

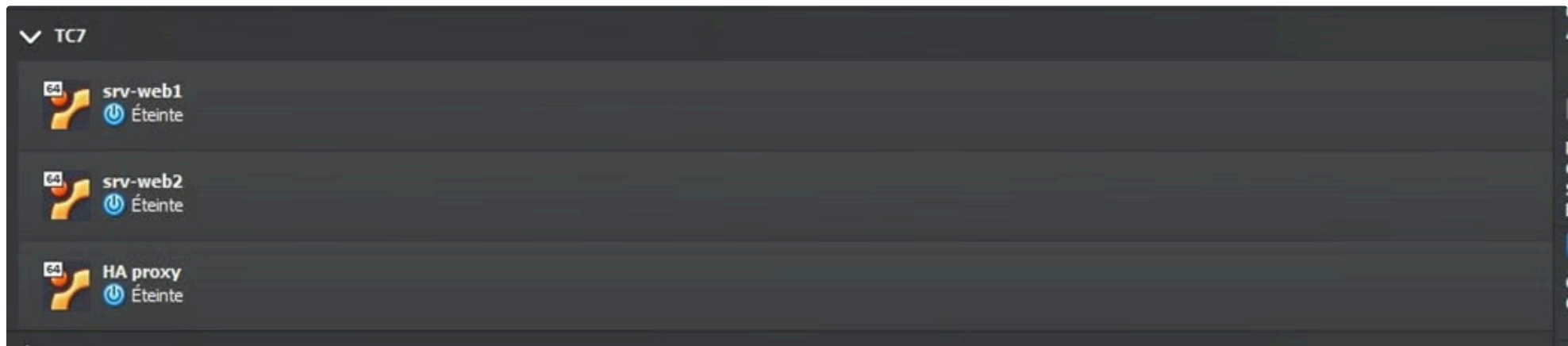
Mise en Place d'une Solution de Haute Disponibilité avec HA PROXY

La haute disponibilité d'un serveur web garantit son accessibilité sans interruption, peu importe le moment. Pour mettre en place un tel système, j'ai utilisé l'outil HA Proxy sous Linux. Cela implique deux serveurs web et une machine virtuelle dédiée à HA Proxy.

Création des Machines Virtuelles

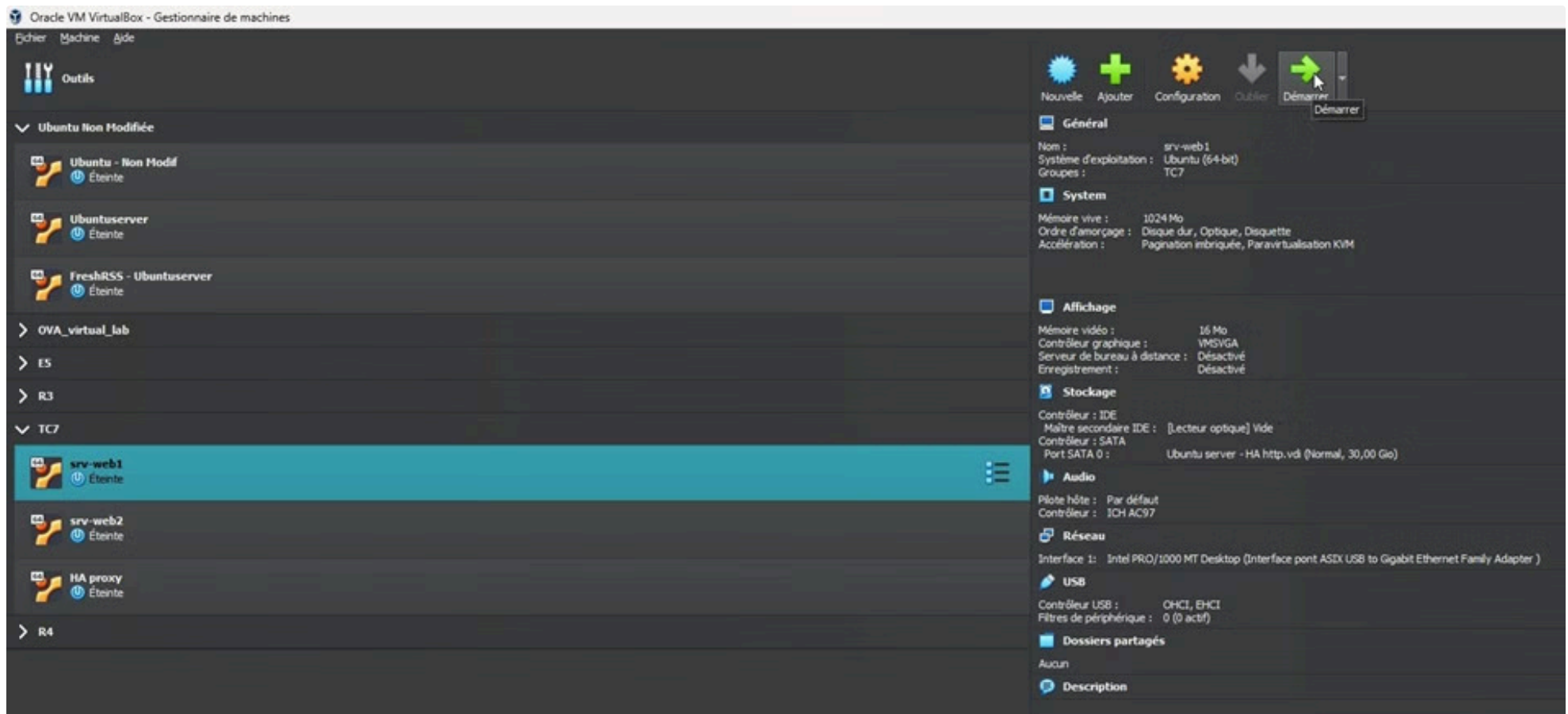
J'ai créé trois machines virtuelles sous VirtualBox :

