

M1 ASR Thème 3 du TP(maquette de test de la tolérance de panne avec keepalived + LVS-NAT.)

Etudiant : DIALLO Boubacar

Professeur : Monsieur Pascal PETIT

Soit la maquette :

Une machine Client-LVS : 192.168.10.3 SUR R1 Routeur :ip virtuelle sur R1:192.168.10.100

Directeur 1: 192.168.10.1 sur R1 ET 192.168.2.1 SUR R2 + deux ip virtuelles 192.168.10.100 et 192.168.2.100

Directeur 2: 192.168.10.2 sur R1 ET 192.168.2.2 SUR R2 + deux ip virtuelles 192.168.10.100 et 192.168.2.100

Serveur web1: sur R2: 192.168.2.13 et routeur : ip virtuelle sur R2:192.168.2.100

Serveur web2: sur R2: 192.168.2.14 et routeur : ip virtuelle sur R2:192.168.2.100

Sorry Serveur: sur R2: 192.168.2.15 et routeur : ip virtuelle sur R2:192.168.2.100

CLIENT LVS-NAT

```
# 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
```

```
#allow-hotplug enp0s3
#iface enp0s3 inet dhcp
```

```
auto enp0s3
```

```
# iface enp0s3 inet dhcp
iface enp0s3 inet static
address 192.168.10.3/24
gateway 192.168.10.100
```

Directeur1:

```
# 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
```

```
#allow-hotplug enp0s3
#iface enp0s3 inet dhcp
```

```
auto enp0s3
```

```
# iface enp0s3 inet dhcp
iface enp0s3 inet static
address 192.168.10.1/24
# gateway 192.168.10.2
```

```
auto enp0s8
```

```
iface enp0s8 inet static
address 192.168.2.1/24
```

Activer le forwarding

```
/etc/sysctl.conf :
```

```
net.ipv4.ip_forward = 1
```

activé les modification

```
sudo sysctl -p
```

Pareil pour le Directeur2

Le Serveur Web :

```
# 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
```

```
#allow-hotplug enp0s3
```

```
#iface enp0s3 inet dhcp
```

```
auto enp0s3
```

```
# iface enp0s3 inet dhcp
```

```
iface enp0s3 inet static
```

```
address 192.168.2.13/24
```

```
gateway 192.168.2.100
```

editer le fichier

```
/var/www/html/index.html
```

Test

```
GNU nano 7.2
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Serveur Apache</title>
  <style>
    body {
      margin: 0;
      height: 100vh;
      background-color: green;
      display: flex;
      justify-content: center;
      align-items: center;
      color: white;
      font-family: Arial, sans-serif;
      font-size: 2rem;
    }
  </style>
</head>
<body>
  <div>Je suis le serveur Apache 1</div>
</body>
</html>
```



Pareil pour le serveur 2 et le sorry server



Installation keepalived sur les Directeurs

apt install keepalived et Editer le fichier nano /etc/keepalived/keepalived.conf

Tolerance de Panne

Directeur1 master

```
vrrp_instance VI_1{
    state MASTER
    interface enp0s3
    virtual_router_id 51
    priority 150
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123
    }
    virtual_ipaddress {
        192.168.10.100/25
    }
}
```

```
vrrp_instance VI_2{
    state MASTER
    interface enp0s8
    virtual_router_id 52
    priority 150
    advert_int 1
    authentication{
        auth_type PASS
        auth_pass 123
    }
    virtual_ipaddress{
        192.168.2.100
    }
}
```

pour le Directeur 2 backup

```

vrrp_instance VI_1{
    state BACKUP
    interface enp0s3
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123
    }
    virtual_ipaddress {
        192.168.10.100/24
    }
}

```

```

vrrp_instance VI_2{
    state BACKUP
    interface enp0s8
    virtual_router_id 52
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123
    }
    virtual_ipaddress {
        192.168.2.100/24
    }
}

