Dernière mise-à-jour : 2021/01/24 12:16

HAR300 - Gestion du Linux Virtual Server

Présentation

Sous Linux, la répartition de charge est obtenue en mettant en place un **LVS** (*Linux Virtual Serveur*). La répartition de charge est gérée par un **directeur** (*Load Balancer*). L'ensemble du LVS est vu de l'extérieur comme un seul serveur.

Nomenclature

Le vocabulaire du LVS est résumé dans le tableau suivant :

Terme	**Description**
VIP	L'adresse IP virtuelle du LVS
RIP	L'adresse IP réelle d'un serveur
DIP	L'adresse IP réelle du directeur
Passerelle Virtuelle	Une passerelle sur le directeur utilisée dans un LVS en NAT
VRRP (*Virtual Router Redundancy Protocol*)	Protocole qui établit et qui régit un ou plusieurs routeurs virtuels
Poids	Valeur de charge pour un serveur réel

Types de LVS

Il existe trois types de LVS :

LVS-NAT

Un LVS-NAT utilise un Directeur comme passerelle virtuelle entre des serveurs réels et les clients.

LVS-DR

Le LVS-DR ou ***Direct Routing*** est une méthode ou le Directeur ne traite que les requêtes en entrée. Les serveurs réels répondent directement aux clients.

LVS-TUN

LE LVS-TUN ou **tunnel IP** fonctionne de la même façon que le LVS-DR mais la communication entre le Directeur et les serveurs réels se fait via des tunnels.

Préparation

Démarrez une machine virtuelle CentOS vierge, puis :

- supprimez le service **iptables** avec la commande **chkconfig –del iptables**,
- configurez **SELinux** en mode permissive avec la commande **setenforce permissive**,
- modifiez la valeur de la directive **SELINUX** du fichier **/etc/selinux/config** pour que celle-ci soit **permissive**.

Créez ensuite 5 clones complets et configurez-les ainsi :

Nom de la VM	**RAM**
client	512 Mo
directeur	512 Mo
secours	512 Mo
serveur1	512 Mo
serveur2	512 Mo

Lors de la création des clones, veillez à réinitialiser l'adresse MAC de toutes les cartes.

Démarrez chaque machine virtuelle, puis :

- supprimez toutes les lignes du fichier **/etc/udev/rules.d/70-persistent-net.rules**,
- Arrêtez la machine virtuelle.

Configuration du client

Avant de démarrer la machine **client**, configurez le premier adaptateur réseau de la machine virtuelle en Réseau Interne **Intnet**.

Démarrez la machine et ouvrez un terminal. Arrêtez et supprimez le service NetworkManager puis activez le service **network** :

```
[root@centos6 ~]# service NetworkManager stop
Arrêt du démon NetworkManager :                            [  OK  ]
[root@centos6 ~]# chkconfig --del NetworkManager
[root@centos6 ~]# chkconfig --list network
network         0:arrêt    1:arrêt    2:marche    3:marche    4:marche    5:marche    6:arrêt
```

Éditez le fichier **/etc/sysconfig/network-scripts/ifcfg-eth0** :

ifcfg-eth0

```
DEVICE="eth0"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
```

```
IPV6INIT=no
NETMASK=255.255.255.0
IPADDR=10.0.2.15
GATEWAY=10.0.2.2
USERCTL=yes
```

Modifiez le fichier **/etc/sysconfig/network** :

network

```
NETWORKING=yes
HOSTNAME=client.i2tch.loc
```

Redémarrez la machine virtuelle :

```
[root@centos6 ~]# reboot
```

Configuration du serveur1

Démarrez la machine et ouvrez un terminal. Installez le paquet **httpd** :

```
[root@centos6 ~]# yum install httpd
```

Arrêtez la machine **serveur1** et configurez le premier adaptateur réseau de la machine virtuelle en Réseau Interne **Intnet1**.

Démarrez la machine et ouvrez un terminal. Arrêtez et supprimez le service NetworkManager puis activez le service **network** :

```
[root@centos6 ~]# service NetworkManager stop
Arrêt du démon NetworkManager :                            [  OK  ]
[root@centos6 ~]# chkconfig --del NetworkManager
[root@centos6 ~]# chkconfig --list network
network         0:arrêt    1:arrêt    2:marche    3:marche    4:marche    5:marche    6:arrêt
```

Éditez le fichier **/etc/sysconfig/network-scripts/ifcfg-eth0** :

ifcfg-eth0

```
DEVICE="eth0"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
IPV6INIT=no
NETMASK=255.255.255.0
NETWORK=10.0.3.0
IPADDR=10.0.3.100
GATEWAY=10.0.3.200
USERCTL=yes
```

Modifiez le fichier **/etc/sysconfig/network** :

network

```
NETWORKING=yes
HOSTNAME=serveur1.i2tch.loc
```

Créez une règle de routage statique sur serveur1 :

```
[root@centos6 ~]# service network restart
[root@centos6 ~]# route del default
[root@centos6 ~]# route add default gw 10.0.3.200
```

Configurez le lancement automatique du serveur **httpd** :

```
[root@centos6 ~]# chkconfig --level 345 httpd on
```

Éditez le fichier **/var/www/html/index.html** :

index.html.server1

```
<html>
<title>
Serveur1
</title>
<body>
<center>Ceci est le serveur 1</center>
</body>
</html>
```

Configurez le serveur httpd pour que celui-ci démarre sur le port **8080** en éditant le fichier **/etc/httpd/conf/httpd.conf** :

```
...
Listen 8080
...
ServerName www.example.com:8080
...
```

Redémarrez la machine virtuelle :

```
[root@centos6 ~]# reboot
```

Configuration du serveur2

Démarrez la machine et ouvrez un terminal. Installez le paquet **httpd** :

```
[root@centos6 ~]# yum install httpd
```

Arrêtez la machine **serveur2** et configurez le premier adaptateur réseau de la machine virtuelle en Réseau Interne **Intnet1**.

Démarrez la machine et ouvrez un terminal. Arrêtez et supprimez le service NetworkManager puis activez le service **network** :

```
[root@centos6 ~]# service NetworkManager stop
Arrêt du démon NetworkManager :                            [  OK  ]
[root@centos6 ~]# chkconfig --del NetworkManager
[root@centos6 ~]# chkconfig --list network
network         0:arrêt    1:arrêt    2:marche    3:marche    4:marche    5:marche    6:arrêt
```

Éditez le fichier **/etc/sysconfig/network-scripts/ifcfg-eth0** :

ifcfg-eth0

```
DEVICE="eth0"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
IPV6INIT=no
NETMASK=255.255.255.0
NETWORK=10.0.3.0
IPADDR=10.0.3.150
GATEWAY=10.0.3.200
```

```
USERCTL=yes
```

Modifiez le fichier **/etc/sysconfig/network** :

network

```
NETWORKING=yes
HOSTNAME=serveur2.i2tch.loc
```

Créez une règle de routage statique sur serveur2 :

```
[root@centos6 ~]# service network restart
[root@centos6 ~]# route del default
[root@centos6 ~]# route add default gw 10.0.3.200
```

Configurez le lancement automatique du serveur **httpd** :

```
[root@centos6 ~]# chkconfig --level 345 httpd on
```

Éditez le fichier **/var/www/html/index.html** :

index.html.server2

```
<html>
<title>
Serveur2
</title>
<body>
<center>Ceci est le serveur 2</center>
</body>
```

```
</html>
```

Configurez le serveur httpd pour que celui-ci démarre sur le port **8080** en éditant le fichier **/etc/httpd/conf/httpd.conf** :

```
...
Listen 8080
...
ServerName www.example.com:8080
...
```

Redémarrez la machine virtuelle :

```
[root@centos6 ~]# reboot
```

Configuration du directeur

Démarrez la machine et ouvrez un terminal. Installez le paquet **ipvsadm** :

```
[root@centos6 ~]# yum install ipvsadm
```

Arrêtez la machine **directeur** et configurez le premier adaptateur réseau de la machine virtuelle en Réseau Interne **Intnet** et le deuxième adaptateur réseau en Réseau Interne **Intnet1**.

Démarrez la machine et ouvrez un terminal. Arrêtez et supprimez le service NetworkManager puis activez le service **network** :

```
[root@centos6 ~]# service NetworkManager stop
Arrêt du démon NetworkManager :                            [  OK  ]
[root@centos6 ~]# chkconfig --del NetworkManager
```

```
[root@centos6 ~]# chkconfig --list network
network         0:arrêt    1:arrêt    2:marche    3:marche    4:marche    5:marche    6:arrêt
```

Éditez le fichier **/etc/sysconfig/network-scripts/ifcfg-eth0** :

ifcfg-eth0

```
DEVICE="eth0"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
IPV6INIT=no
NETMASK=255.255.255.0
IPADDR=10.0.2.50
USERCTL=yes
```

Éditez le fichier **/etc/sysconfig/network-scripts/ifcfg-eth1** :

ifcfg-eth1

```
DEVICE="eth1"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
IPV6INIT=no
NETMASK=255.255.255.0
IPADDR=10.0.3.15
USERCTL=yes
```

Configurez la VIP 10.0.2.100 sur l'adaptateur **eth0** en créant le fichier **/etc/sysconfig/network-scripts/ifcfg-eth0:0**

ifcfg-eth0:0

```
DEVICE="eth0:0"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
IPV6INIT=no
NETWORK=10.0.2.0
NETMASK=255.255.255.0
IPADDR=10.0.2.100
USERCTL=yes
```

Configurez la passerelle virtuelle 10.0.3.200 sur l'adaptateur **eth1** en créant le fichier **/etc/sysconfig/network-scripts/ifcfg-eth1:0**

ifcfg-eth1:0

```
DEVICE="eth1:0"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
IPV6INIT=no
NETWORK=10.0.3.0
NETMASK=255.255.255.0
IPADDR=10.0.3.200
USERCTL=yes
```