- Srv-web1 : Serveur web principal.
- Srv-web2 : Serveur web secondaire.
- HA Proxy : Machine virtuelle assurant la haute disponibilité.



Mise en Place des Pages Web

Configuration du Serveur Web



1. Installation des mises à jour des paquets avec `sudo apt-get update`.

```
user@user:~$ sudo apt update && sudo apt upgrade
```

2. Installation du service web Apache2 : `sudo apt-get install apache2`.

```
user@user:~$ sudo apt-get install apache2_
```

3. Vérification du statut d'Apache2 : `sudo systemctl status apache2`.

```
user@srv-web1:/$ sudo systemctl status apache2
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-12-11 19:05:21 UTC; 11min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 11631 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 11635 (apache2)
    Tasks: 55 (limit: 2276)
   Memory: 6.0M (peak: 6.1M)
      CPU: 147ms
  CGroup: /system.slice/apache2.service
          └─11635 /usr/sbin/apache2 -k start
            └─11636 /usr/sbin/apache2 -k start
              └─11665 /usr/sbin/apache2 -k start

Dec 11 19:05:21 srv-web1 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 11 19:05:21 srv-web1 apachectl[11634]: AH00558: apache2: Could not reliably determine the
Dec 11 19:05:21 srv-web1 systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-17/17 (END)
```

Aucunes erreurs.

Création des Pages HTML

J'ai modifié la page index.html via nano /var/www/html/index.html pour différencier les deux serveurs :

```
user@srv-web1:/$ cd /var/www/html/
```

```
user@srv-web1:/var/www/html$ sudo nano index.html
```

Je vais concevoir une page HTML basique avec un contenu statique. Sur le serveur srv- web2, le message "Serveur web numéro 2" sera affiché, permettant ainsi de différencier facilement les deux sites web et de mieux comprendre leur fonctionnement.

Srv-web1 : Contenu « Serveur web numéro 1 ».

```
GNU nano 7.2
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>srv-web1</title>
  </head>
  <body>
    <p>Serveur web numéro 1</p>
  </body>
</html>
```

Srv-web2 : Contenu « Serveur web numéro 2 ».

```
GNU nano 7.2
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>srv-web2</title>
  </head>
  <body>
    <p>Serveur web numéro 2</p>
  </body>
</html>
```

Pour sauvegarder le fichier HTML, utilisez le raccourci clavier CTRL + X, puis appuyez sur Y pour confirmer et terminez en appuyant sur la touche Entrée pour valider

Redémarrage du service Apache2

```
user@srv-web1:/var/www/html$ sudo systemctl restart apache2
```


Vérification du Service Apache2

Vérification du statut du service Apache2

```
user@srv-web1:/var/www/html$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-12-11 19:46:20 UTC; 1min 19s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 11830 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 11835 (apache2)
    Tasks: 55 (limit: 2276)
  Memory: 5.2M (peak: 5.5M)
     CPU: 25ms
   CGroup: /system.slice/apache2.service
           └─11835 /usr/sbin/apache2 -k start
             └─11837 /usr/sbin/apache2 -k start
               └─11838 /usr/sbin/apache2 -k start

Dec 11 19:46:20 srv-web1 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 11 19:46:20 srv-web1 apachectl[11834]: AH00558: apache2: Could not reliably determine the server's f
Dec 11 19:46:20 srv-web1 systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-17/17 (END)
```

Les pages web sont accessibles depuis les adresses IP des serveurs respectifs.

