

Situation professionnelle N°2 :

Serveur Apache redonder via HAProxy

### Description :

Une entreprise me demande de mettre en place une solution pour garder la continuité de son site web qui est héberger sur un serveur web apache.

# **SOMMAIRE :**

<u>Cahier des charges :</u> .....	Page : 3
<u>Le plan de l'infra :</u> .....	Page : 3
<u>Avant de commencer :</u> .....	Page : 4
<u>I) installation de Apache2:</u> .....	Page : 5
<u>II) création d'un site de test :</u> .....	Page : 7
<u>III) Tests des deux serveurs:</u> .....	Page : 10
<u>IV) configuration de la redondance sur le HAproxy:</u> .....	Page : 11
<u>V) Les tests:</u> .....	Page : 17

s

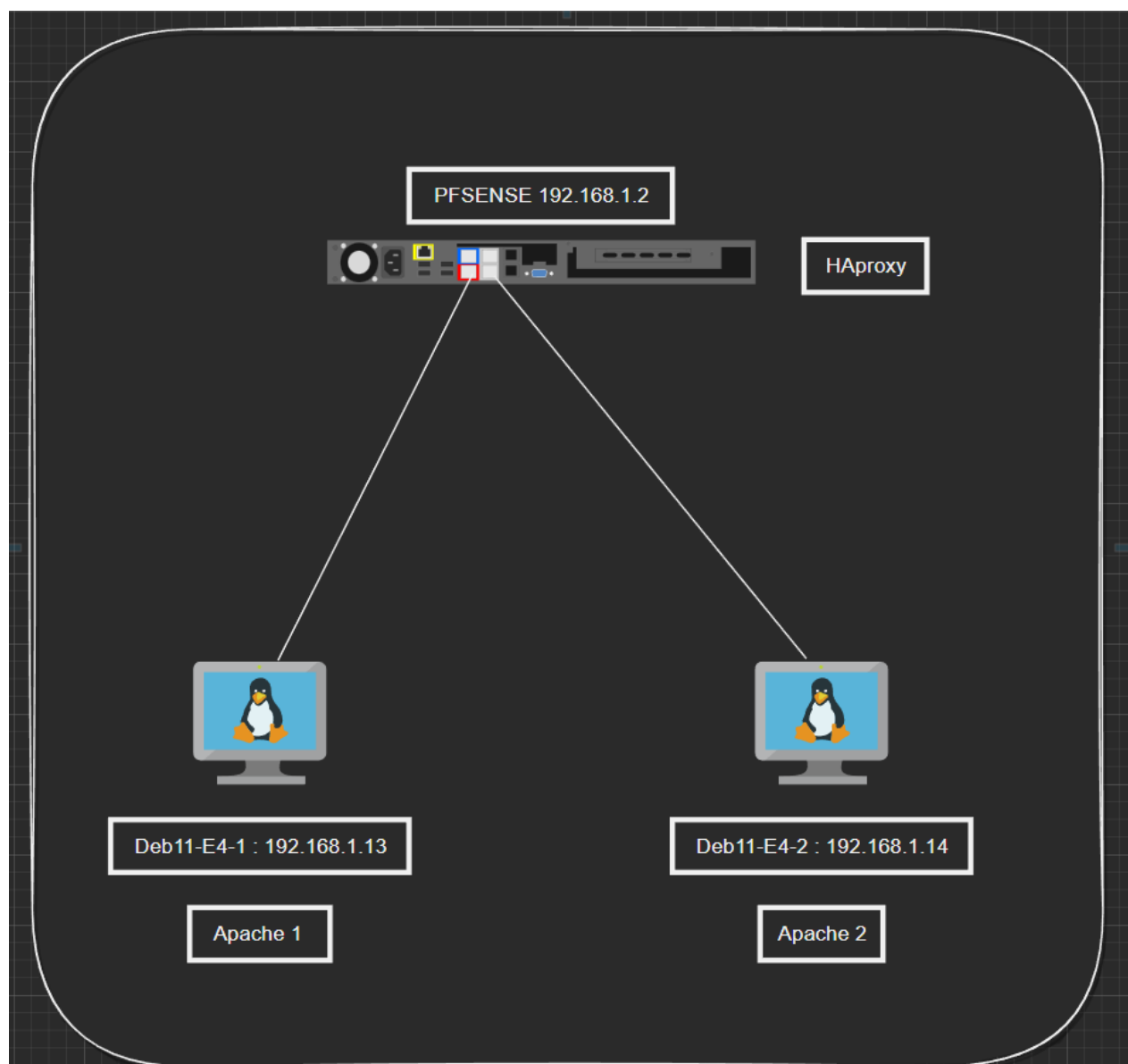
Cahier des charges :

La société info-connect souhaite améliorer sa continuité de service en cas de problème et me demande de trouver une solution pour augmenter la continuité de son site web.

Pour permettre la continuité de service des serveur apache nous allons mettre en place un HAproxy qui fera la redondance du site web en cas de pannes d'un des deux serveurs.

- Pour apache il sera installé sur deux Debian différente
- Pour le HAproxy il est installé dans le pfsense de l'entreprise et il est déjà fonctionnel avec un certificat auto-signé.

### Le plan de l'infra :



## Avant de commencer :

La commande sudo est à utiliser si vous n'êtes pas en root !

Il faut mettre les deux debian en ip static : `sudo nano /etc/network/interfaces`

Et de modifier le fichier en fonction de votre réseau !

DEBIAN N°1 : DEB11-E4-1

```
GNU nano 5.4
# 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
auto ens18
iface ens18 inet static
    address 192.168.1.13
    netmask 255.255.255.0
    gateway 192.168.1.2
    dns-nameservers 192.168.1.2 8.8.8.8
```

DEBIAN N°2 : DEB11-E4-2

```
GNU nano 5.4
# 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
auto ens18
iface ens18 inet static
    address 192.168.1.14
    netmask 255.255.255.0
    gateway 192.168.1.2
    dns-nameservers 192.168.1.2 8.8.8.8
```

## I) INSTALLATION DE Apache2:

La commande sudo est à utiliser si vous n'êtes pas en root !