Modifiez le fichier **/etc/sysconfig/network** :

network

```
NETWORKING=yes
HOSTNAME=directeur.i2tch.loc
```

Configurez le système pour que les modules **ip_vs**, **ip_vs_rr** et **ip_vs_wrr** soient chargés au démarrage :

```
[root@centos6 ~]# echo modprobe ip_vs >> /etc/rc.modules
[root@centos6 ~]# echo modprobe ip_vs_rr >> /etc/rc.modules
[root@centos6 ~]# echo modprobe ip_vs_wrr >> /etc/rc.modules
[root@centos6 ~]# chmod +x /etc/rc.modules
```

Éditez le fichier **/etc/sysctl.conf** ainsi :

```
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

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

puis rechargez le fichier modifié :

```
[root@centos6 ~]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
error: "net.bridge.bridge-nf-call-ip6tables" is an unknown key
error: "net.bridge.bridge-nf-call-iptables" is an unknown key
```

```
error: "net.bridge.bridge-nf-call-arptables" is an unknown key
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 4294967295
kernel.shmall = 268435456
```

Redémarrez la machine virtuelle :

```
[root@centos6 ~]# reboot
```

LAB #1 - Configuration Manuelle d'un LVS

La Commande ipvsadm

La commande **ipvsadm** est utilisée pour administrer les LVS. Les options de cette commande sont :

```
[root@centos6 ~]# ipvsadm --help
ipvsadm v1.25 2008/5/15 (compiled with popt and IPVS v1.2.1)
Usage:
  ipvsadm -A|E -t|u|f service-address [-s scheduler] [-p [timeout]] [-M netmask]
  ipvsadm -D -t|u|f service-address
  ipvsadm -C
  ipvsadm -R
  ipvsadm -S [-n]
  ipvsadm -a|e -t|u|f service-address -r server-address [options]
  ipvsadm -d -t|u|f service-address -r server-address
  ipvsadm -L|l [options]
  ipvsadm -Z [-t|u|f service-address]
  ipvsadm --set tcp tcpfin udp
  ipvsadm --start-daemon state [--mcast-interface interface] [--syncid sid]
  ipvsadm --stop-daemon state
```

```
  ipvsadm -h

Commands:
Either long or short options are allowed.
  --add-service     -A        add virtual service with options
  --edit-service    -E        edit virtual service with options
  --delete-service  -D        delete virtual service
  --clear           -C        clear the whole table
  --restore         -R        restore rules from stdin
  --save            -S        save rules to stdout
  --add-server      -a        add real server with options
  --edit-server     -e        edit real server with options
  --delete-server   -d        delete real server
  --list            -L|-l     list the table
  --zero            -Z        zero counters in a service or all services
  --set tcp tcpfin udp        set connection timeout values
  --start-daemon              start connection sync daemon
  --stop-daemon               stop connection sync daemon
  --help            -h        display this help message

Options:
  --tcp-service  -t service-address   service-address is host[:port]
  --udp-service  -u service-address   service-address is host[:port]
  --fwmark-service  -f fwmark         fwmark is an integer greater than zero
  --ipv6         -6                   fwmark entry uses IPv6
  --scheduler    -s scheduler         one of rr|wrr|lc|wlc|lblc|lblcr|dh|sh|sed|nq,
                                      the default scheduler is wlc.
  --persistent   -p [timeout]         persistent service
  --netmask      -M netmask           persistent granularity mask
  --real-server  -r server-address    server-address is host (and port)
  --gatewaying   -g                   gatewaying (direct routing) (default)
  --ipip         -i                   ipip encapsulation (tunneling)
  --masquerading -m                   masquerading (NAT)
  --weight       -w weight            capacity of real server
```

```
  --u-threshold  -x uthreshold        upper threshold of connections
  --l-threshold  -y lthreshold        lower threshold of connections
  --mcast-interface interface         multicast interface for connection sync
  --syncid sid                        syncid for connection sync (default=255)
  --connection   -c                   output of current IPVS connections
  --timeout                           output of timeout (tcp tcpfin udp)
  --daemon                            output of daemon information
  --stats                             output of statistics information
  --rate                              output of rate information
  --exact                             expand numbers (display exact values)
  --thresholds                        output of thresholds information
  --persistent-conn                   output of persistent connection info
  --nosort                            disable sorting output of service/server entries
  --sort                              does nothing, for backwards compatibility
  --ops          -O                   one-packet scheduling
  --numeric      -n                   numeric output of addresses and ports
```

Commencez par configurer la VIP dans le LVS :

```
[root@centos6 ~]# ipvsadm -A -t 10.0.2.100:80 -s rr
```

Configurez ensuite les deux serveurs NAT :

```
[root@centos6 ~]# ipvsadm -a -t 10.0.2.100:80 -r 10.0.3.100:8080 -m -w 1
[root@centos6 ~]# ipvsadm -a -t 10.0.2.100:80 -r 10.0.3.150:8080 -m -w 1
```

Vérifiez votre configuration avec la commande **ipvsadm** sans options :

```
[root@centos6 ~]# ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.0.2.100:http rr
  -> 10.0.3.100:webcache          Masq    1      0          0
```

```
  -> 10.0.3.150:webcache          Masq    1      0          0
```

Notez la présence du mot **Masq** qui indique que nous avons un LVS-NAT.

Testez le LVS-NAT

A partir du navigateur web de votre machine virtuelle **client**, ouvrez l'adresse http://10.0.2.100.

Selon le cas vous obtiendrez une page avec le texte :

```
Ceci est le serveur 1
```

ou

```
Ceci est le serveur 2
```

Rafraichissez la page est vous obtiendrez, selon la cas :

```
Ceci est le serveur 2
```

ou

```
Ceci est le serveur 1
```

Notez que chaque serveur répond à tour de rôle. Ceci atteste du bon fonctionnement du LVS-NAT en utilisant une **répartition** de type **round-robin**.

Lancez maintenant la commande suivante dans une console du directeur :

```
[root@centos6 ~]# ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.0.2.100:80 rr
  -> 10.0.3.100:8080              Masq    1      0          3
  -> 10.0.3.150:8080              Masq    1      0          3
```

Notez que les valeurs de la colonne **InActConn** ont augmentées. La colonne **InActConn** indique le nombre de connexions établi par des clients dans tous les états, sauf l'état **ESTABLISHED**.

Connexions Persistantes et le Timeout TCP

Persistance

Afin d'éviter qu'un client soit baladé d'un serveur ver un autre, il est possible de configurer la persistance des connexions. Saisissez donc cette commande dans la machine virtuelle **directeur** :

```
[root@centos6 ~]# ipvsadm -E -t 10.0.2.100:80 -s rr -p 600
```

A partir du navigateur web de votre machine virtuelle **client**, ouvrez l'adresse http://10.0.2.100.

Vous noterez que cette fois-ci, votre connexion à un des deux serveurs est persistante. Cette information est visible à partir du directeur :

```
*[root@centos6 ~]# ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
```

```
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.0.2.100:80 rr persistent 600
  -> 10.0.3.100:8080              Masq    1      0          0
  -> 10.0.3.150:8080              Masq    1      0          22
```

Timeout TCP

Le *idle timeout* d'une connexion TCP en statut ESTABLISHED est de 15 minutes par défaut. Afin de permettre le traitement de requêtes lentes, cette valeur peut être modifiée en utilisant la commande **ipvsadm** :

```
[root@centos6 ~]# ipvsadm --set 1800 0 0
```

Le premier chiffre est le nouveau *idle timeout* d'une connexion TCP exprimé en secondes. Les deux autres valeurs de 0 ne modifient pas la configuration courante du *timeout* d'une session TCP après la réception d'un paquet FIN dans la deuxième colonne et le timeout des paquets UDP pour la troisième colonne.

La Table des Connexions

La Table des Connexions, aussi appelée la **table de hash** peut être visualiser en utilisant le commande suivante :

```
[root@centos6 ~]# ipvsadm -Lnc
IPVS connection entries
pro expire state       source             virtual            destination
TCP 01:53  TIME_WAIT   10.0.2.15:53598    10.0.2.100:80      10.0.3.150:8080
TCP 09:55  NONE        10.0.2.15:0        10.0.2.100:80      10.0.3.150:8080
TCP 01:55  TIME_WAIT   10.0.2.15:53602    10.0.2.100:80      10.0.3.150:8080
TCP 01:55  TIME_WAIT   10.0.2.15:53603    10.0.2.100:80      10.0.3.150:8080
TCP 01:54  TIME_WAIT   10.0.2.15:53600    10.0.2.100:80      10.0.3.150:8080
TCP 01:53  TIME_WAIT   10.0.2.15:53597    10.0.2.100:80      10.0.3.150:8080
TCP 01:46  TIME_WAIT   10.0.2.15:53596    10.0.2.100:80      10.0.3.150:8080
TCP 01:54  TIME_WAIT   10.0.2.15:53601    10.0.2.100:80      10.0.3.150:8080
```

```
TCP 01:54  TIME_WAIT   10.0.2.15:53599    10.0.2.100:80      10.0.3.150:8080
```

LAB #2 - Configuration d'un LVS avec Piranha

Présentation

Piranha est un outil web écrit en PHP pour la configuration de LVS. L'outil est spécifique aux distributions Redhat, CentOS et Fedora.

Configuration du secours

Arrêtez la machine virtuelle **directeur**. Démarrez la machine **secours** et installez le paquet **piranha** :

```
[root@centos6 ~]# yum install piranha
```

Configurez le service de piranha :

```
[root@centos6 ~]# chkconfig piranha-gui on
[root@centos6 ~]# chkconfig --list piranha-gui
piranha-gui     0:arrêt    1:arrêt    2:marche    3:marche    4:marche    5:marche    6:arrêt
```

Arrêtez la machine **secours** et configurez le premier adaptateur réseau de la machine virtuelle **secours** en Réseau Interne **Intnet** et le deuxième adaptateur réseau en Réseau Interne **Intnet1**.

Démarrez la machine **secours** et ouvrez un terminal. Configurez ensuite le mot de passe de l’administration Piranha :