```

Dans wireshark

The screenshot shows a Wireshark capture of network traffic. The packet list pane displays the following entries:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.2.1	224.0.0.18	VRRP	62	Announcement (v2)
2	0.000047855	192.168.10.1	224.0.0.18	VRRP	62	Announcement (v2)
3	1.044260583	192.168.2.1	224.0.0.18	VRRP	62	Announcement (v2)
4	1.044409322	192.168.10.1	224.0.0.18	VRRP	62	Announcement (v2)
5	1.419973201	192.168.2.2	192.168.2.14	TCP	76	57658 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=901044610 TSecr=0 WS=128
6	1.420853897	192.168.2.14	192.168.2.2	TCP	76	80 → 57658 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2007607242 TSecr=901044610 WS=128
7	1.420947044	192.168.2.2	192.168.2.14	TCP	68	57658 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=901044611 TSecr=2007607242
8	1.421103434	192.168.2.2	192.168.2.14	TCP	68	57658 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=901044611 TSecr=2007607242
9	1.422617769	192.168.2.14	192.168.2.2	TCP	68	80 → 57658 [FIN, ACK] Seq=1 Ack=2 Win=65280 Len=0 TSval=2007607244 TSecr=901044611
10	1.422674539	192.168.2.2	192.168.2.14	TCP	68	57658 → 80 [ACK] Seq=2 Ack=2 Win=64256 Len=0 TSval=901044613 TSecr=2007607244
11	1.575566682	192.168.2.2	192.168.2.13	TCP	76	46844 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=90068590 TSecr=0 WS=128
12	1.576587084	192.168.2.13	192.168.2.2	TCP	76	80 → 46844 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3448648191 TSecr=90068590 WS=128
13	1.576658575	192.168.2.2	192.168.2.13	TCP	68	46844 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=90068592 TSecr=3448648191
14	1.576799370	192.168.2.2	192.168.2.13	TCP	68	46844 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=90068592 TSecr=3448648191
15	1.578328255	192.168.2.13	192.168.2.2	TCP	68	80 → 46844 [FIN, ACK] Seq=1 Ack=2 Win=65280 Len=0 TSval=3448648192 TSecr=90068592
16	1.578370907	192.168.2.2	192.168.2.13	TCP	68	46844 → 80 [ACK] Seq=2 Ack=2 Win=64256 Len=0 TSval=90068593 TSecr=3448648192

nous pouvons constaté les annonce vrrp du directeur qui signal qu'il est vivant

Redemarer le service keepalived

sudo systemctl enable keepalived

sudo systemctl restart keepalived

les deux serveurs actifs le master recupere l'ip virtuelle

Master

```
root@debian:~# ip a | grep 192
    inet 192.168.10.1/24 brd 192.168.10.255 scope global enp0s3
    inet 192.168.10.100/25 scope global enp0s3
    inet 192.168.2.1/24 brd 192.168.2.255 scope global enp0s8
    inet 192.168.2.100/32 scope global enp0s8
```

BACKUP

```
root@debian:~# ip a | grep 192
    inet 192.168.10.2/24 brd 192.168.10.255 scope global enp0s3
    inet 192.168.2.2/24 brd 192.168.2.255 scope global enp0s8
root@debian:~# █
```

Après avoir etteint le serveur 1

le backup recuper l'ip virtuelle


```
root@debian:~# ip a | grep 192
  inet 192.168.10.2/24 brd 192.168.10.255 scope global enp0s3
  inet 192.168.2.2/24 brd 192.168.2.255 scope global enp0s8
root@debian:~# ip a | grep 192
  inet 192.168.10.2/24 brd 192.168.10.255 scope global enp0s3
  inet 192.168.10.100/24 scope global secondary enp0s3
  inet 192.168.2.2/24 brd 192.168.2.255 scope global enp0s8
  inet 192.168.2.100/24 scope global secondary enp0s8
root@debian:~# █
```

