

DOC TPE HAPROXY ET HEARTBEAT

-Définitions importantes :

HProxy : HAProxy est un logiciel libre de répartition de charge écrit par Willy Tarreau. HAProxy peut aider à mettre en place des solutions de hautes disponibilités, de répartition de charge et de proxy pour tous les types de protocole.

Heartbeat : HeartBeat est un logiciel de surveillance de la disponibilité des programmes, pour les SE, Linux, FreeBSD, OpenBSD, Solaris et MacOS X. Il est distribué sous licence GPL. HeartBeat écoute les battements de coeur -des signaux émis par les services d'une grappe de serveurs lorsqu'ils sont opérationnels. Il exécute des scripts d'initialisations lorsqu'une machine tombe (plus d'entente du battement de coeur) ou est à nouveau disponible. Il permet d'échanger une adresse IP entre les machines à l'aide de mécanisme ARP. HeartBeat fonctionne à partir de deux machines et peut être mis en place pour des architectures réseaux plus complexes.

Apache2 : Le logiciel libre Apache HTTP Server (Apache) est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.

Chmod : chmod (abréviation de change mode) est la commande correspondante qui permet de changer les permissions d'accès d'un fichier ou d'un répertoire.

GPL :

ARP : ARP (protocole de résolution d'adresse) est un protocole utilisé pour traduire une adresse de protocole de couche réseau.

-Exercice :

Deux machine Debian Stretch où on a installé **apache2** , php et **heartbeat** sont serveurs Web :

```

root@srvweb1:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:99:67:02 brd ff:ff:ff:ff:ff:ff
    inet 172.20.0.21/24 brd 172.20.0.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe99:6702/64 scope link
        valid_lft forever preferred_lft forever
root@srvweb1:~# _

```

Ici , srvweb1 en LAN Segment sur VMWare et static avec comme adresse ip 172.20.0.21/24 .

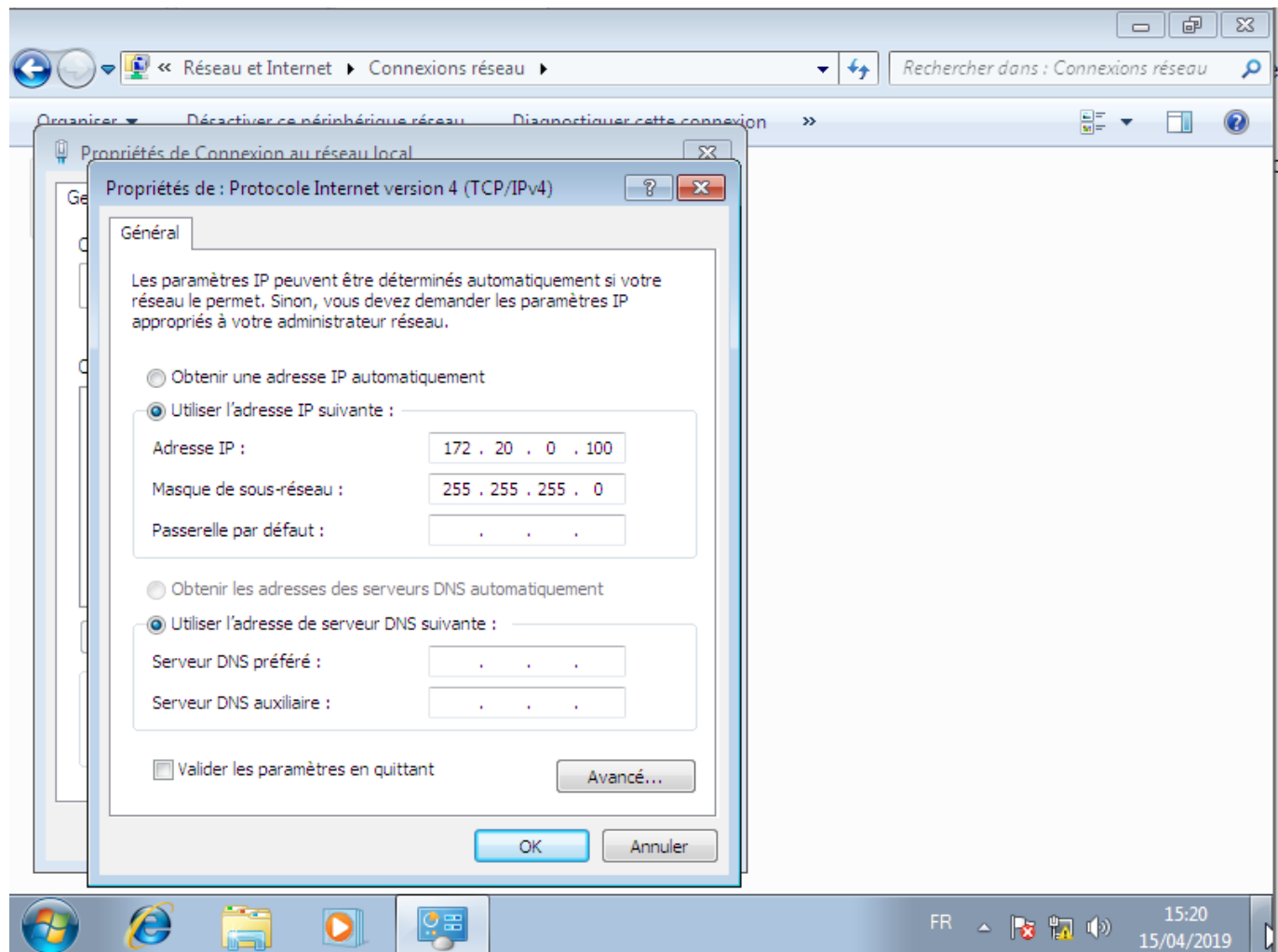
```

root@srvweb2:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:78:9a:ff brd ff:ff:ff:ff:ff:ff
    inet 172.20.0.22/24 brd 172.20.0.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe78:9aff/64 scope link
        valid_lft forever preferred_lft forever
root@srvweb2:~# _