- 1) Pour commencer l'installation de Apache on va mettre à jour la liste des paquets :

`sudo apt update`

```
root@DEB11-E4-2:~# apt update
Atteint :1 http://deb.debian.org/debian bullseye InRelease
Atteint :2 http://deb.debian.org/debian bullseye-updates InRelease
Atteint :3 http://security.debian.org/debian-security bullseye-security InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
root@DEB11-E4-2:~# █
```

```
root@DEB11-E4-1:~# apt update
Atteint :1 http://deb.debian.org/debian bullseye InRelease
Atteint :2 http://deb.debian.org/debian bullseye-updates InRelease
Atteint :3 http://security.debian.org/debian-security bullseye-security InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
root@DEB11-E4-1:~# █
```

- 2) Après avoir mis à jour la liste des paquets on va pouvoir installer le serveur web apache avec la commande :

`Sudo apt install apache2 -y`

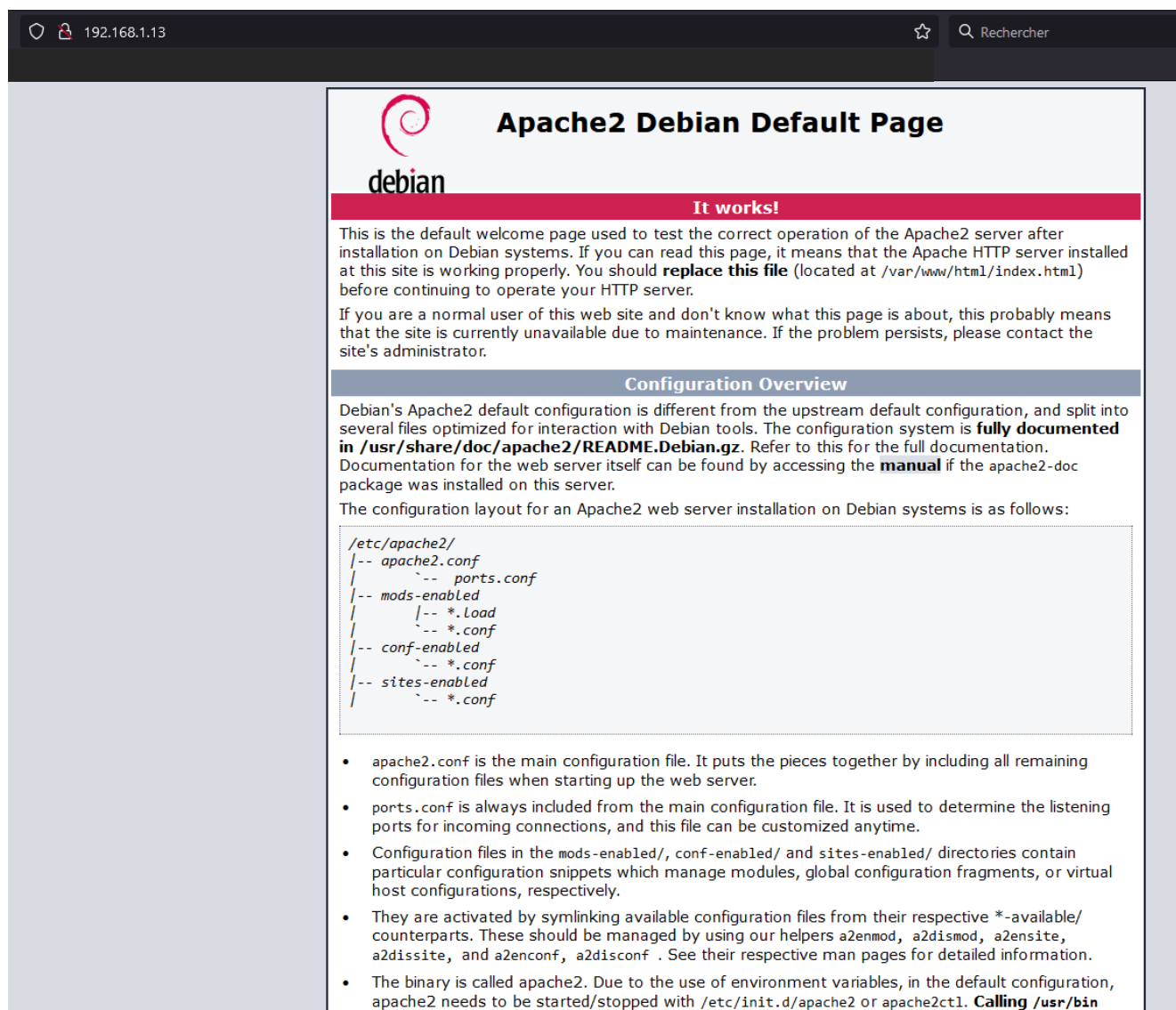
Puis pour vérifier que apache est bien installé on peut faire :