-Vérification des adresses IP des serveur web :

On voit que l'adresse IP de mon srv-web1 est : 192.168.1.88

```
user@srv-web1:/var/www/html$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
    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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
    link/ether 08:00:27:ba:e4:f6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.88/24 metric 100 brd 192.168.1.255 scope
        valid_lft 82747sec preferred_lft 82747sec
    inet6 fe80::a00:27ff:feba:e4f6/64 scope link
        valid_lft forever preferred_lft forever
```

On voit que l'adresse IP de mon srv-web2 est : 192.168.1.6

```
user@srv-web2:/var/www/html$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
    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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
    link/ether 08:00:27:02:7a:65 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.6/24 metric 100 brd 192.168.1.255 scope g
        valid_lft 83287sec preferred_lft 83287sec
    inet6 fe80::a00:27ff:fe02:7a65/64 scope link
        valid_lft forever preferred_lft forever
```

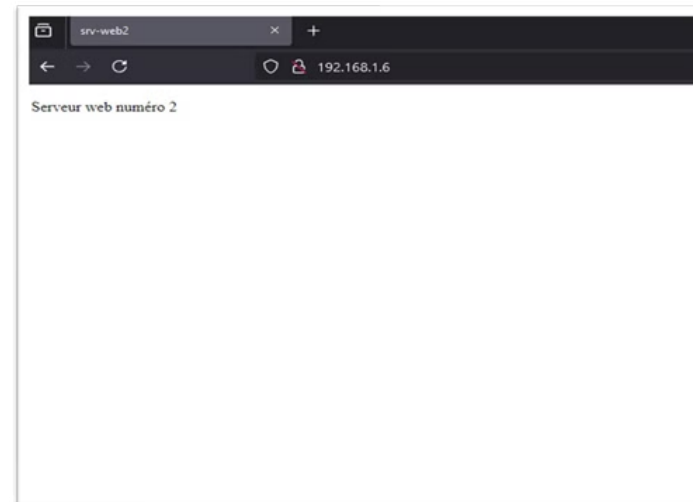
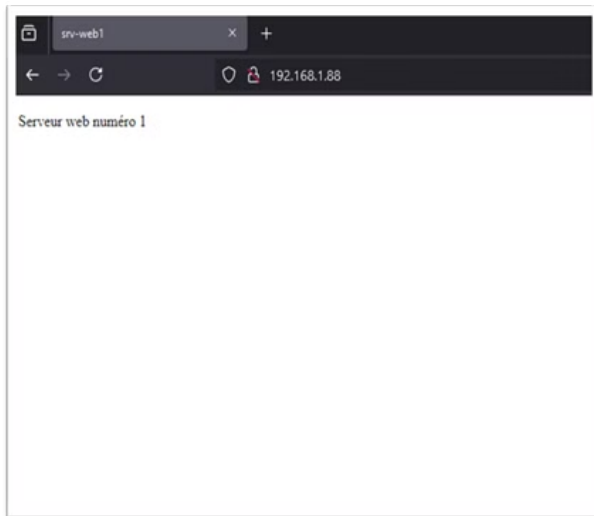
Accès aux Pages Web

Vérification de l'accès aux pages web via un navigateur :

Pour cela, saisissez l'adresse IP de votre serveur web dans la barre de recherche du navigateur.