Repartition de charge (Toujour dans le fichier keepalived.conf)

```
#configuration LVS-NAT
virtual_server 192.168.10.100 80 {

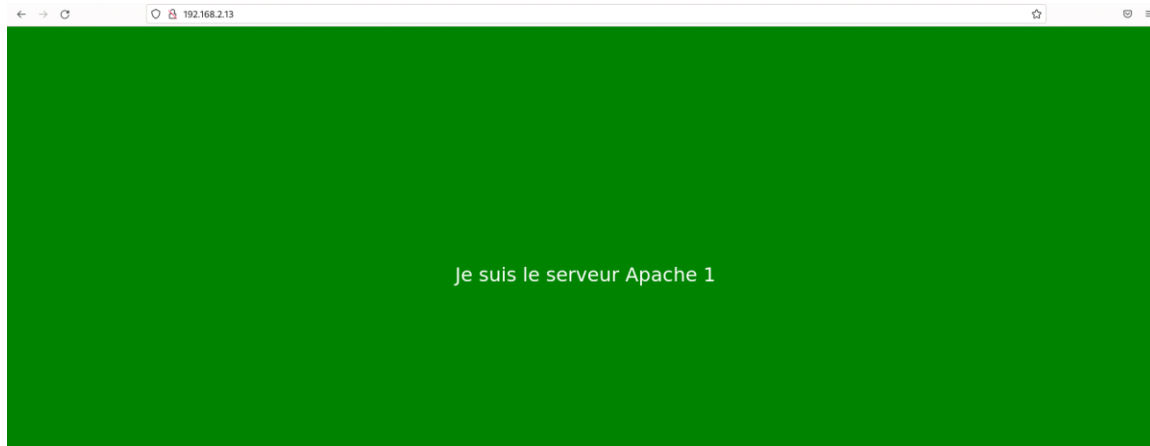
    delay_loop 2
    lb_algo rr
    lb_kind NAT
    persistence_timeout 0
    protocol TCP

    real_server 192.168.2.13 80 {
        weight 1
        TCP_CHECK {
            connect_timeout 1
            nb_get_retry 1
            delay_before_retry 1
        }
    }

    real_server 192.168.2.14 80 {
        weight 1
        TCP_CHECK {
            connect_timeout 1
            nb_get_retry 1
            delay_before_retry 1
        }
    }

    sorry_server 192.168.2.15 80
}
█
```

à partir du client nous pouvons accéder directement au serveur web passant par le directeur en utilisant l'ip du serveur



Maintenant utilisant l'ip virtuelle le directeur sélectionnera le serveur actif dans sa table que nous pouvons vérifier

on peut voir la table en live avec la commande :

`sudo watch ipvsadm -L -n --stats`

```
root@debian:~# ipvsadm -L -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  192.168.10.100:80 rr
  -> 192.168.2.13:80              Masq    1      0         1
  -> 192.168.2.14:80              Masq    1      0         0
root@debian:~# █
```

Nous avons la même table dans le directeur2

Donc le Directeur va sélectionner un des serveurs disponibles dans la table à travers l'algorithme round robin

En mettant un serveur 192.168.2.13 hors service nous affichons la table du directeur pour voir son état

```

root@debian:~# ipvsadm -L -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  192.168.10.100:80 rr
    -> 192.168.2.14:80            Masq    1      0      0
root@debian:~# █

```

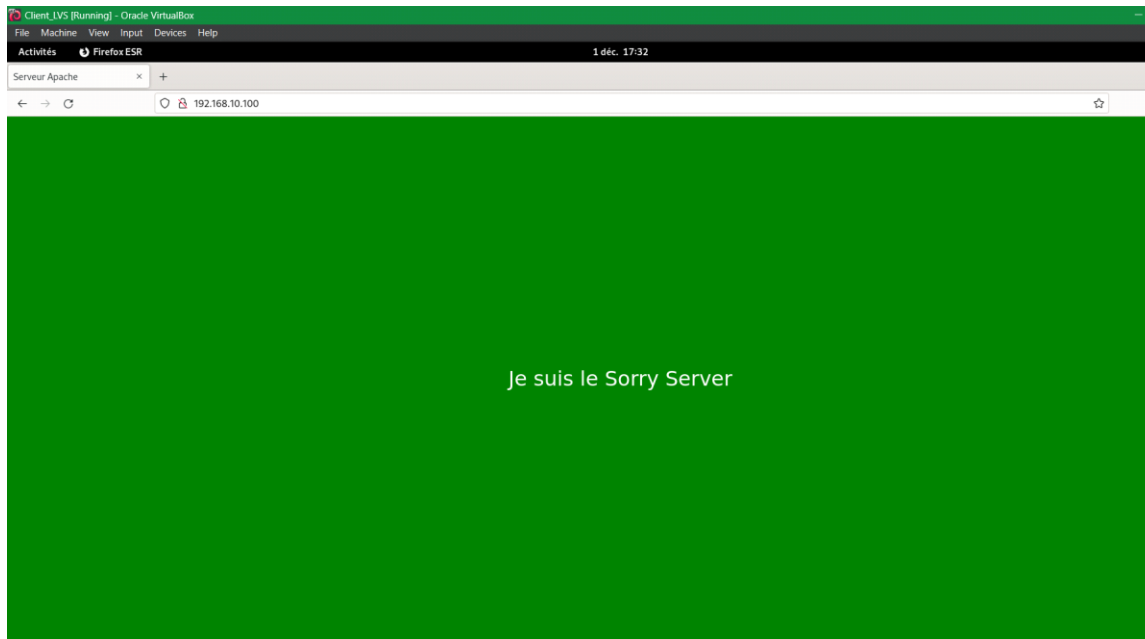
Arretons le serveur 2 et revoyons la table nous verrons apparaitre le sorry serveur

```

root@debian:~# nano /etc/keepalived/keepalived.conf
root@debian:~# ipvsadm -L -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  192.168.10.100:80 rr
    -> 192.168.2.13:80            Masq    1      0      1
    -> 192.168.2.14:80            Masq    1      0      0
root@debian:~# ipvsadm -L -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  192.168.10.100:80 rr
    -> 192.168.2.14:80            Masq    1      0      0
root@debian:~# ipvsadm -L -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  192.168.10.100:80 rr
    -> 192.168.2.15:80            Masq    1      0      0
root@debian:~# █