`Systemctl status apache2.services`

```
root@DEB11-E4-1:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2022-11-19 11:44:57 CET; 1min 43s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 3972 (apache2)
      Tasks: 55 (limit: 4591)
    Memory: 9.2M
       CPU: 44ms
    CGroup: /system.slice/apache2.service
            └─3972 /usr/sbin/apache2 -k start
              └─3974 /usr/sbin/apache2 -k start
                └─3975 /usr/sbin/apache2 -k start

nov. 19 11:44:57 DEB11-E4-1 systemd[1]: Starting The Apache HTTP Server...
nov. 19 11:44:57 DEB11-E4-1 apachectl[3971]: AH00558: apache2: Could not reliably determine
nov. 19 11:44:57 DEB11-E4-1 systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

- 3) Pour être sûr que le serveur web fonctionne on peut aller sur la page par défaut de apache :  
Exemple : <http://yourserverip/>



The screenshot shows a web browser window with the address bar displaying '192.168.1.13'. The page title is 'Apache2 Debian Default Page'. The page content includes the Debian logo, a red banner saying 'It works!', and a message stating that the Apache HTTP server is installed and working properly. It also provides information about the configuration system and a list of configuration files.

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

**Configuration Overview**

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|
|   |-- ports.conf
|   |-- mods-enabled
|   |   |-- *.load
|   |   |-- *.conf
|   |-- conf-enabled
|   |   |-- *.conf
|   |-- sites-enabled
|   |   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. Calling `/usr/bin`

Parfait notre serveur tourne !

## II) création d'un site de test :

- 1) Pour pouvoir tester la redondance nous allons créer deux fichiers html différents sur les deux Debian :

Pour commencer on va dans le dossier par défaut d'apache pour y ajouter notre dossier :

```
cd /var/www/html/
```

Pour créer le dossier on utilise la commande `mkdir` puis le nom du dossier :

Exemple : `mkdir dossier1`

```
mkdir site1
```

et

```
mkdir site2
```

- 2) Après cela on fait `ls` pour voir que notre dossier c'est bien créé :

DEB11-E4-1 :

```
root@DEB11-E4-1:~# cd /var/www/html/
root@DEB11-E4-1:/var/www/html# mkdir site1
root@DEB11-E4-1:/var/www/html# ls
index.html  site1
root@DEB11-E4-1:/var/www/html#
```

DEB11-E4-2 :

```
root@DEB11-E4-2:~# cd /var/www/html/
root@DEB11-E4-2:/var/www/html# mkdir site2
root@DEB11-E4-2:/var/www/html# ls
index.html  site2
root@DEB11-E4-2:/var/www/html#
```

- 3) On peut maintenant créer notre fichier .html  
On se place dans le dossier : `cd /site1` ou `/site2` (en fonction sur quel Debian on n'est)

Puis pour créer le .html il faut faire : `nano index.html` (va créer et ouvrir notre document en même temps !)

- 4) Une fois dans le fichier on va écrire la base d'un document html : on rajoute un paragraphe pour avoir un moyen de différencier les deux serveurs :

Pour DEB11-E4-1 :

```
<!DOCTYPE html>
<html>

<head>
</head>
<p>SITE 1 !</p>
<body>

</body>

</html>
```

Pour DEB11-E4-2 :

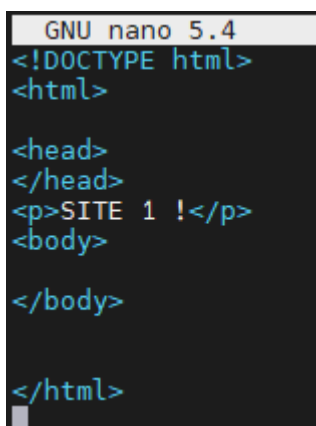
```
<!DOCTYPE html>
<html>

<head>
</head>
<p>SITE 2 ! </p>
<body>

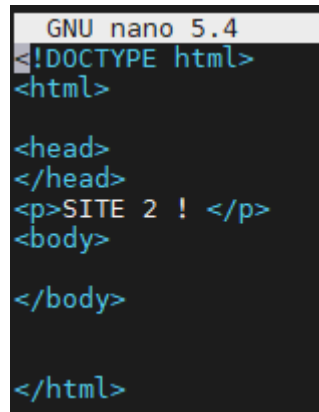
</body>

</html>
```

DEB11-E4-1 :



DEB11-E4-2 :



- 5) Pour enregistrer on fait ctrl + o puis entrée pour enregistrer puis ctrl + x pour quitter.



## II) configuration de Apache :

- 1) Une fois les deux fichiers html correctement remplis il faut configurer apache pour afficher la bonne page à la place de la page par défaut pour cela il faut se rendre dans le fichier de conf : `000-default.conf` Qui se trouve dans `/etc/apache2/sites-available/`

Pour cela faire : `cd /etc/apache2/sites-available/ puis nano 000-default.conf`

```
GNU nano 5.4
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

- 2) Pour changer de page html que le serveur va afficher il suffit de modifier le chemin du « DocumentRoot »

Par : `/var/www/html/site1` et `/site2` pour la deuxième Debian :

DEB11-E4-1 :

DEB11-E4-2 :

```
GNU nano 5.4
<VirtualHost *:80>
# The ServerName directive sets th
# the server uses to identify itse
# redirection URLs. In the context
# specifies what hostname must app
# match this virtual host. For the
# value is not decisive as it is u
# However, you must set it for any
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/site1
```

```
GNU nano 5.4
<VirtualHost *:80>
# The ServerName directive sets th
# the server uses to identify itse
# redirection URLs. In the context
# specifies what hostname must app
# match this virtual host. For the
# value is not decisive as it is u
# However, you must set it for any
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/site2
```

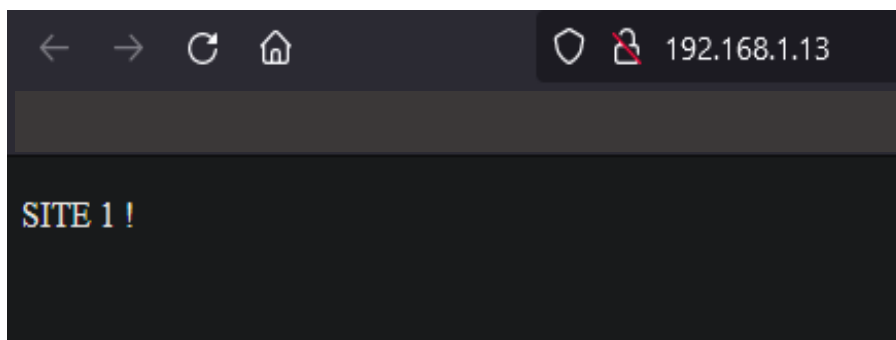
Pour finir faire `ctrl + o` puis entrée pour enregistrer puis `ctrl + x` pour quitter.