Dans mon cas, les adresses IP sont les suivantes :

- 192.168.1.88 pour le serveur 1
- 192.168.1.6 pour le serveur 2



Configuration de HA Proxy

Installation et Configuration

1. Installation de HA Proxy : `sudo apt-get install haproxy`.

```
user@HA-proxy:~$ sudo apt-get install haproxy -y
```

2. Modification du fichier de configuration : `/etc/haproxy/haproxy.cfg`.

```
user@HA-proxy:/$ sudo nano etc/haproxy/haproxy.cfg
```

3. Ajout des lignes suivante à la fin du fichier `haproxy.cfg`

```
frontend front_webservers
    bind *:80
    default_backend backend_webservers
    option forwardfor

backend backend_webservers
    balance roundrobin
    server srv-web1 192.168.1.88:80 check
    server srv-web2 192.168.1.6:80 check_
```

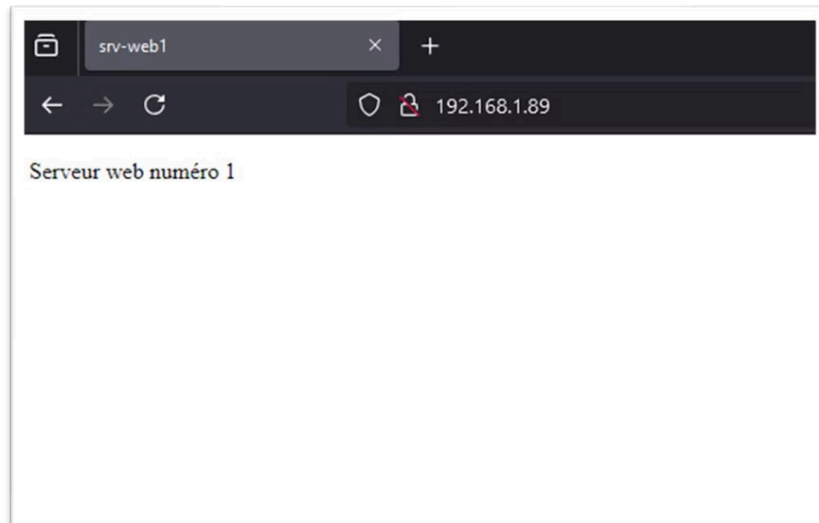
- Frontend : Redirection du trafic HTTP (port 80) vers les serveurs web.
- Backend : Définition de l'algorithme round-robin et des serveurs.
- Redémarrage du service haproxy :

```
user@HA-proxy:/$ sudo systemctl restart haproxy
user@HA-proxy:/$ sudo systemctl status haproxy
• haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-12-11 20:52:49 UTC; 7s ago
     Docs: man:haproxy(1)
           file:/usr/share/doc/haproxy/configuration.txt.gz
   Main PID: 11039 (haproxy)
    Status: "Ready."
     Tasks: 2 (limit: 1068)
  Memory: 39.7M (peak: 39.9M)
     CPU: 104ms
   CGroup: /system.slice/haproxy.service
           └─11039 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/
             11042 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/

Dec 11 20:52:48 HA-proxy haproxy[10896]: [WARNING] (10896) : Exiting Master process...
Dec 11 20:52:48 HA-proxy haproxy[10896]: [ALERT] (10896) : Current worker (10898) exited with co
Dec 11 20:52:48 HA-proxy haproxy[10896]: [WARNING] (10896) : All workers exited. Exiting... (0)
Dec 11 20:52:48 HA-proxy systemd[1]: Stopping haproxy.service - HAProxy Load Balancer...
Dec 11 20:52:48 HA-proxy systemd[1]: haproxy.service: Deactivated successfully.
Dec 11 20:52:48 HA-proxy systemd[1]: Stopped haproxy.service - HAProxy Load Balancer.
Dec 11 20:52:48 HA-proxy systemd[1]: Starting haproxy.service - HAProxy Load Balancer...
Dec 11 20:52:49 HA-proxy haproxy[11039]: [NOTICE] (11039) : New worker (11042) forked
Dec 11 20:52:49 HA-proxy systemd[1]: Started haproxy.service - HAProxy Load Balancer.
Dec 11 20:52:49 HA-proxy haproxy[11039]: [NOTICE] (11039) : Loading success.
```

