrique Cluster. Une fois les repository bien configurés, l'installation à l'aide de yum est la plus aisée.

Le package piranha-gui nécessite l'installation du service httpd.

```
yum install httpd
```

## 4 Post installation

### Activation des services au démarrage

Lancer les commandes suivantes sur chacun des noeuds LVS :

```
chkconfig pulse --level 35 on
chkconfig iptables --level 35 on
chkconfig piranha-gui --level 35 on #sur le noeud principal
```

### Mot de passe pour l'interface piranha

Pour utiliser l'interface web de configuration du service piranha, il faut y associer un mot de passe. Pour se faire lancer la commande suivante :

```
/usr/sbin/piranha-passwd
```

### Démarrage des service

Sur le noeud actif, lancer les commandes suivantes :

```
service piranha-gui start
service pulse start
```

## 5 Configuration du cluster

### 5.1 Configuration du cluster (fichier lvs.d)

La configuration du service cluster se fait par l'intermédiaire du fichier lvs.cf situé dans le répertoire /etc/syconfig/ha.

```
serial_no = 174          #id du cluster (unique dans le réseau)
primary = 192.168.0.3   #IP public du n|{oe} ud principal
primary_private = 192.168.0.19 #IP privée du n|oe ud principal
service = lvs
backup_active = 1      #Activation / désactivation du n|oe ud passif
backup = 192.168.0.4   #IP public du noeud passif
backup_private = 192.168.0.20 #IP privée du noeud passif
heartbeat = 1          #activation du check heartbeat
heartbeat_port = 539   #port UDP heartbeat
keepalive = 6          #délai entre deux check
deadtime = 18          #délai avant perte du second noeud
network = nat           #type de cluster: ici nat
nat_router = 192.168.0.30 eth0.3:1 #adresse DIP
nat_nmask = 255.255.255.240
debug_level = NONE
monitor_links = 0
#creation du service http haute disponibilité
virtual [CLUSTER_HTTP] {
    active = 1
    address = 192.168.0.14 eth0.2:1 #adresse VIP
    vip_nmask = 255.255.255.240
    port = 80 #port du service
    #parametre de check des service HTTP
    send = "GET_/_HTTP/1.0\r\n\r\n"
    #résultat attendu
    expect = "HTTP/1.1_200_OK"
    use_regex = 0
    load_monitor = none
    scheduler = rr      #nous sommes en mode Round Robin
    protocol = tcp
    timeout = 6         #délai avant perte
    reentry = 15        #durée avant retour actif d'un
                        #serveur HTTP

    quiesce_server = 0
    #liste des serveurs réels
    server HTTP1 {
        address = 192.168.0.21
        active = 1
        weight = 1
    }
    server HTTP2 {
        address = 192.168.0.22
```

```

        active = 1
        weight = 1
    }
}

```

## 5.2 Configuration du firewall sur les serveur LVS

```

*nat
:PREROUTING ACCEPT [122:8957]
:POSTROUTING ACCEPT[12:864]
:OUTPUT ACCEPT [12:864]
COMMIT

*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [3431:364885]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
#accept multicast trafic , not used here but just in case
-A RH-Firewall-1-INPUT -d 224.0.0.251 -p udp -m udp --dport 5353 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 1823 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
#authorized connection from public to private on port 80
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 80 -j ACCEPT
#accept UDP trafic for heartbeat check
-A RH-Firewall-1-INPUT -s 192.168.0.4/28 -p udp -m udp -j ACCEPT
-A RH-Firewall-1-INPUT -s 192.168.0.3/28 -p udp -m udp -j ACCEPT
#accept trafic from web server to public network
-A RH-Firewall-1-INPUT -s 192.168.0.22/28 -p tcp -m tcp --sport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -s 192.168.0.21/28 -p tcp -m tcp --sport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 3636 -j ACCEPT
#log rules , remaining here just in case
#-A RH-Firewall-1-INPUT -j LOG --log-prefix "[Rejected packet]"
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT

```

## 5.3 redémarrage des services

Après toutes modifications des fichiers iptables et lvs.cf, un redémarrage des services est nécessaire :

```
service pulse restart
service iptables restart
```

A ce stade, doit rediriger le trafic réseau vers les deux serveurs de service sur le port 80. Si les serveurs web sont actifs, il suffira de lancer la commande suivante sur le noeud actif pour vérifier que le cluster est opérationnel.

```
ipvsadm
```

Ouvrez un browser web sur une machine du réseau et tenter de joindre l'adresse `http://vip:80/`.

## 6 Synchronisation des configurations

### 6.1 Synchronisation via scp

Sur le noeud secondaire, il faut synchroniser la configuration. Une copie manuelle est possible mais il est préférable de mettre en place une synchronisation régulière par scp.

Pour cela, il est nécessaire de créer une clé asymétrique sur le noeud secondaire puis de la copier sur le noeud principal :

```
cd /root
ssh-keygen -f cluster
#valider toutes les étapes, les fichiers cluster et
#cluster.pub doivent être créés.
scp /root/cluster.pub root@192.168.0.3:/root/.ssh/cluster.pub
#entrer le password du premier noeuds
#puis se connecter sur le noeud principal
ssh root@192.168.0.3
#entrer le password
cd ../ssh
cat cluster.pub >> authorized_keys
exit
```

Une fois la clé mise à jour sur le serveur primaire, créer un fichier `synchro_cluster.sh` dans le répertoire `/usr/local/bin` contenant :

```
#!/bin/bash
scp -i /root/rsa
root@192.168.0.3:/etc/sysconfig/ha/lvs.cf
/etc/sysconfig/ha/lvs.cf.tmp scp -i /root/rsa
root@192.168.0.3:/etc/sysconfig/iptables /etc/sysconfig/iptables.tmp

diff /etc/sysconfig/ha/lvs.cf /etc/sysconfig/ha/lvs.cf.tmp &>
```

```
/dev/null if [ $? -ne 0 ] then
  rm -f /etc/sysconfig/ha/lvs.cf
  mv /etc/sysconfig/ha/lvs.cf.tmp /etc/sysconfig/ha/lvs.cf
  service pulse restart
fi

diff /etc/sysconfig/iptables /etc/sysconfig/iptables.tmp &>
/dev/null if [ $? -ne 0 ] then
  rm -f /etc/sysconfig/iptables
  mv /etc/sysconfig/iptables.tmp /etc/sysconfig/iptables
  service iptables restart
fi
```

## 6.2 configuration crond

### Synchronisation lvs.cf

Modification de la table cron pour synchroniser le fichier lvs.cf

```
crontab -e
5,15,25,35,45,55 * * * * /usr/local/bin/synchro_cluster.sh
```

## 7 Interface graphique

Il est possible de gérer la configuration du cluster via une interface graphique. Une fois le service piranha démarré, il suffit de se connecter sur l'URL `http://192.168.0.3:3636` pour accéder à l'application.