```

Le client demande une requête avec l'ip virtuel le directeur check sa table il ne voit que le sorry serveur il affiche



1. Notion de Healthchecker

Le healthchecker est un mécanisme utilisé pour surveiller l'état des serveurs dans un pool backend. Sa fonction principale est de vérifier régulièrement si les serveurs répondent correctement et, en cas d'échec, de retirer temporairement le serveur défaillant du pool.

Principe :

Le healthchecker envoie des requêtes périodiques (ping ICMP, requêtes HTTP) aux serveurs backend.

Si un serveur ne répond pas ou répond avec une erreur, il est marqué comme indisponible.

Une fois le serveur rétabli, il est réintégré automatiquement dans le pool.

5	1.419973201	192.168.2.2	192.168.2.14	TCP	76 57658 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=901044610 TSecr=0 WS=128
6	1.420853897	192.168.2.14	192.168.2.2	TCP	76 80 → 57658 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2807607242 TSecr=901044610 WS=128
7	1.420947044	192.168.2.2	192.168.2.14	TCP	68 57658 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=901044611 TSecr=2807607242
8	1.421103434	192.168.2.2	192.168.2.14	TCP	68 57658 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=901044611 TSecr=2807607242
9	1.422617789	192.168.2.14	192.168.2.2	TCP	68 80 → 57658 [FIN, ACK] Seq=1 Ack=2 Win=65280 Len=0 TSval=2807607244 TSecr=901044611
10	1.422674539	192.168.2.2	192.168.2.14	TCP	68 57658 → 80 [ACK] Seq=2 Ack=2 Win=64256 Len=0 TSval=901044613 TSecr=2807607244
11	1.575566682	192.168.2.2	192.168.2.13	TCP	76 46844 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=90068590 TSecr=0 WS=128
12	1.576587004	192.168.2.13	192.168.2.2	TCP	76 80 → 46844 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3448648191 TSecr=90068590 WS=128
13	1.576585875	192.168.2.2	192.168.2.13	TCP	68 46844 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=90068592 TSecr=3448648191
14	1.576799370	192.168.2.2	192.168.2.13	TCP	68 46844 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=90068592 TSecr=3448648191
15	1.578322825	192.168.2.13	192.168.2.2	TCP	68 80 → 46844 [FIN, ACK] Seq=1 Ack=2 Win=65280 Len=0 TSval=3448648192 TSecr=90068592
16	1.578392997	192.168.2.2	192.168.2.13	TCP	68 46844 → 80 [ACK] Seq=2 Ack=2 Win=64256 Len=0 TSval=90068593 TSecr=3448648192