```

Puis srvweb2 en LAN Segment sur VMWare et static avec comme adresse ip 172.20.0.22/24.

Nous avons aussi besoin d'une machine Windows en segment LAN d'adresse IP statique :
172.20.0.100/24



VALIDATION

Depuis srvweb1 :

```

root@srvweb1:~# service apache2 status
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-04-15 14:52:27 CEST; 35min ago
   Process: 333 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 456 (apache2)
     Tasks: 6 (limit: 19660)
    CGroup: /system.slice/apache2.service
            └─456 /usr/sbin/apache2 -k start
              └─480 /usr/sbin/apache2 -k start
                └─481 /usr/sbin/apache2 -k start
                  └─482 /usr/sbin/apache2 -k start
                    └─483 /usr/sbin/apache2 -k start
                      └─484 /usr/sbin/apache2 -k start

avril 15 14:52:27 srvweb1 systemd[1]: Starting The Apache HTTP Server...
avril 15 14:52:27 srvweb1 apachectl[333]: AH00558: apache2: Could not reliably determine the server's
fully qualified domain name, using 172.20.0.21. Set the 'ServerName' directive globally to suppress
s this message
avril 15 14:52:27 srvweb1 systemd[1]: Started The Apache HTTP Server.
root@srvweb1:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:99:67:02 brd ff:ff:ff:ff:ff:ff
    inet 172.20.0.21/24 brd 172.20.0.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe99:6702/64 scope link
        valid_lft forever preferred_lft forever
root@srvweb1:~# _

```

Nous voyons donc que le service apache2 est activé sur srvweb1 et que son adresse ip est bien 172.20.0.21/24

Depuis srvweb2 :

```

root@srvweb2:~# service apache2 status
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2019-04-15 14:52:31 CEST; 40min ago
  Process: 427 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 466 (apache2)
  Tasks: 6 (limit: 19660)
  CGroup: /system.slice/apache2.service
          └─466 /usr/sbin/apache2 -k start
            └─501 /usr/sbin/apache2 -k start
              └─502 /usr/sbin/apache2 -k start
                └─503 /usr/sbin/apache2 -k start
                  └─504 /usr/sbin/apache2 -k start
                    └─505 /usr/sbin/apache2 -k start

avril 15 14:52:31 srvweb2 systemd[1]: Starting The Apache HTTP Server...
avril 15 14:52:31 srvweb2 apachectl[427]: AH00558: apache2: Could not reliably determine the server's
fully qualified domain name, using 172.20.0.22. Set the 'ServerName' directive globally to suppress
this message
avril 15 14:52:31 srvweb2 systemd[1]: Started The Apache HTTP Server.
root@srvweb2:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:78:9a:ff brd ff:ff:ff:ff:ff:ff
    inet 172.20.0.22/24 brd 172.20.0.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe78:9aff/64 scope link
        valid_lft forever preferred_lft forever
root@srvweb2:~# _