Puis `systemctl restart apache2.service` pour redémarrer

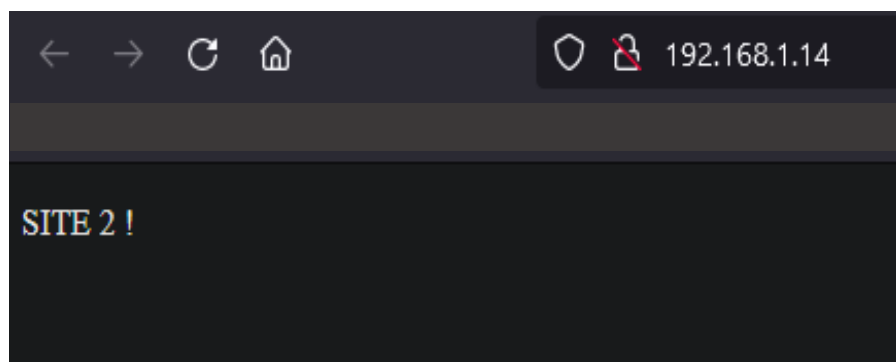
### III) Tests des deux serveurs :

- 1) Avant de passer à la redondance on va tester les deux serveurs séparément pour être sûr d'avoir deux page web différentes :

DEB11-E4-1 :



DEB11-E4-2 :



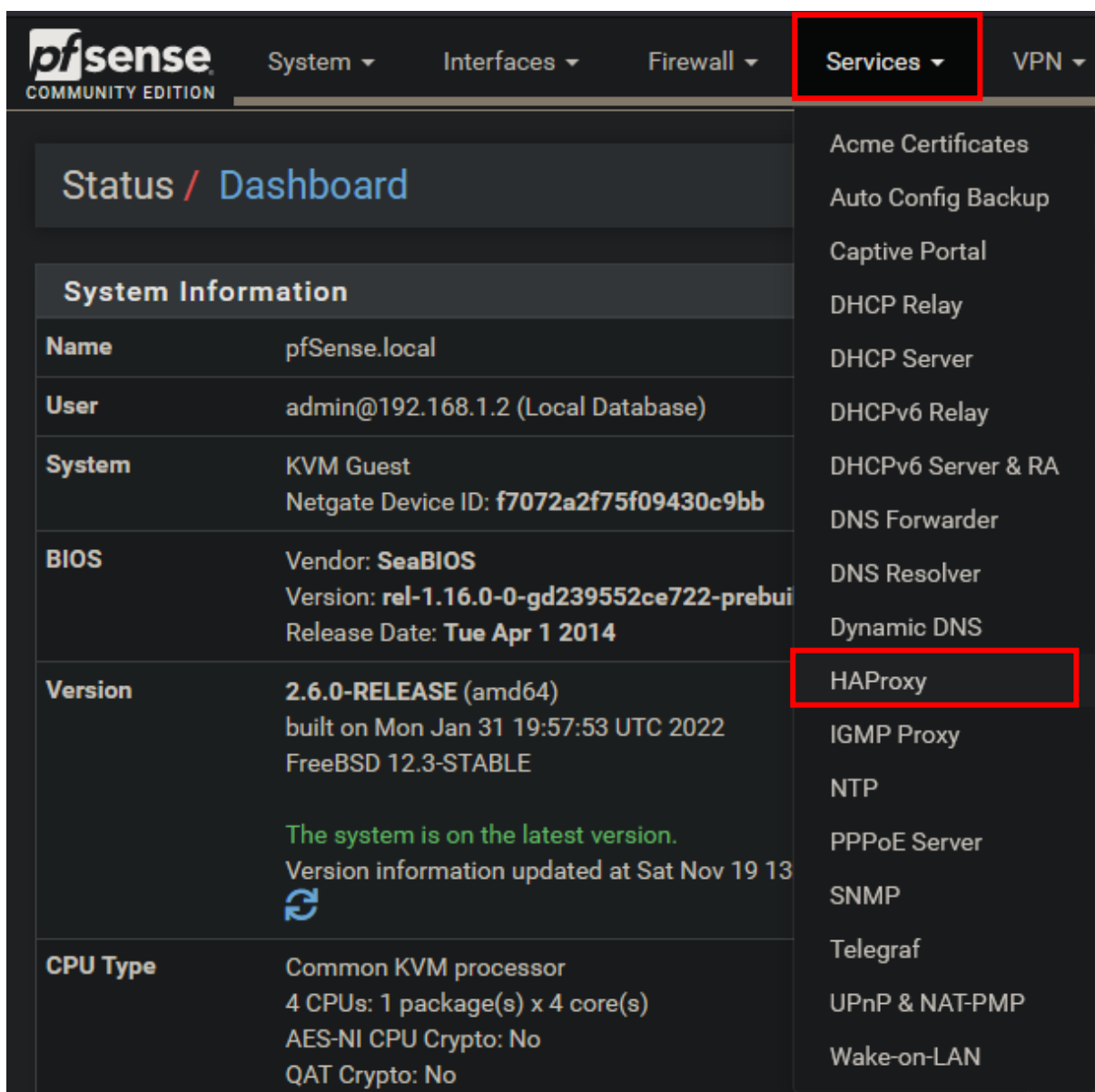
Les sites son bien différents on va pouvoir passer à la redondance !