```
[root@centos6 ~]# piranha-passwd
New Password: fenestros
```

```
Verify: fenestros
Adding password for user piranha
```

Arrêtez et supprimez le service NetworkManager puis activez le service **network** :

```
[root@centos6 ~]# service NetworkManager stop
Arrêt du démon NetworkManager :                            [  OK  ]
[root@centos6 ~]# chkconfig --del NetworkManager
[root@centos6 ~]# chkconfig --list network
network         0:arrêt    1:arrêt    2:marche    3:marche    4:marche    5:marche    6:arrêt
```

Éditez le fichier **/etc/sysconfig/network-scripts/ifcfg-eth0** :

ifcfg-eth0

```
DEVICE="eth0"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
IPV6INIT=no
NETMASK=255.255.255.0
IPADDR=10.0.2.51
USERCTL=yes
```

Éditez le fichier **/etc/sysconfig/network-scripts/ifcfg-eth1** :

ifcfg-eth1

```
DEVICE="eth1"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
```

```
BOOTPROTO=static
IPV6INIT=no
NETMASK=255.255.255.0
IPADDR=10.0.3.16
USERCTL=yes
```

Créez le fichier **/etc/sysconfig/network-scripts/ifcfg-eth0:0** :

ifcfg-eth0:0

```
DEVICE="eth0:0"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
IPV6INIT=no
NETWORK=10.0.2.0
NETMASK=255.255.255.0
IPADDR=10.0.2.100
USERCTL=yes
```

Créez le fichier **/etc/sysconfig/network-scripts/ifcfg-eth1:0** :

ifcfg-eth1:0

```
DEVICE="eth1:0"
NM_CONTROLLED="no"
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
IPV6INIT=no
NETWORK=10.0.3.0
```

```
NETMASK=255.255.255.0
IPADDR=10.0.3.200
USERCTL=yes
```

Modifiez le fichier **/etc/sysconfig/network** :

network

```
NETWORKING=yes
HOSTNAME=secours.i2tch.loc
```

Modifiez le fichier **/etc/hosts** :

hosts

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.2.100  secours.i2tch.loc
```

Modifiez le fichier **/etc/httpd/conf/httpd.conf** :

httpd.conf

```
...
# ServerName gives the name and port that the server uses to identify itself.
ServerName serveur.i2tch.loc:80
# ServerName directive.
#    ServerName dummy-host.example.com
...
```

Configurez le système pour que les modules **ip_vs**, **ip_vs_rr** et **ip_vs_wrr** soient chargés au démarrage :

```
[root@centos6 ~]# echo modprobe ip_vs >> /etc/rc.modules
[root@centos6 ~]# echo modprobe ip_vs_rr >> /etc/rc.modules
[root@centos6 ~]# echo modprobe ip_vs_wrr >> /etc/rc.modules
[root@centos6 ~]# chmod +x /etc/rc.modules
```

Éditez le fichier **/etc/sysctl.conf** ainsi :

```
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

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

puis rechargez le fichier modifié :

```
[root@centos6 ~]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
error: "net.bridge.bridge-nf-call-ip6tables" is an unknown key
error: "net.bridge.bridge-nf-call-iptables" is an unknown key
error: "net.bridge.bridge-nf-call-arptables" is an unknown key
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 4294967295
kernel.shmall = 268435456
```

Redémarrez la machine virtuelle :

```
[root@centos6 ~]# reboot
```

Vérification de la Configuration

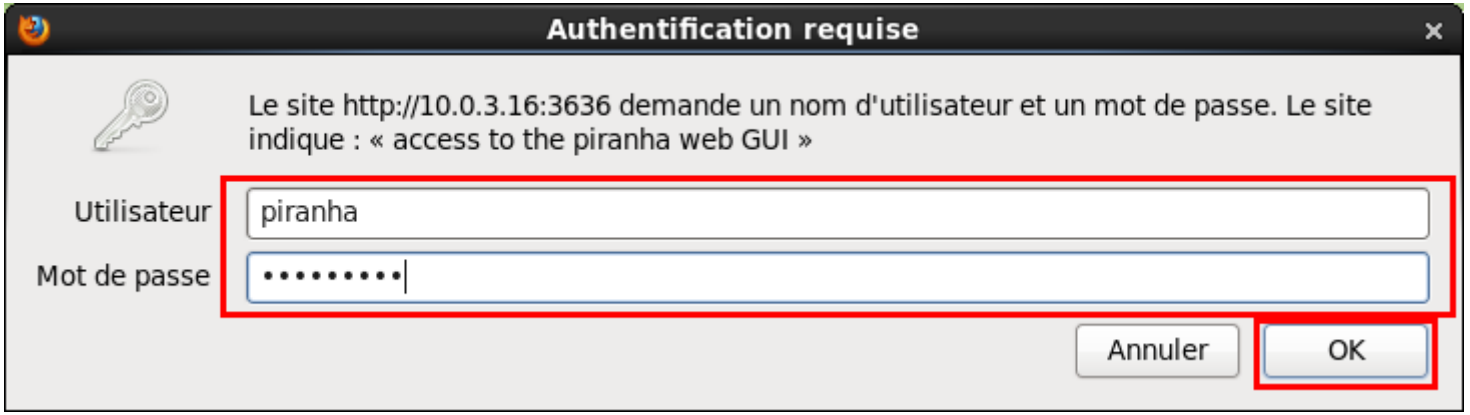
Quand la machine virtuelle a redémarré, vérifiez votre configuration réseau :

```
[root@secours ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:9B:55:B1
          inet adr:10.0.2.51  Bcast:10.0.2.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe9b:55b1/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:29 errors:0 dropped:0 overruns:0 frame:0
          TX packets:206 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:4180 (4.0 KiB)  TX bytes:26964 (26.3 KiB)

eth0:0    Link encap:Ethernet  HWaddr 08:00:27:9B:55:B1
          inet adr:10.0.2.100  Bcast:10.0.2.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

eth1      Link encap:Ethernet  HWaddr 08:00:27:E0:3E:F4
          inet adr:10.0.3.16  Bcast:10.0.3.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fee0:3ef4/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4846 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:0 (0.0 b)  TX bytes:245442 (239.6 KiB)