l'etat du directeur deux après avoir stop le directeur 1 et le serveur web 1

IL EST MASTER AVEC LE HEALTHCHECKER IL MET A JOUR L'etat de sa table

```

root@debian:~# systemctl status keepalived
● keepalived.service - Keepalived Daemon (LVS and VRRP)
   Loaded: loaded (/lib/systemd/system/keepalived.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-12-08 11:48:40 CET; 32min ago
     Docs: man:keepalived(8)
           man:keepalived.conf(5)
           man:genhash(1)
           https://keepalived.org
   Main PID: 665 (keepalived)
     Tasks: 3 (limit: 2285)
    Memory: 6.0M
       CPU: 956ms
    CGroup: /system.slice/keepalived.service
            └─665 /usr/sbin/keepalived --dont-fork
              └─691 /usr/sbin/keepalived --dont-fork
                └─692 /usr/sbin/keepalived --dont-fork

déc. 08 12:02:13 debian Keepalived_healthcheckers[691]: Removing service [192.168.2.13]:tcp:80 from VS [192.168.10.100]:tcp:80
déc. 08 12:02:29 debian Keepalived_healthcheckers[691]: TCP_CHECK on service [192.168.2.14]:tcp:80 failed after 1 retries.
déc. 08 12:02:29 debian Keepalived_healthcheckers[691]: Removing service [192.168.2.14]:tcp:80 from VS [192.168.10.100]:tcp:80
déc. 08 12:02:29 debian Keepalived_healthcheckers[691]: Lost quorum 1-0=1 > 0 for VS [192.168.10.100]:tcp:80
déc. 08 12:02:29 debian Keepalived_healthcheckers[691]: Adding sorry server [192.168.2.15]:tcp:80 to VS [192.168.10.100]:tcp:80
déc. 08 12:02:29 debian Keepalived_healthcheckers[691]: Removing alive servers from the pool for VS [192.168.10.100]:tcp:80
déc. 08 12:21:30 debian Keepalived_vrrp[692]: (VI_2) Master received advert from 192.168.2.1 with higher priority 150, ours 100
déc. 08 12:21:30 debian Keepalived_vrrp[692]: (VI_2) Entering BACKUP STATE
déc. 08 12:21:30 debian Keepalived_vrrp[692]: (VI_1) Master received advert from 192.168.10.1 with higher priority 150, ours 100
déc. 08 12:21:30 debian Keepalived_vrrp[692]: (VI_1) Entering BACKUP STATE
root@debian:~# █

```

Quorum et Sorry Server :

Le quorum définit le nombre minimal de serveurs nécessaires Si le nombre de serveurs tombe sous le quorum :

Le sorry server est activé Il sert de backup de dernier recours Il permet d'assurer un service minimal

Synchronisation des données dynamiques :

Les données à synchroniser incluent : La table de connexions actives

L'état des serveurs Les statistiques La synchronisation se fait via le protocole VRRP

Traitement de la requête Client

a. Le client envoie une requête HTTP à la VIP

Le client entre l'adresse VIP dans son navigateur, par exemple : <http://192.168.10.100/>. Une requête TCP est initiée depuis le client (IP source : 192.168.10.3) vers la VIP (IP destination : 192.168.10.100).

Le paquet TCP passe par le routeur ou le commutateur jusqu'à atteindre le premier directeur LVS actif (Directeur1 ou Directeur2).

b. Le directeur LVS traite la requête

Réception de la requête par le directeur actif :Le paquet arrive à l'adresse VIP sur le directeur actif. Keepalived gère la VIP en collaboration avec ipvsadm pour la répartition de charge.

Décision de répartition de charge : Le directeur LVS examine sa table LVS (Virtual Server Table), configurée par ipvsadm. Il sélectionne un serveur réel (Web1, Web2, ou Sorry Server) en fonction de l'algorithme de répartition RR.

Etudes détaillée du flux des paquets

Le client 192.168.10.3 avec un port random dans notre le port 49292 envoi un paquet de type TCP sync vers la destination 192.168.10.100 sur le port 80 (paquet numero 31 dans captures client avec tous les cas)

LVS intercepte ce paquet et le transmet au Directeur1 qui est le maitre Source: 192.168.10.3:49292
Destination: 192.168.10.1:80 Type: TCP SYN (avec marquage IPVS)

le directeur fait une traduction d'adresse pour envoyé la requête au serveur web au choix dans sa pool de serveur real disponible

Source: 192.168.10.3:49292

Destination: 192.168.2.13:80

Type: TCP SYN (avec translation d'adresse)

le serveur web repond au serveur qui fait une traduction d'adresse pour le retour

Source: 192.168.2.100:80

Destination: 192.168.10.3:sur le port 49292

Type: TCP SYN-ACK

Entre les deux directeurs

le directeur 1 directeur envoi un paquet vrrp par diffusion multicast avec la destination 224.0.0.18