## IV) configuration de la redondance sur le HAproxy :

*PS : le HAproxy étant déjà fonctionnel avec déjà un certificat ssl nous allons utiliser celui déjà présent !*

- 1) Pour commencer nous allons faire un backend c'est là que nous renseignons les ip et les ports des serveurs :

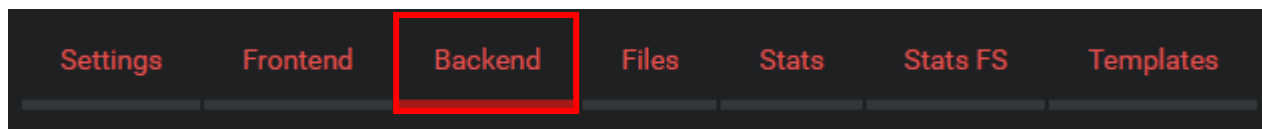
Pour cela cliquer sur service puis HAproxy :



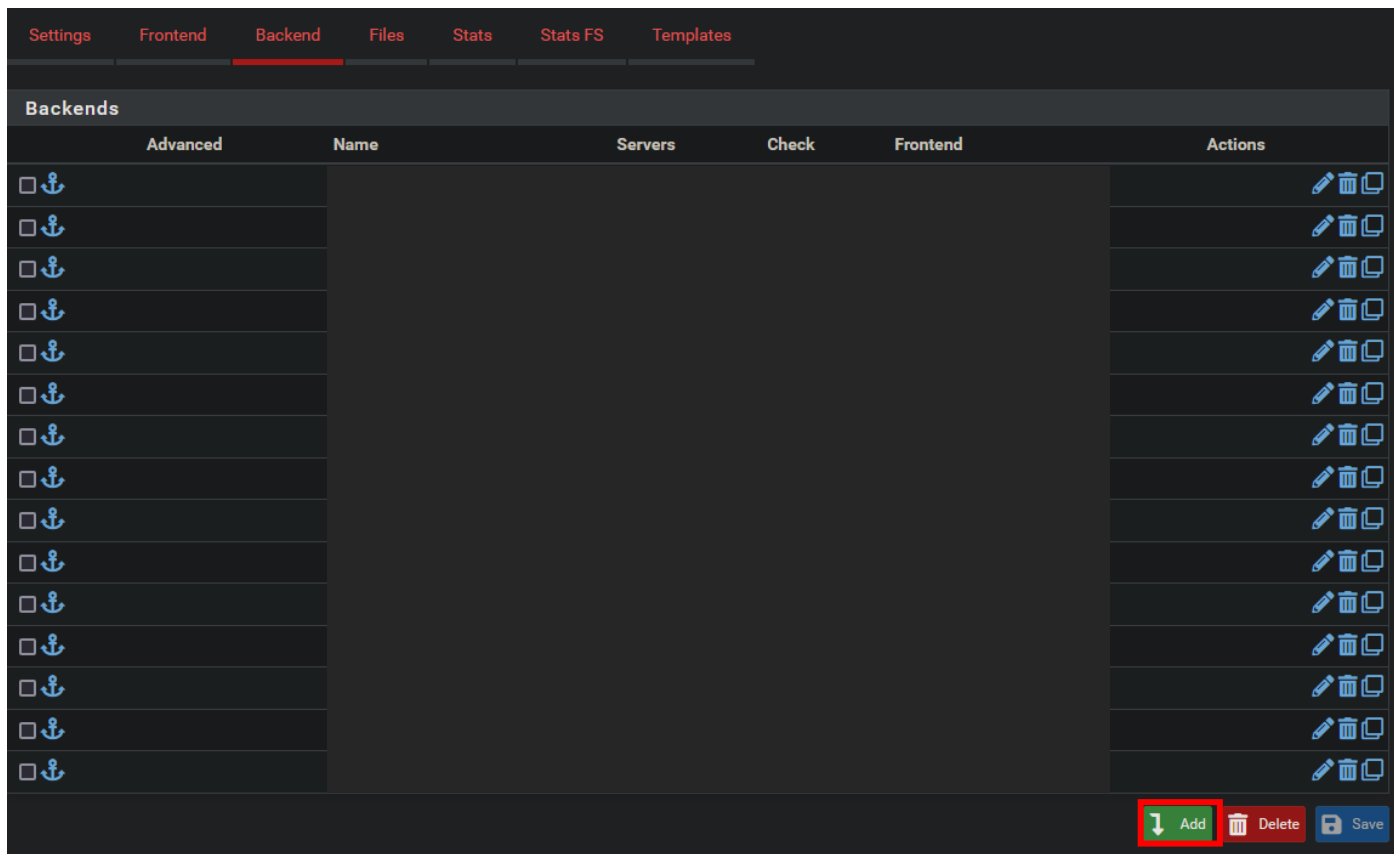
The screenshot shows the pfSense Community Edition web interface. The top navigation bar includes 'System', 'Interfaces', 'Firewall', 'Services', and 'VPN'. The 'Services' menu is highlighted. The left sidebar shows 'Status / Dashboard' and 'System Information'. Under 'System Information', the 'HAProxy' service is highlighted. The main content area displays system information:

System Information	
Name	pfSense.local
User	admin@192.168.1.2 (Local Database)
System	KVM Guest Netgate Device ID: f7072a2f75f09430c9bb
BIOS	Vendor: SeaBIOS Version: rel-1.16.0-0-gd239552ce722-prebui Release Date: Tue Apr 1 2014
Version	2.6.0-RELEASE (amd64) built on Mon Jan 31 19:57:53 UTC 2022 FreeBSD 12.3-STABLE  The system is on the latest version. Version information updated at Sat Nov 19 13:00:00 UTC 2022
CPU Type	Common KVM processor 4 CPUs: 1 package(s) x 4 core(s) AES-NI CPU Crypto: No QAT Crypto: No