eth1:0    Link encap:Ethernet  HWaddr 08:00:27:E0:3E:F4
          inet adr:10.0.3.200  Bcast:10.0.3.255  Masque:255.255.255.0
```

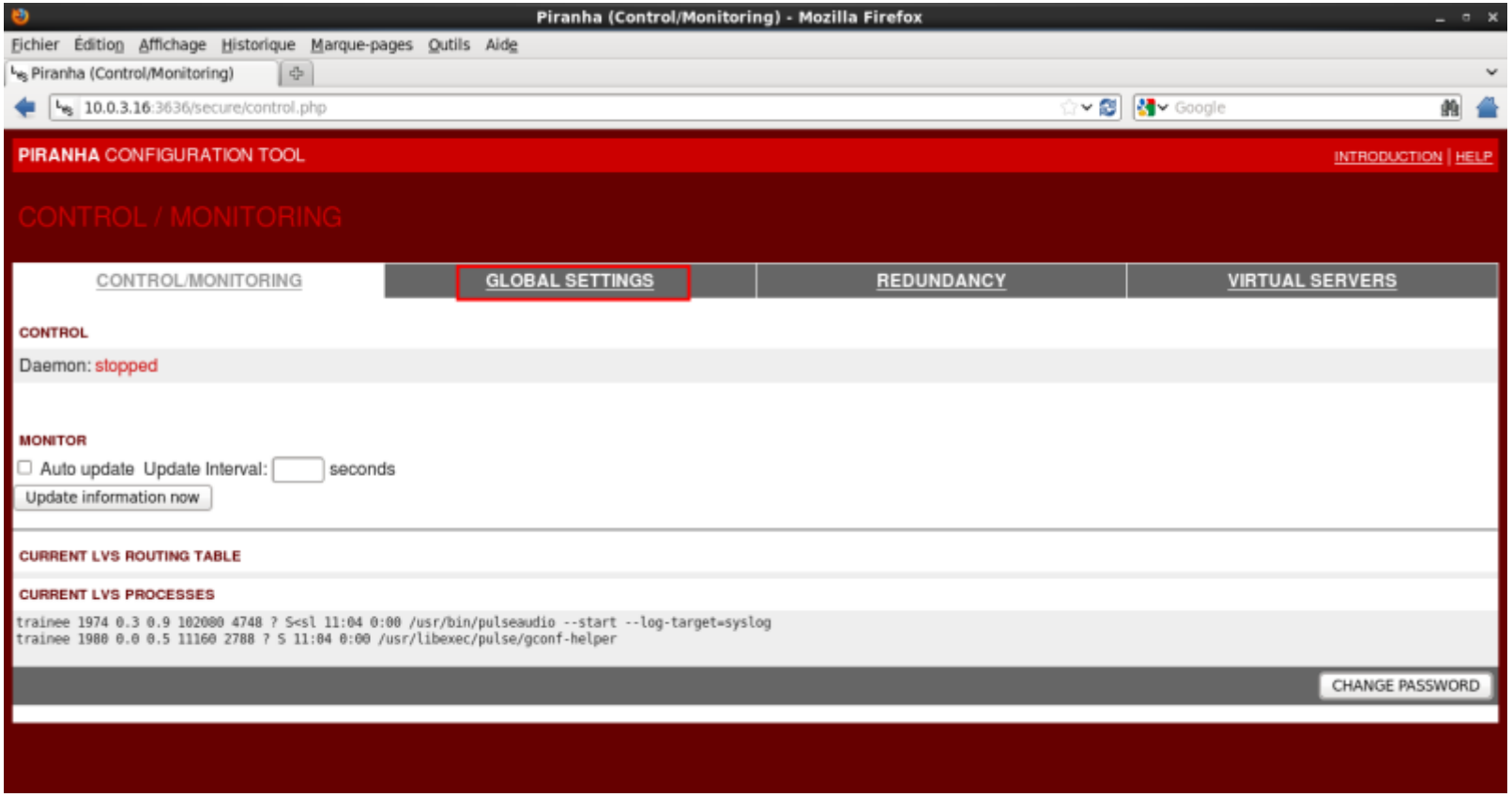
```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:3041 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3041 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:285576 (278.8 KiB)  TX bytes:285576 (278.8 KiB)
```

Configuration du LVS avec Piranha

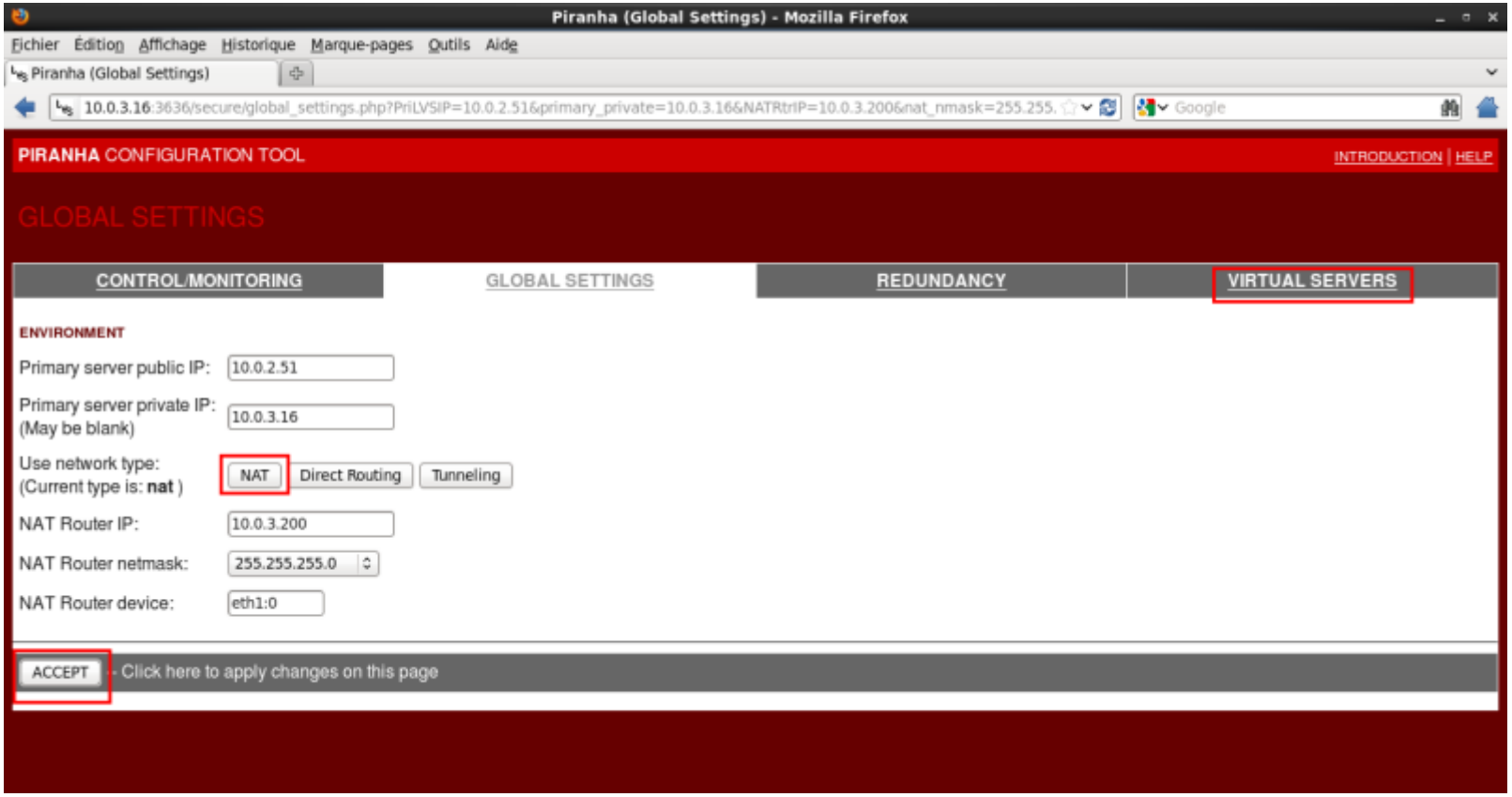
Ouvrez le navigateur web et consultez l'adresse **http://localhost:3636/secure/control.php**. Vous obtiendrez une page similaire à celle-ci :



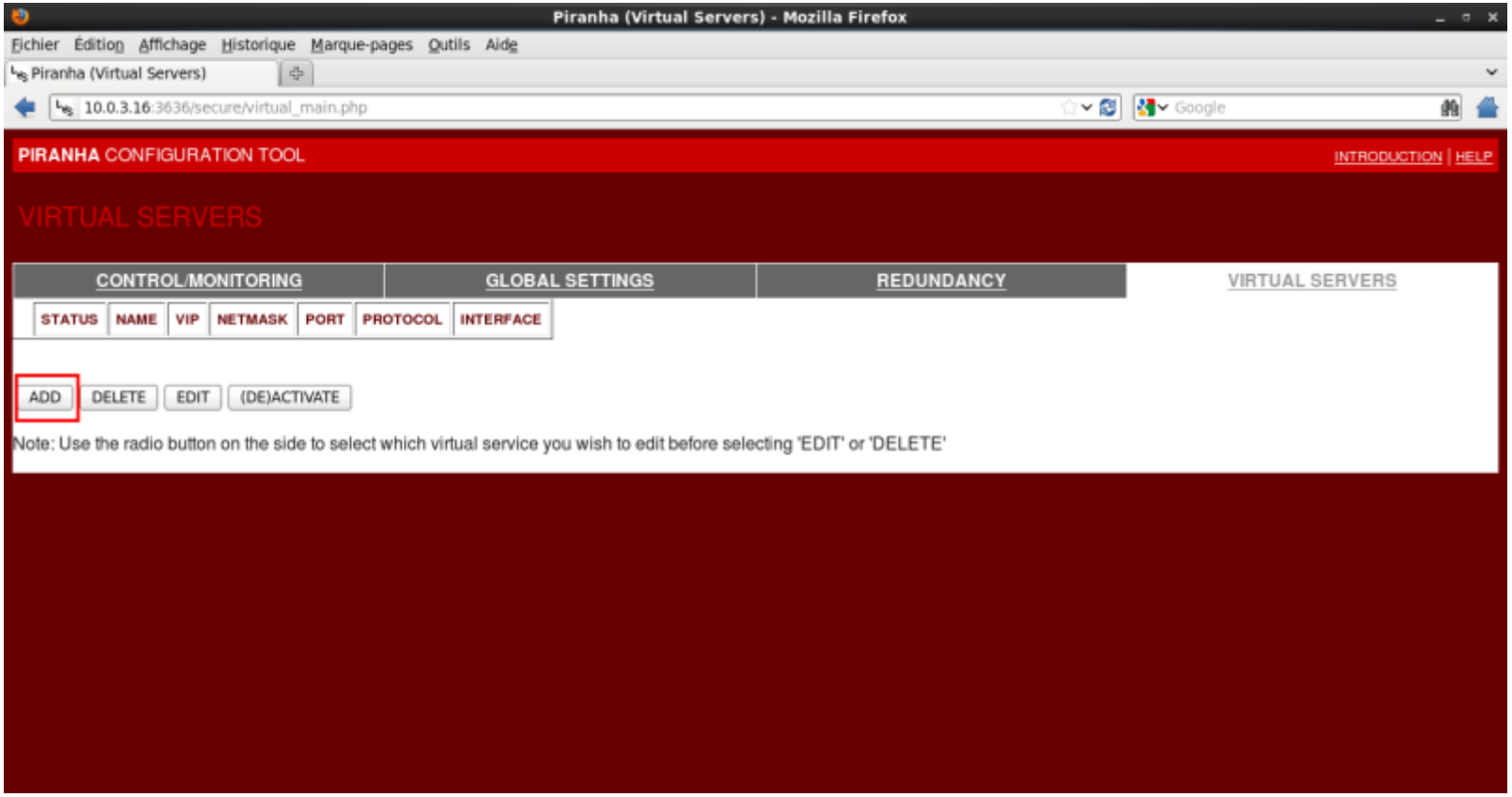