Resumé des protocols

TCP/IP pour le trafic applicatif

VRRP pour la haute disponibilité des Directeurs

IPVS pour la distribution de charge

ARP pour la gestion des adresses virtuelles

HTTP/HTTPS pour le trafic web

ICMP/TCP pour les health checks

No.	Time	Source	Destination	Protocol	Length	Info
195	45.218595215	192.168.10.3	10.0.2.3	DNS	101	Standard query 0x446c A contile.services.mozilla.com.univ-evry.fr
196	45.218567306	192.168.10.3	10.0.2.3	DNS	101	Standard query 0x8c6e AAAA contile.services.mozilla.com.univ-evry.fr
197	45.671026327	192.168.10.3	10.0.2.3	DNS	98	Standard query 0x8ca7 A push.services.mozilla.com.univ-evry.fr
198	45.671059311	192.168.10.3	10.0.2.3	DNS	98	Standard query 0x8cab AAAA push.services.mozilla.com.univ-evry.fr
199	45.744463418	192.168.10.3	192.168.10.100	HTTP	516	GET / HTTP/1.1
200	45.746179258	192.168.10.100	192.168.10.3	HTTP	731	HTTP/1.1 200 OK (text/html)
201	45.746219085	192.168.10.3	192.168.10.100	TCP	66	53312 - 80 [ACK] Seq=901 Ack=1332 Win=64128 Len=0 TSval=683851205 TSecr=1230189901
202	46.166417368	192.168.10.2	224.0.0.18	VRRP	60	Announcement (v2)
203	46.324609774	192.168.10.3	192.168.10.100	HTTP	516	GET / HTTP/1.1
204	46.326197026	192.168.10.100	192.168.10.3	HTTP	731	HTTP/1.1 200 OK (text/html)
205	46.326237238	192.168.10.3	192.168.10.100	TCP	66	53312 - 80 [ACK] Seq=1351 Ack=1997 Win=64128 Len=0 TSval=683851785 TSecr=1230190480
206	46.713702567	192.168.10.3	10.0.2.3	DNS	95	Standard query 0x9256 A content-signature-2.cdn.mozilla.net
207	46.713773688	192.168.10.3	10.0.2.3	DNS	95	Standard query 0x836b AAAA content-signature-2.cdn.mozilla.net
208	47.171422510	192.168.10.2	224.0.0.18	VRRP	60	Announcement (v2)
209	47.353957070	192.168.10.3	192.168.10.100	HTTP	516	GET / HTTP/1.1
210	47.356948032	192.168.10.100	192.168.10.3	HTTP	731	HTTP/1.1 200 OK (text/html)
211	47.356971972	192.168.10.3	192.168.10.100	TCP	66	53312 - 80 [ACK] Seq=1801 Ack=2662 Win=64128 Len=0 TSval=683852816 TSecr=1230191511
212	47.583481368	192.168.10.3	192.168.10.100	HTTP	516	GET / HTTP/1.1
213	47.583515178	192.168.10.100	192.168.10.3	HTTP	731	HTTP/1.1 200 OK (text/html)
214	47.585195923	192.168.10.3	192.168.10.100	TCP	66	53312 - 80 [ACK] Seq=2251 Ack=3327 Win=64128 Len=0 TSval=683853044 TSecr=1230191740
215	47.637810966	192.168.10.3	10.0.2.3	DNS	90	Standard query 0x88ed A metadata.ftp-master.debian.org
216	47.637836418	192.168.10.3	10.0.2.3	DNS	90	Standard query 0xaceaa AAAA metadata.ftp-master.debian.org
217	48.181719256	192.168.10.2	224.0.0.18	VRRP	60	Announcement (v2)
218	49.182461467	192.168.10.2	224.0.0.18	VRRP	60	Announcement (v2)
219	50.052487971	192.168.10.3	192.168.10.100	HTTP	516	GET / HTTP/1.1
220	50.054124744	192.168.10.100	192.168.10.3	HTTP	731	HTTP/1.1 200 OK (text/html)
221	50.054171509	192.168.10.3	192.168.10.100	TCP	66	53312 - 80 [ACK] Seq=2701 Ack=3992 Win=64128 Len=0 TSval=683855513 TSecr=1230194209
222	50.084813560	192.168.10.3	10.0.2.3	DNS	84	Standard query 0x4d31 A detectportal.firefox.com
223	50.084839678	192.168.10.3	10.0.2.3	DNS	84	Standard query 0x2a3f AAAA detectportal.firefox.com

FIN