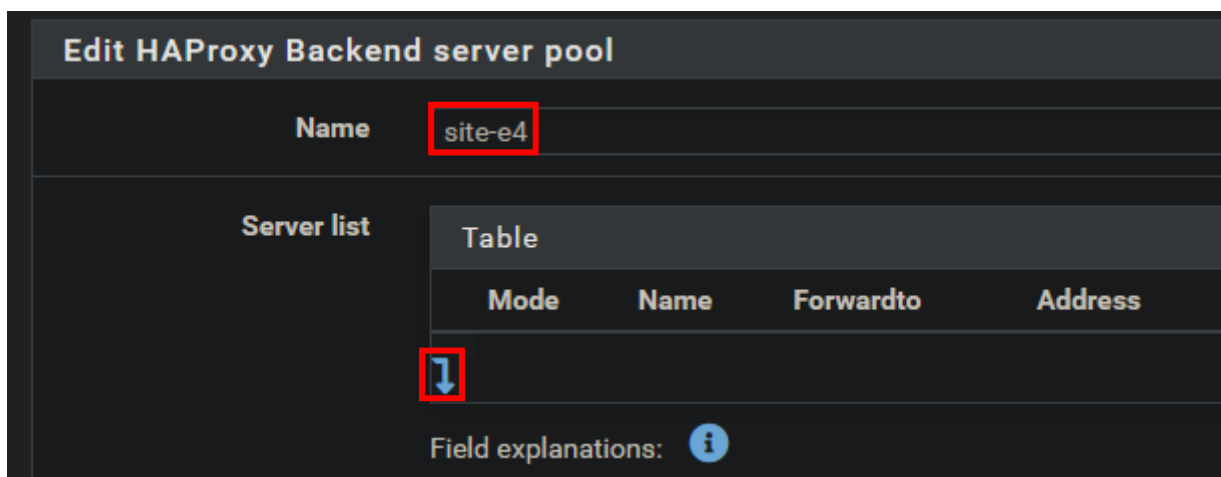
2) Puis dans Backend :



3) Puis on clique sur "Add" :



4) On ajoute un nom a notre backend puis on clique sur la flèche pour ajouter le premier serveur :



5) puis on rentre les info correspondantes à notre serveur N°1 (on oublie pas me certificat) :

Mode	Name	Forwardto	Address	Port	Encri
<input type="checkbox"/> active	site-1	Address+Port:	192.168.1.13	80	<input type="checkbox"/>
<p>Check certificate: <input type="checkbox"/> SSL servers only, The server certificate will be verified against the CA and CRL certificate configured below.</p> <p>Certificate check CN: <input type="text"/> SSL servers only, when set, must match the hostnames in the subject and subjectAlternateNames of the certificate provide</p> <p>CA: <input type="text"/> SSL servers only, Select the CA authority to check the server certificate against.</p> <p>CRL: <input type="text"/> None SSL servers only, Select the CRL to check revoked certificates.</p> <p>Client certificate: <input type="text"/> None SSL servers only, This certificate will be sent if the server send a client certificate request.</p> <p>Cookie: <input type="text"/> Persistence only, Used to identify server when cookie persistence is configured for the backend.</p> <p>Max conn: <input type="text"/> Tuning, If the number of incoming concurrent requests goes higher than this value, they will be queued</p> <p>Advanced: <input type="text"/> Advanced, Allows for adding custom HAProxy settings to the server. These are passed as written, use escaping where need</p> <p>DNS template count: <input type="text"/> If set configures this server item as a template to provision servers from dns/srv responses.</p>					

6) Pour le serveur N°2 on clique sur la flèche après avoir correctement renseigner les infos du serveur N°1 :

Petite différence pour le serveur N°2 on le mets en mode backup (il sera utilisé que si le premier à un souci)

Mode	Name	Forwardto	Address	Port	SSL Encrypt(SSL)	checks	Weight	Acti
<input type="checkbox"/> active	site-1	Address+Port:	192.168.1.13	80	no	no		
<p>CA: <input type="text"/> Acmecert: O=Let's Encrypt, CN=R3, C=US (CA: Acmecert: O=Internet Security Research Group, CN=ISRG Root X1, C=US) [Server cert]</p>								
<input type="checkbox"/> backup	site-2	Address+Pc	192.168.1.14	80	<input type="checkbox"/>	<input type="checkbox"/>		
<p>Check certificate: <input type="checkbox"/> SSL servers only, The server certificate will be verified against the CA and CRL certificate configured below.</p> <p>Certificate check CN: <input type="text"/> SSL servers only, when set, must match the hostnames in the subject and subjectAlternateNames of the certificate provided by the server.</p> <p>CA: <input type="text"/> SSL servers only, Select the CA authority to check the server certificate against.</p> <p>CRL: <input type="text"/> None SSL servers only, Select the CRL to check revoked certificates.</p>								

7) nos serveur utilise le port 80 alors en Health check method on choisi http

**Health checking**

Health check method: **HTTP**  
HTTP protocol to check on the servers health, can also be used for HTTPS servers(requires checking the SSL box for the

Check frequency:   
milliseconds  
For HTTP/HTTPS defaults to 1000 if left blank. For TCP no check will be performed if left empty.

Log checks: ☐ When this option is enabled, any change of the health check status or to the server's health will be logged.  
By default, failed health check are logged if server is UP and successful health checks are logged if server is DOWN, so the amount of additional information is limited.

Http check method: **OPTIONS**  
OPTIONS is the method usually best to perform server checks, HEAD and GET can also be used. If the server gets marked as down in the stats page then changing this to GET usually has the biggest chance of working, but might cause more processing overhead on the webserver and is less easy to filter out of its logs.

Url used by http check requests:   
Defaults to / if left blank.

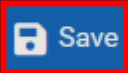
Http check version:   
Defaults to "HTTP/1.0" if left blank. Note that the Host field is mandatory in HTTP/1.1, and as a trick, it is possible to pass it after "\r\n" following the version string like this:  
`HTTP/1.1\r\nHost:\ www`  
Also some hosts might require an accept parameter like this:  
`HTTP/1.0\r\nHost:\ webservername:8080\r\nAccept:\ */*`

8) Pour finir on clique sur Save tout en bas et sur Apply change :

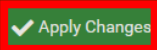
**Error files**

**HSTS / Cookie protection**

**Advanced settings**

 Save

The haproxy configuration has been changed.  
You must apply the changes in order for them to take effect.