```

Nous voyons donc que le service apache2 est activé sur srvweb2 et que son adresse ip est bien 172.20.0.22/24

```

root@srvweb2:~# ping 172.20.0.21
PING 172.20.0.21 (172.20.0.21) 56(84) bytes of data.
64 bytes from 172.20.0.21: icmp_seq=1 ttl=64 time=0.356 ms
64 bytes from 172.20.0.21: icmp_seq=2 ttl=64 time=0.264 ms
64 bytes from 172.20.0.21: icmp_seq=3 ttl=64 time=0.309 ms
^C
--- 172.20.0.21 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2040ms
rtt min/avg/max/mdev = 0.264/0.309/0.356/0.042 ms
root@srvweb2:~# _

```

Ici , nous voyons que nous pouvons communiquer avec srvweb1 en pingant celle-ci.

Depuis la machine Windows :

```
C:\Windows\system32\cmd.exe
Envoi d'une requête 'Ping' 172.20.0.21 avec 32 octets de données :
Réponse de 172.20.0.21 : octets=32 temps<1ms TTL=64
Réponse de 172.20.0.21 : octets=32 temps<1ms TTL=64
Réponse de 172.20.0.21 : octets=32 temps<1ms TTL=64

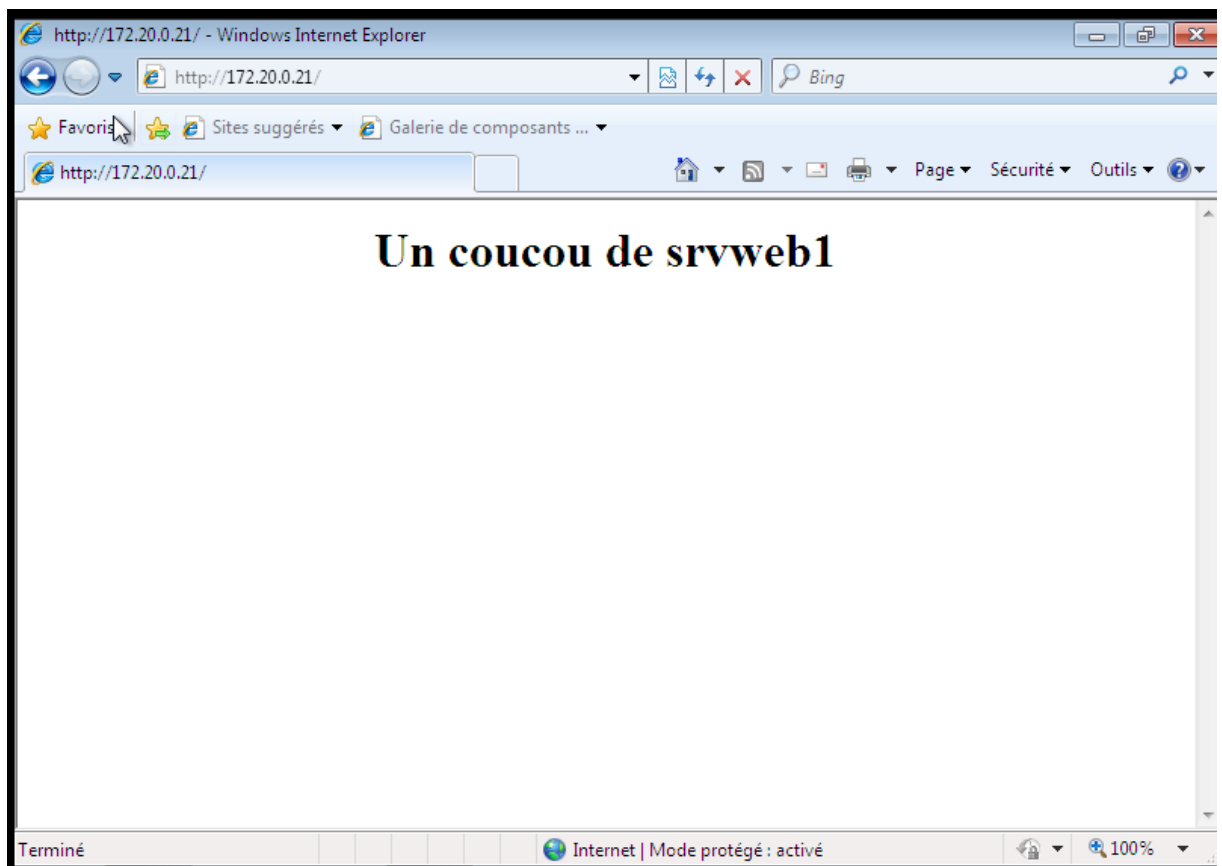
Statistiques Ping pour 172.20.0.21:
    Paquets : envoyés = 3, reçus = 3, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
Ctrl+C
^C
C:\Users\Melanie>ping 172.20.0.22

Envoi d'une requête 'Ping' 172.20.0.22 avec 32 octets de données :
Réponse de 172.20.0.22 : octets=32 temps<1ms TTL=64
Réponse de 172.20.0.22 : octets=32 temps<1ms TTL=64
Réponse de 172.20.0.22 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 172.20.0.22:
    Paquets : envoyés = 3, reçus = 3, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
Ctrl+C
^C
C:\Users\Melanie>
```