Saisissez le nom d'utilisateur **piranha** ainsi que votre mot de passe et cliquez sur le bouton **OK**. Vous obtiendrez une fenêtre similaire à celle-ci :



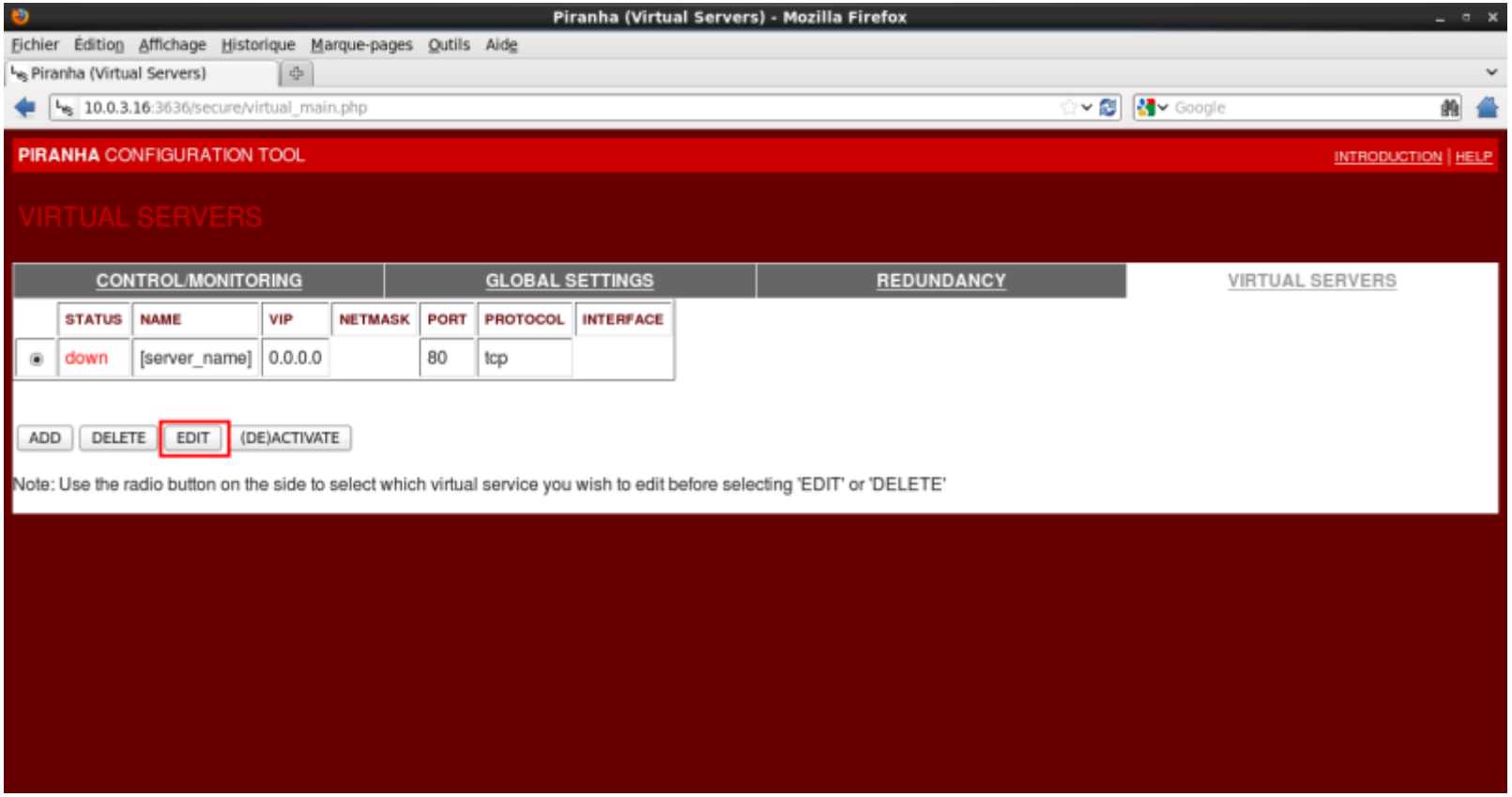
Cliquez sur l'onglet **GLOBAL SETTINGS**, vous obtiendrez une fenêtre similaire à celle-ci :



Assurez-vous que le *Current type is: NAT*, sinon, cliquez sur le bouton **NAT**. Remplissez les champs et cliquez sur le bouton **ACCEPT**. Cliquez ensuite sur l'onglet **VIRTUAL SERVERS**. Vous obtiendrez une fenêtre similaire à celle-ci :

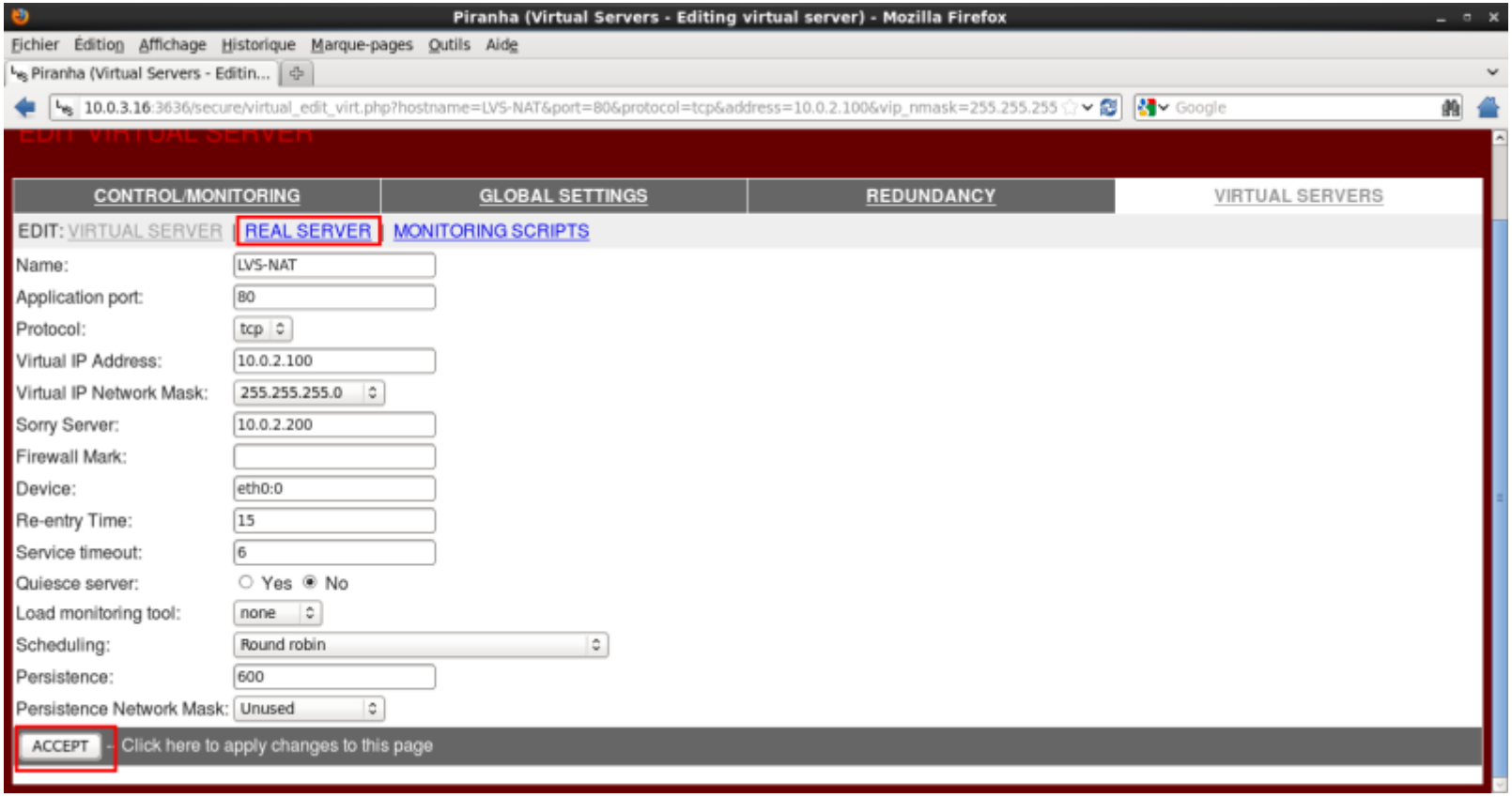