 Apply Changes

9) Maintenant on va aller créer le frontend pour cela cliquer sur frontend :

**Settings** **Frontend** Backend Files Stats Stats FS Templates

On va créer une règle qui permet au frontend de savoir que quand le lien commence par "site-e4" par exemple il redirige les requêtes vers le bon backend donc vers nos serveurs :

Frontends									
Primary	Shared	On	Advanced	Name	Description	Address	Type	Backend	Actions
<input type="checkbox"/>		<input checked="" type="checkbox"/>		hugogergereau.fr	Fronted				

Default backend, access control lists and actions

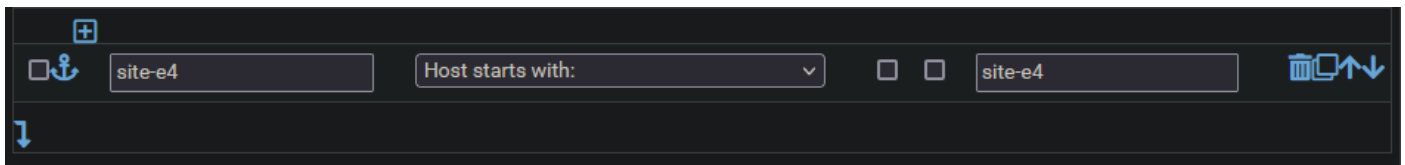
Access Control lists

Use these to define criteria that will be used with actions defined below to perform them only when certain conditions are met.

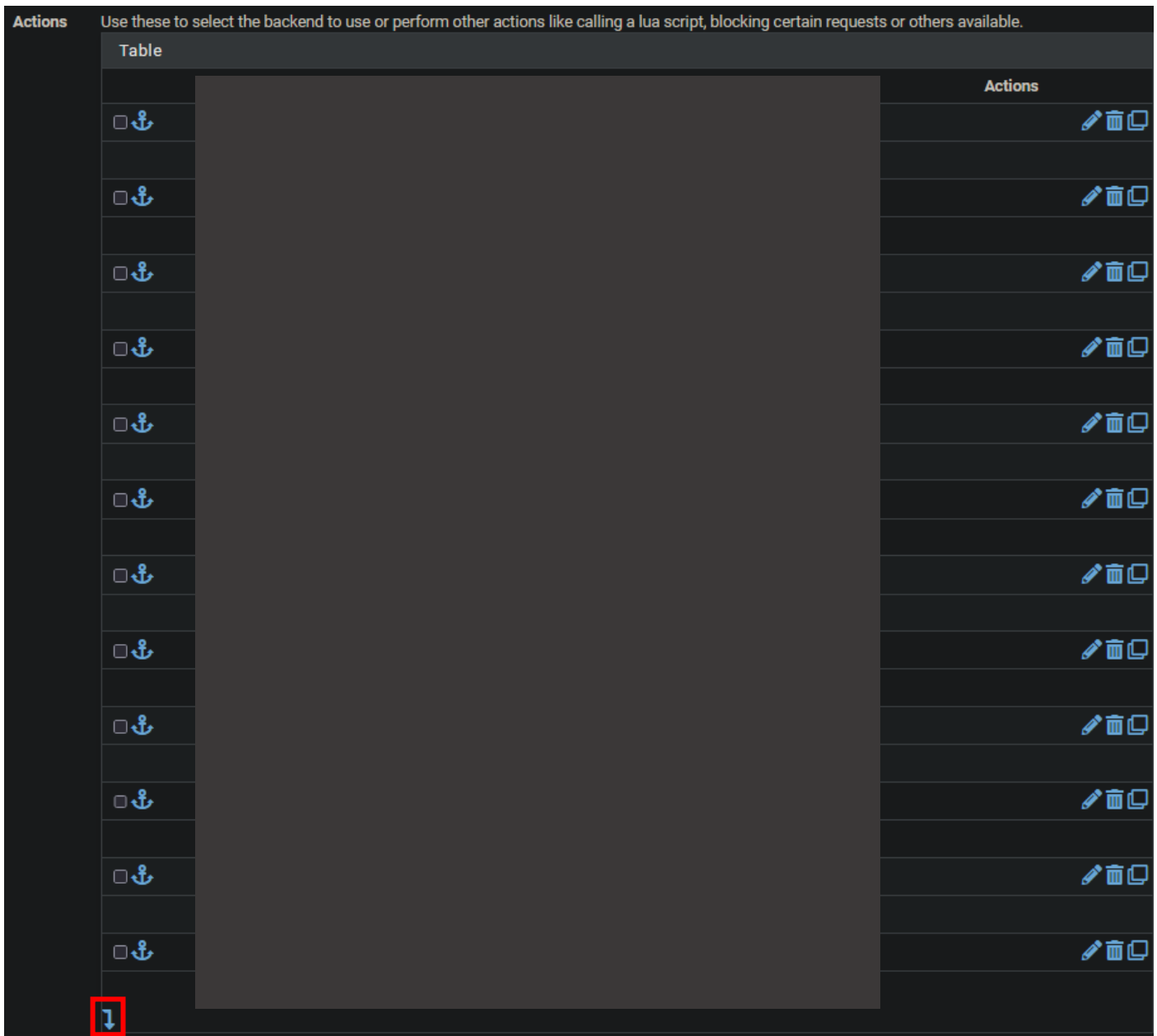
Table

		Actions
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		

- 12) Dans la partie de gauche on nomme notre entré, au milieu c'est la méthode qu'utilise cette entré pour nous se sera quand le nom de sous domaine commence par "site-e4" et à droite on met le nom du sous domaine souhaité.