Sur la machine Windows nous communiquons bien avec srvweb1 et srvweb2

De plus , grace à un fichier php nommé « index.php » , lorsque nous tapons l'adresse ip de srvweb 1 :



Il en va de même pour srvweb2.

Dans le fichier « hosts » nous allons ajouter les lignes suivantes :

127.0.0.1 localhost

172.20.0.21 srvWeb1

172.20.0.22 srvWeb2

```
127.0.0.1      localhost
127.0.1.1      debian
172.20.0.21    srvweb1
172.20.0.22    srvweb2

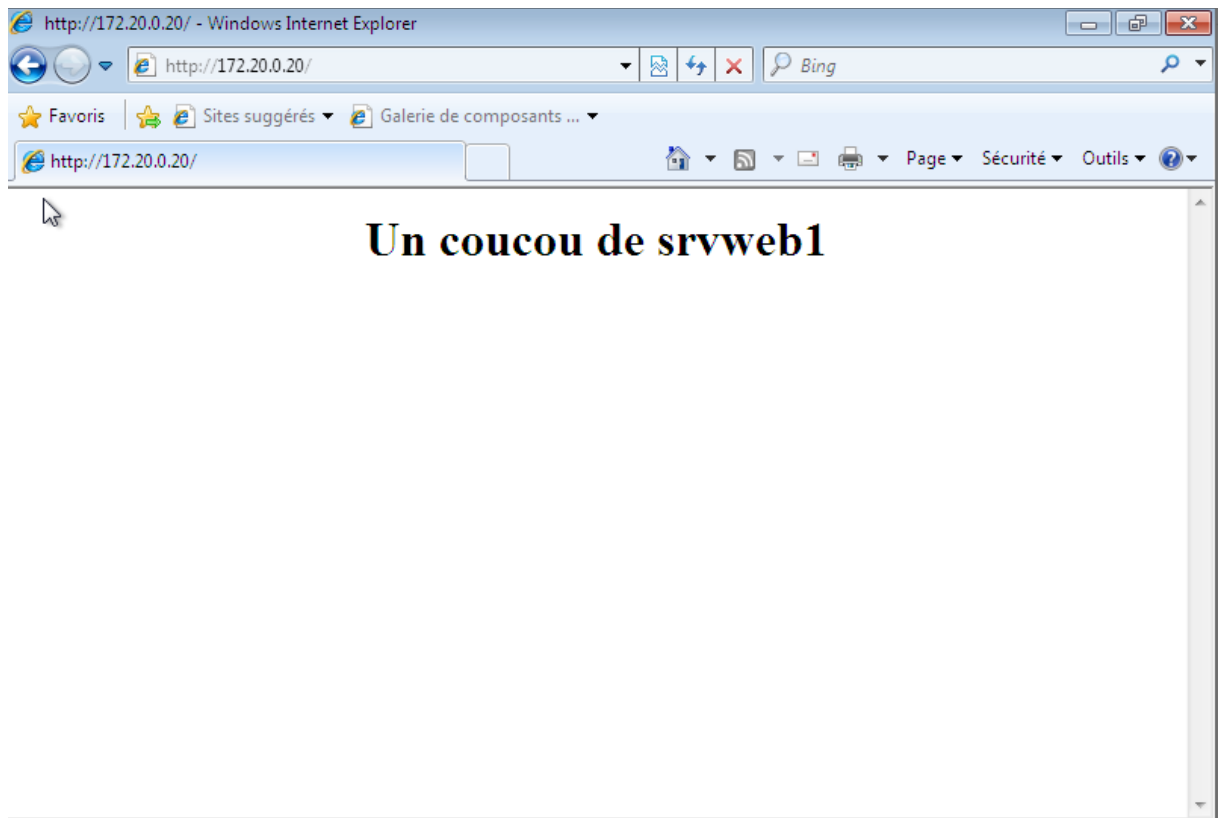
# The following lines are desirable for IPv6 capable hosts
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
```

Puis créer 3 dossiers « **haresources** », activer l'interface IP Virtuelle avec comme noeud principal srvWeb1, « **authkeys** » pour la sécurité de communication des heartbeats et « **ha.cf** » qui permet de configurer le heartbeat .