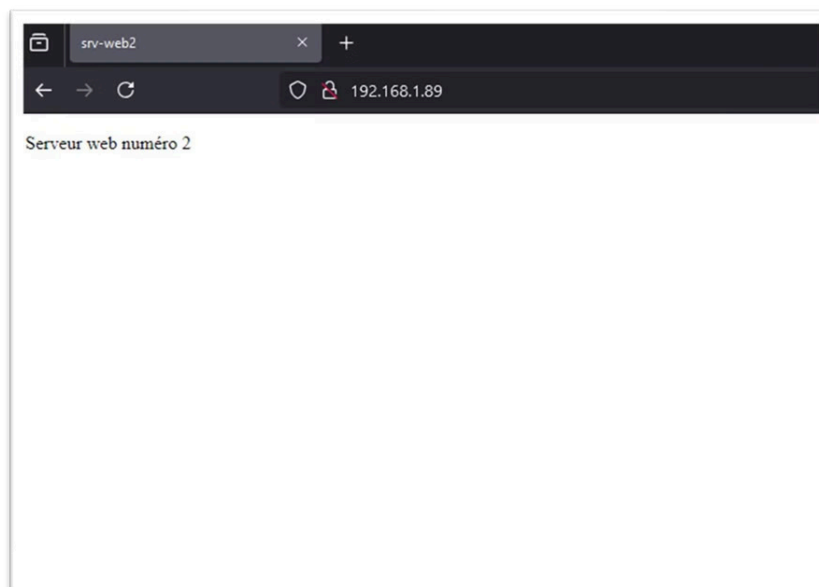

Test du « Reverse Proxy »

On peut désormais accéder à nos pages web grâce à l'adresse IP du HA Proxy (pour ma part : 192.168.1.89)

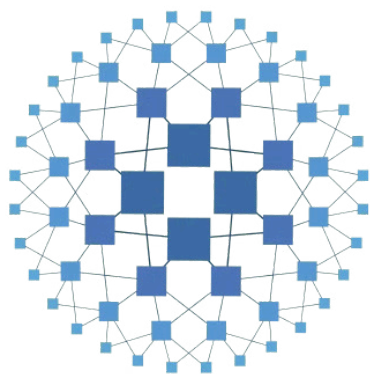


Lorsqu'on désactive le service apache2 sur un serveur web, l'autre prend le relais.

– Résultat après la désactivation du service apache2 du serveur srv-web1



Démonstration de la Haute Disponibilité



HAPROXY



Serveur 1 tombe en panne

Le service Apache2 est arrêté sur srv-web1



HAProxy détecte la panne

Redirection automatique du trafic

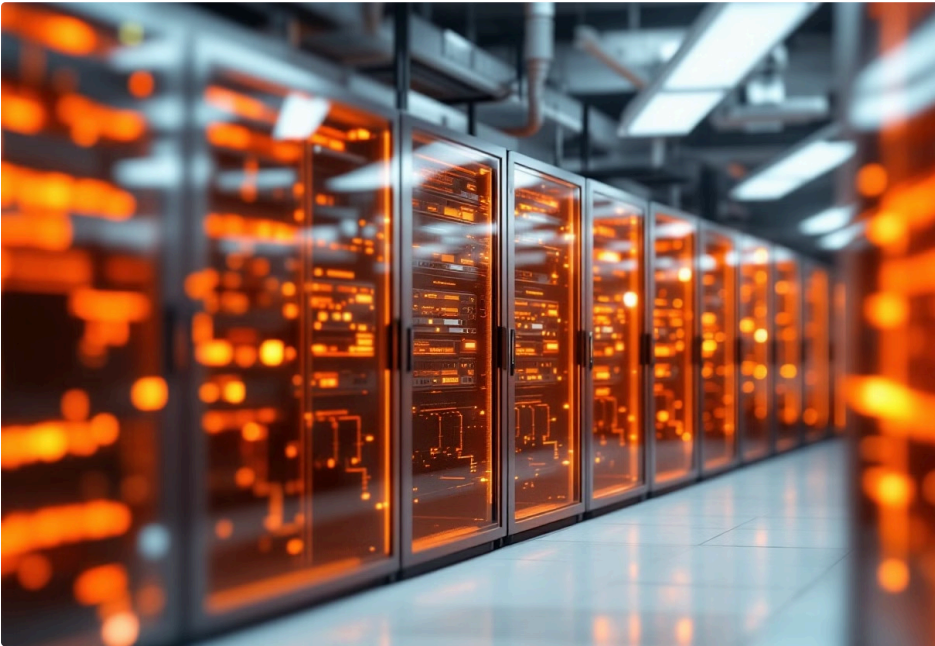


Continuité de service

Le site reste accessible via srv-web2

Cette démonstration prouve l'efficacité du système de haute disponibilité mis en place, garantissant une continuité de service même en cas de défaillance d'un des serveurs web.

Conclusion



Avec cette configuration, j'ai déployé un système garantissant la haute disponibilité et l'équilibrage de charge entre deux serveurs Apache grâce à HA Proxy. Ce projet améliore à la fois la résistance aux pannes et les performances des sites web, un atout majeur pour tout environnement professionnel.