Cliquez sur le bouton **ADD** pour ajouter un serveur virtuel. Vous obtiendrez une fenêtre similaire à celle-ci :

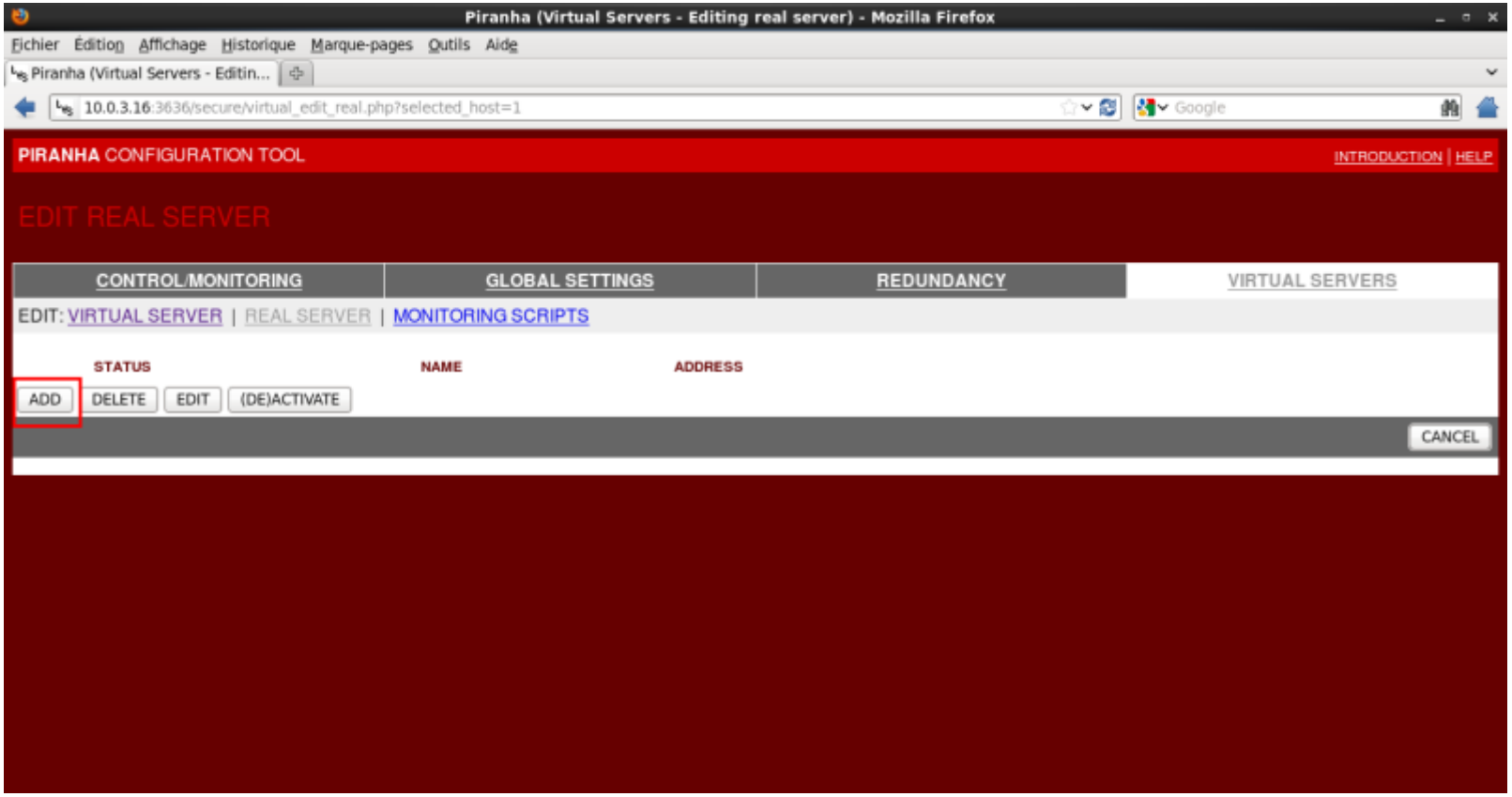


}

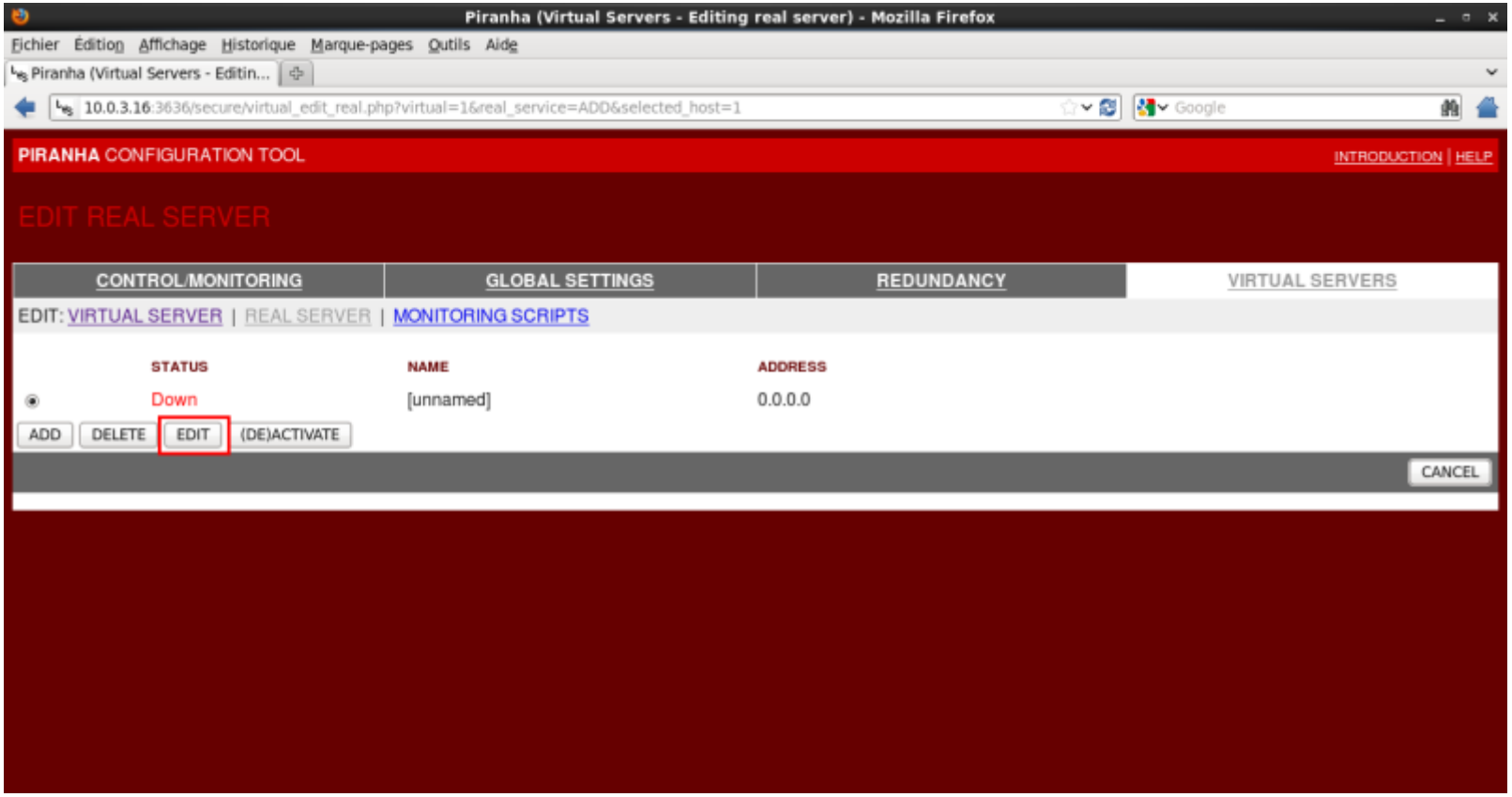
Cliquez sur le bouton **EDIT**. Vous obtiendrez une fenêtre similaire à celle-ci :



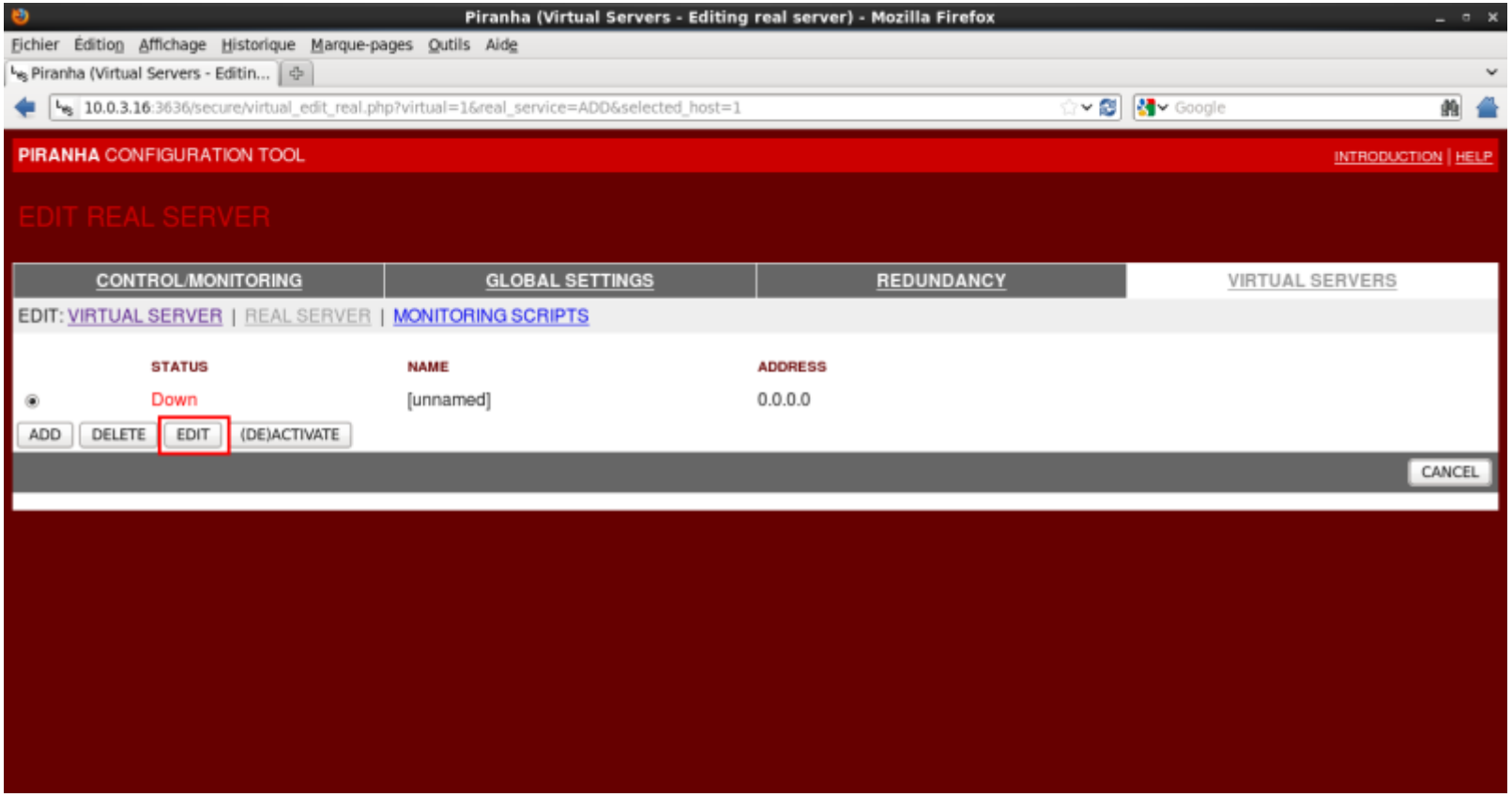
Remplissez le formulaire, cliquez sur le bouton **ACCEPT** puis sur l'onglet **REAL SERVER**. Vous obtiendrez une fenêtre similaire à celle-ci :



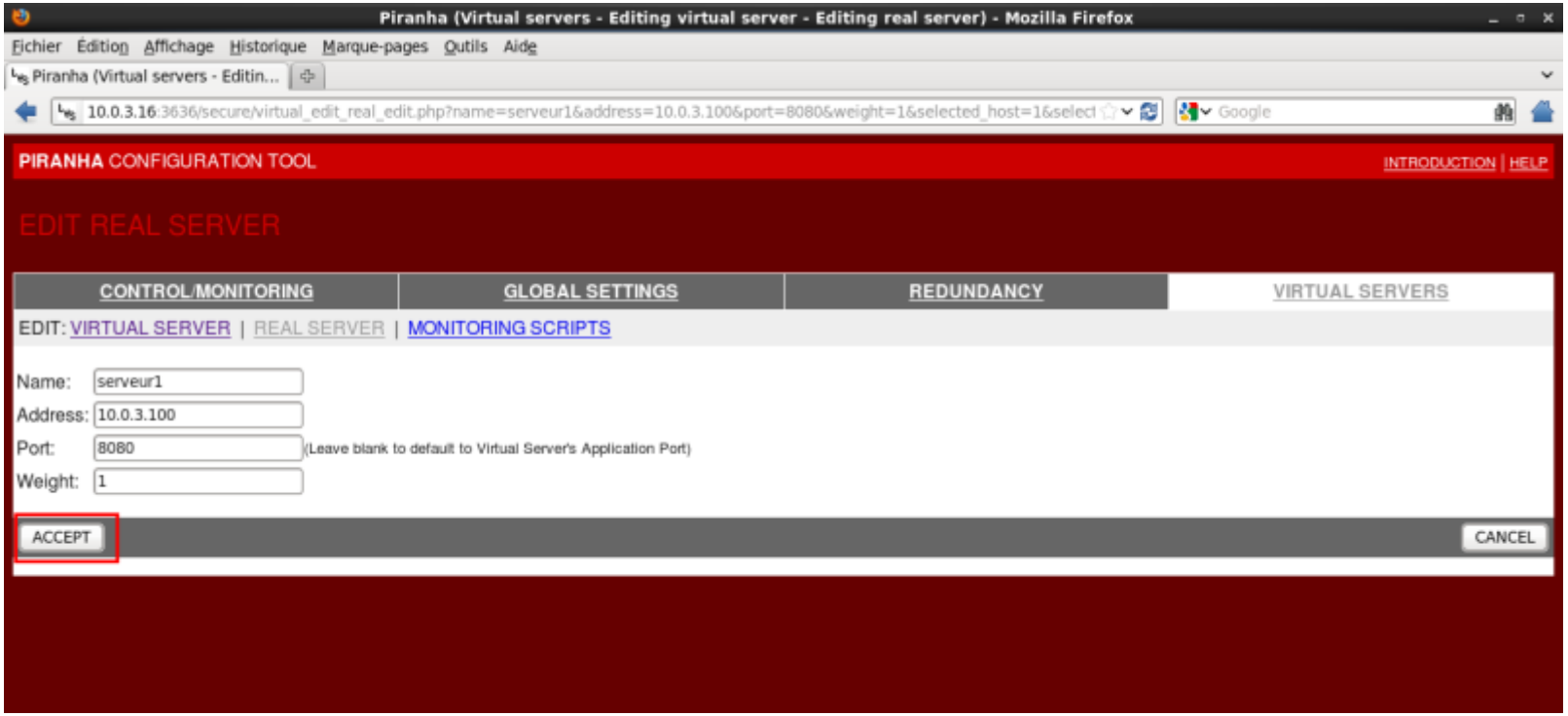
Cliquez sur le bouton **ADD**. Vous obtiendrez une fenêtre similaire à celle-ci :



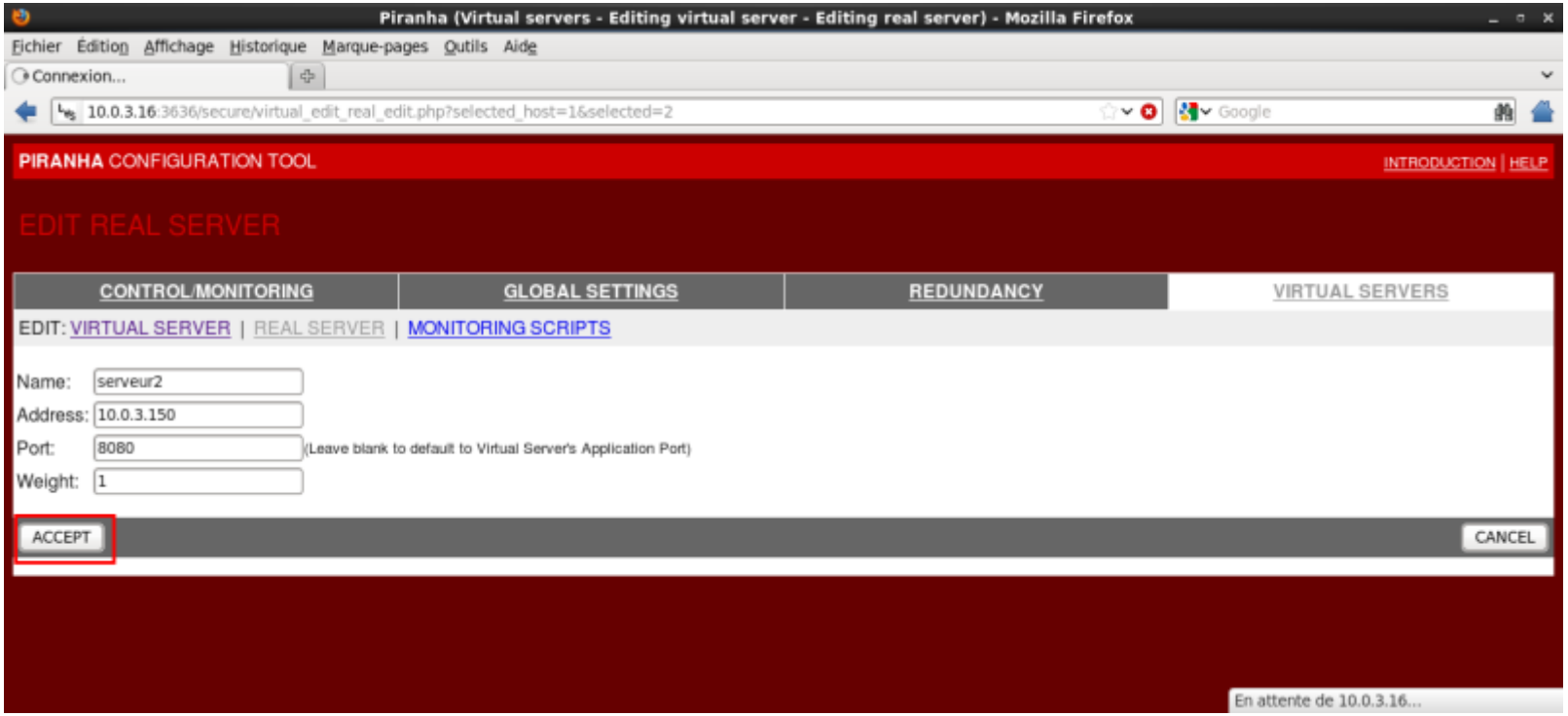
Cliquez sur le bouton **EDIT**. Vous obtiendrez une fenêtre similaire à celle-ci :