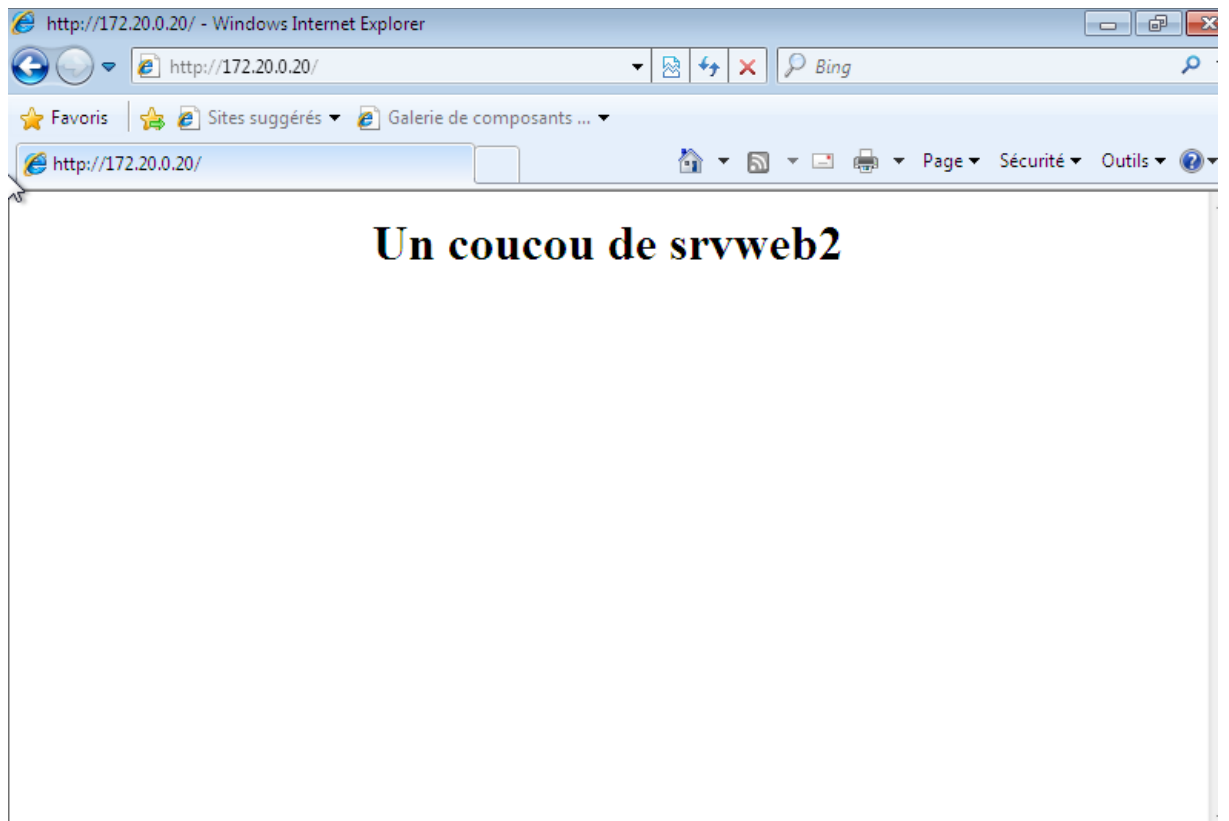
Il faut sécuriser l'accès à ce dernier grâce à la commande **chmod 600 /etc/ha.d/authkeys**

Puis nous desactivons au niveau des deux serveurs Web le démarrage automatique d'apache car c'est heartbeat (haresources) qui va gérer le lancement de ce dernier.

Pour finir, nous désactivons le service apache2 et activons heartbeat ce qui nous donne, lorsque l'on se connecte avec l'adresse ip 172.20.0.20 sur la machine windows :



Et lorsque nous désactivons l'interface réseau de srvweb1 , ça nous montre :



Donc le load balancing est fonctionnel !