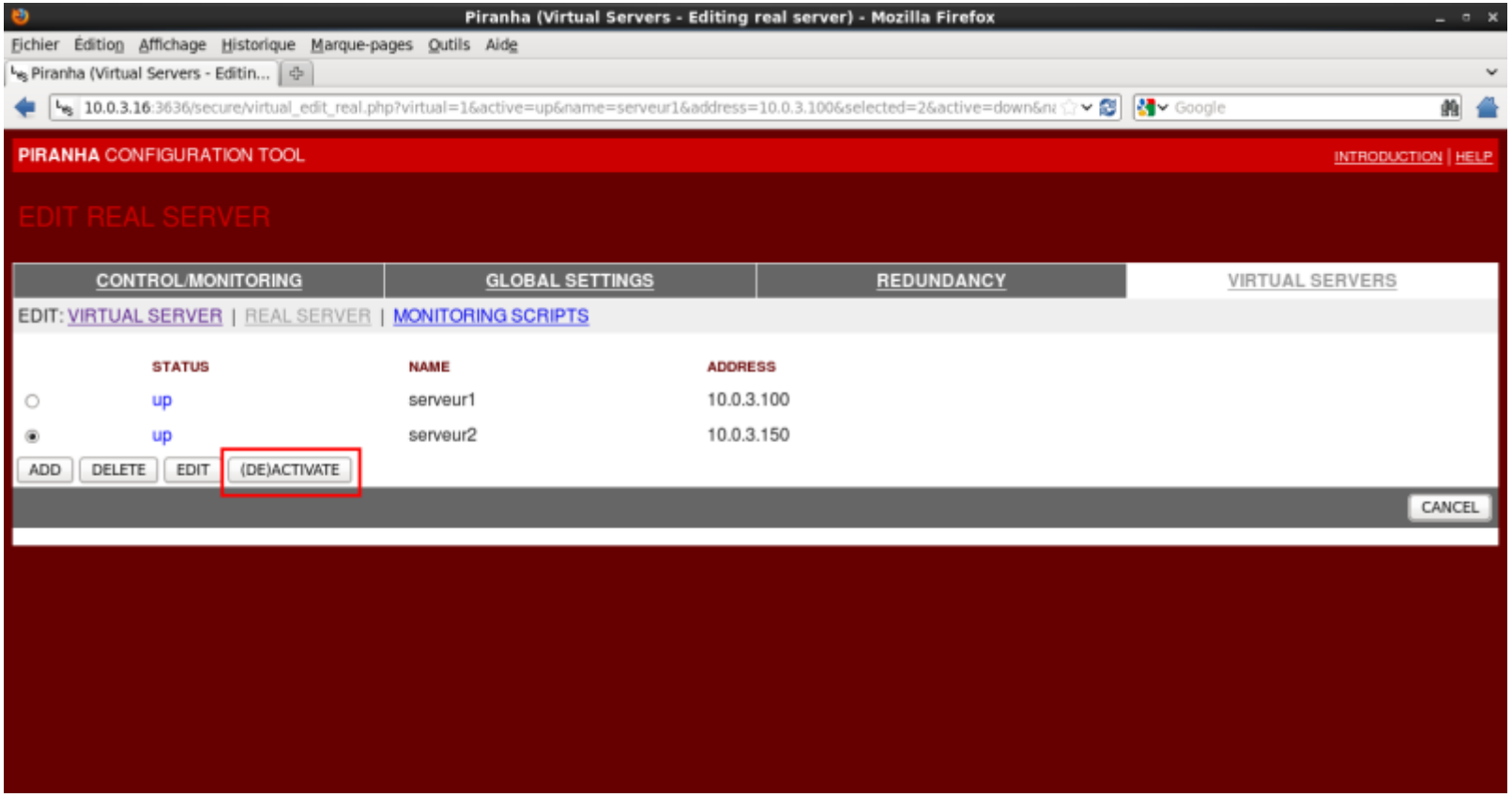
Remplissez les informations pour serveur1 et cliquez sur le bouton **ACCEPT**. Vous obtiendrez une fenêtre similaire à celle-ci :



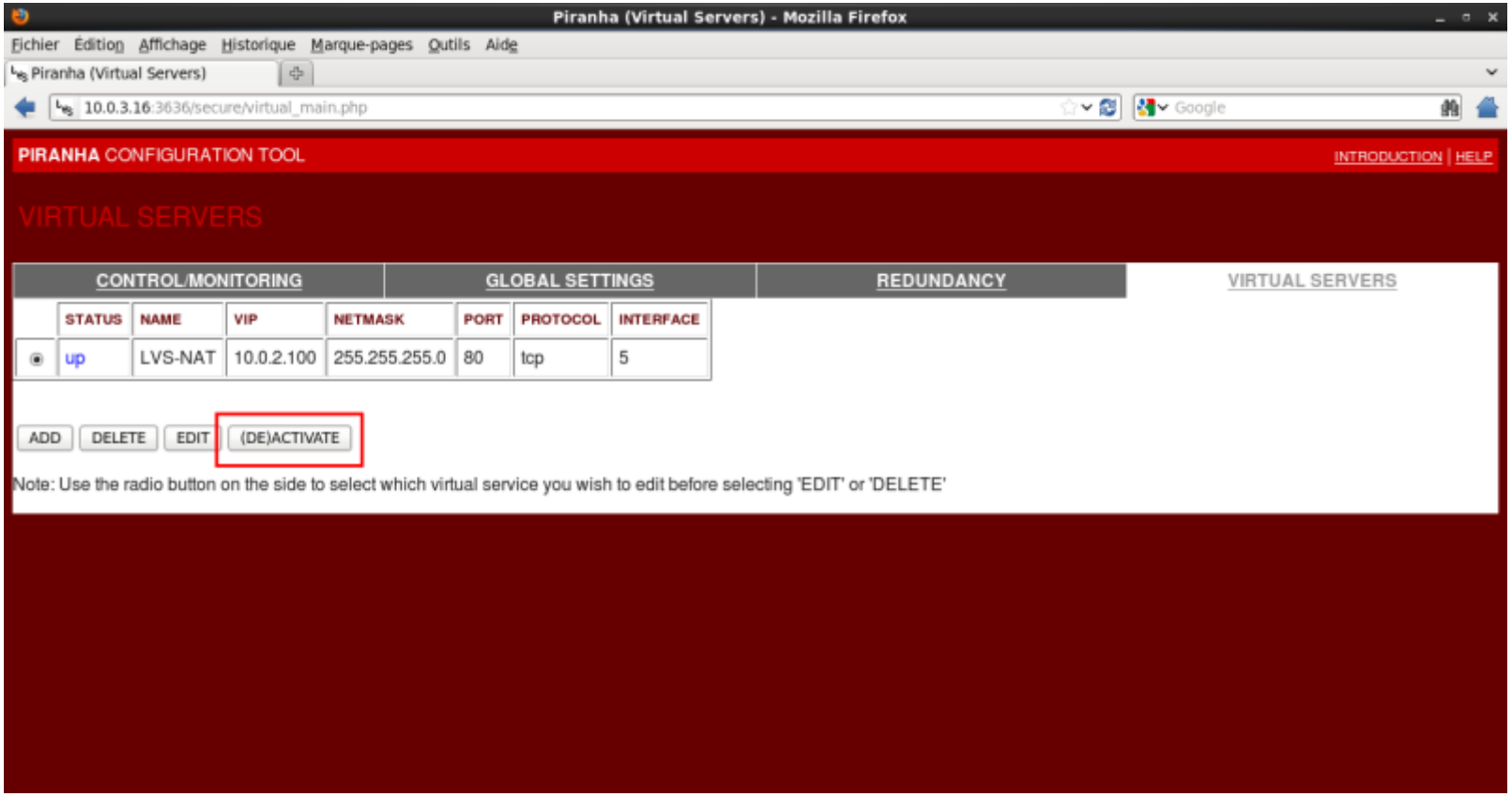
Faites la même chose pour serveur2. Vous obtiendrez une fenêtre similaire à celle-ci :



Activez serveur1 et serveur2 grâce au bouton **(DE)ACTIVATE**. Vous obtiendrez une fenêtre similaire à celle-ci :



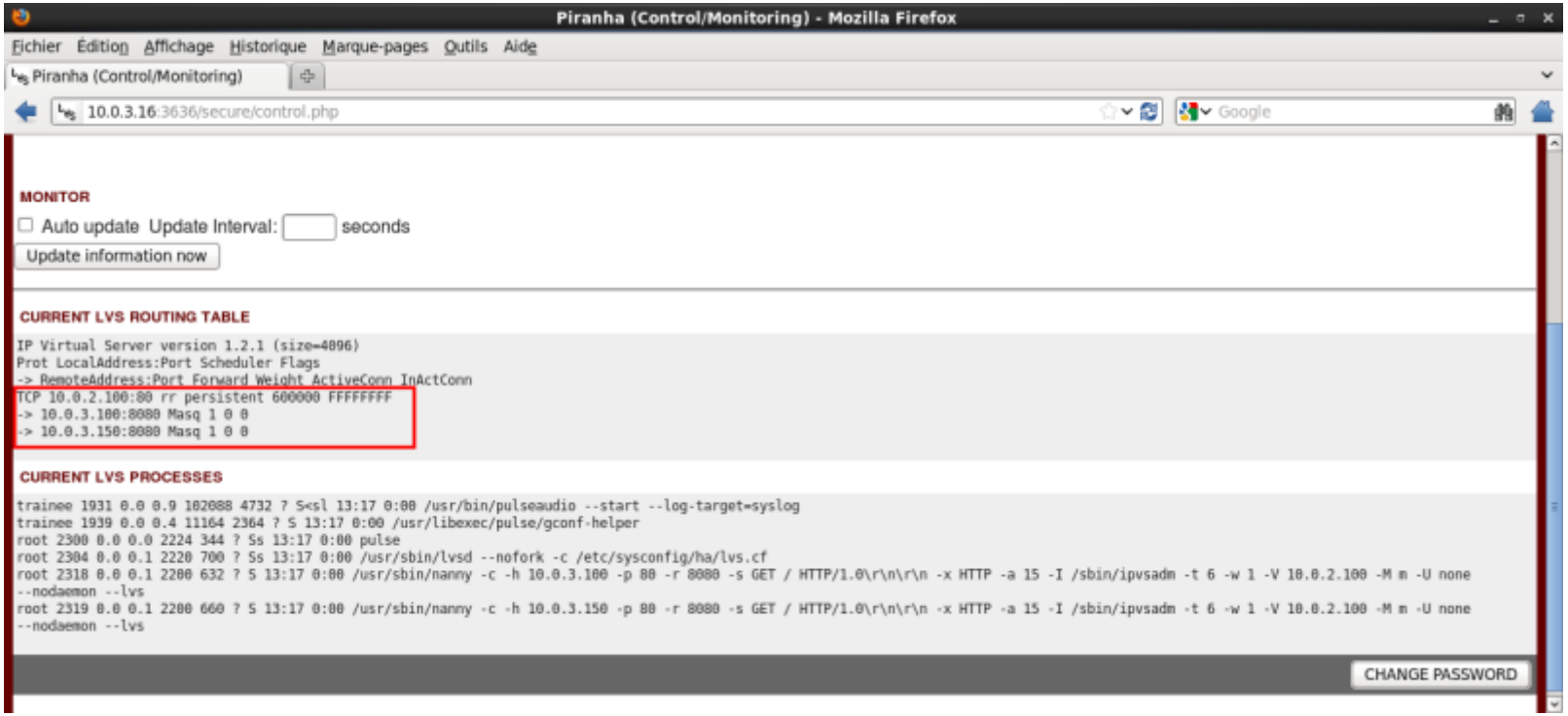
Activez ensuite le serveur virtuel de la même façon. Vous obtiendrez une fenêtre similaire à celle-ci :



Démarrez et configurez le service **pulse** de Piranha :

```
[root@centos6 ~]# chkconfig pulse on
[root@centos6 ~]# chkconfig --list pulse
pulse           0:arrêt    1:arrêt    2:marche    3:marche    4:marche    5:marche    6:arrêt
[root@centos6 ~]# service pulse start
```

Constatez le démarrage du LVS sur **secours** :



Ainsi que les traces dans le journal **/var/log/messages** :

```
...
Dec 20 18:02:52 secours pulse[2782]: STARTING PULSE AS MASTER
Dec 20 18:02:57 secours pulse[2782]: partner dead: activating lvs
Dec 20 18:02:57 secours lvsd[2785]: starting virtual service LVS_NAT active: 80
Dec 20 18:02:57 secours lvsd[2785]: create_monitor for LVS_NAT/serveur1 running as pid 2789
Dec 20 18:02:57 secours lvsd[2785]: create_monitor for LVS_NAT/serveur2 running as pid 2790
Dec 20 18:02:57 secours nanny[2789]: starting LVS client monitor for 10.0.2.100:80 -> 10.0.3.100:8080
Dec 20 18:02:57 secours nanny[2790]: starting LVS client monitor for 10.0.2.100:80 -> 10.0.3.150:8080
Dec 20 18:02:57 secours kernel: send_arp uses obsolete (PF_INET,SOCK_PACKET)
Dec 20 18:02:58 secours nanny[2789]: [ active ] making 10.0.3.100:8080 available
Dec 20 18:02:58 secours nanny[2790]: [ active ] making 10.0.3.150:8080 available
Dec 20 18:03:02 secours pulse[2793]: gratuitous lvs arps finished
```

Testez maintenant la configuration à partir du **client**.

Nanny

L'exécutable **nanny** est ce que l'on appelle un ***healthchecker***. nanny veille à la bonne santé des services cibles en envoyant une chaîne sur le port réel de serveur1 et de serveur2 et en attendant une chaîne en retour.

Dans notre cas, la chaîne envoyée est **GET / HTTP/1.0\r\n\r\n** et la chaîne attendue est **HTTP**.

nanny est aussi capable d'exécuter un programme ou script extérieur.

<html>

</html>