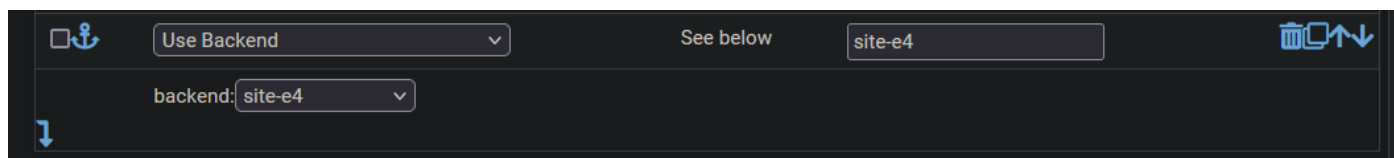


- 13) Dans cette partie nous allons définir quel backend sera utiliser par quel frontend :  
Pour ajouter notre lien entre le frontend et le backend cliquer sur le crayon :

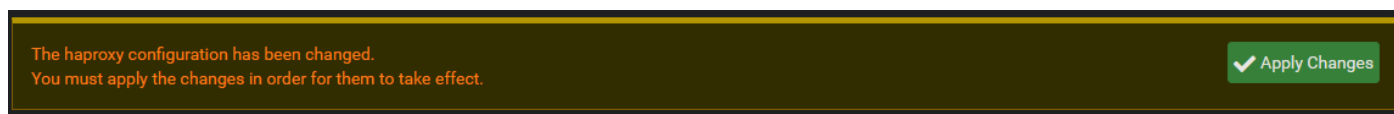
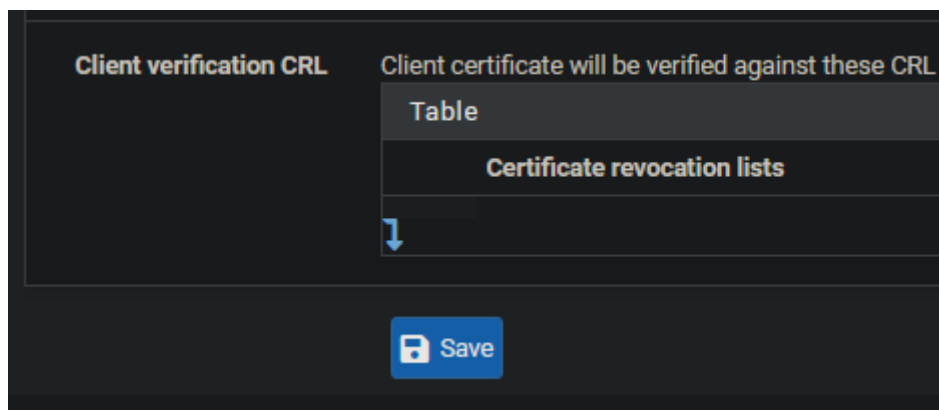




14) A droite ont rempli avec le nom de la règle créer juste avant, en bas ont choisi le backend qui correspond à nos serveurs :

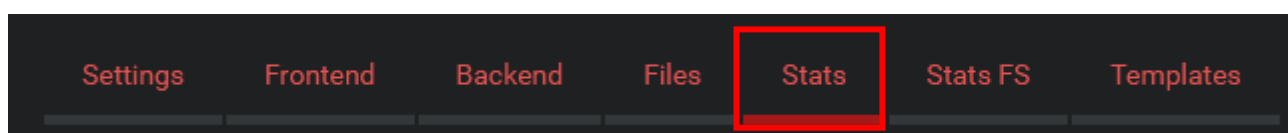


15) On clique sur Save puis sur Apply change :



## V) Les tests :

1) HAProxy a une interface pour voir les statuts de ces serveurs pour cela se rendre dans stats :

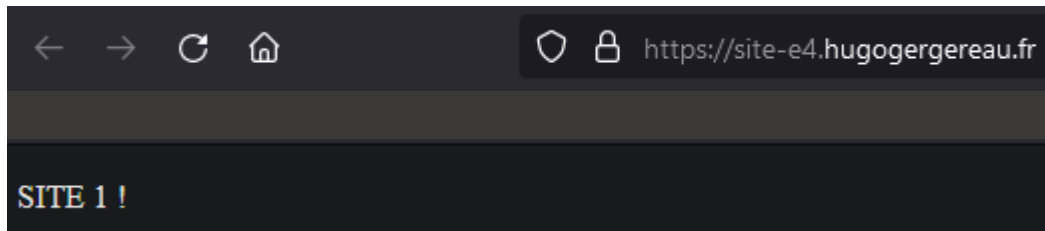


Puis trouver notre backend avec nos deux serveurs :

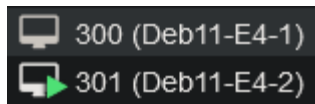
site-e4_ipvANY																															
<input type="checkbox"/>		Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrtl
<input type="checkbox"/>	site-1	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	1h21m UP	L7OK/200 in 1ms	1	Y	-	1	1	23s	-
<input type="checkbox"/>	site-2	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	1h21m UP	L7OK/200 in 1ms	1	-	Y	0	0	0s	-
<input type="checkbox"/>	Backend	0	0		0	0		0	0	200	0	0	?	0	0	0	0		0	0	0	0	1h21m UP		1	1	1		0	0s	

Ont voit bien que nos deux serveur le premier en vert car c'est le principal et le deuxième en gris au cas ou le premier aurait des problèmes :

- 2) Nous allons tester si le serveur web est bien accessible depuis internet grâce au HAproxy  
Pour cela dans votre navigateur préféré taper les infos que nous avons rentré dans la première partie du frontend :  
Exemple : "https://site-e4.youdomain.fr"  
Dans notre cas c'est : "https://site-e4.hugogergereau.fr.fr"

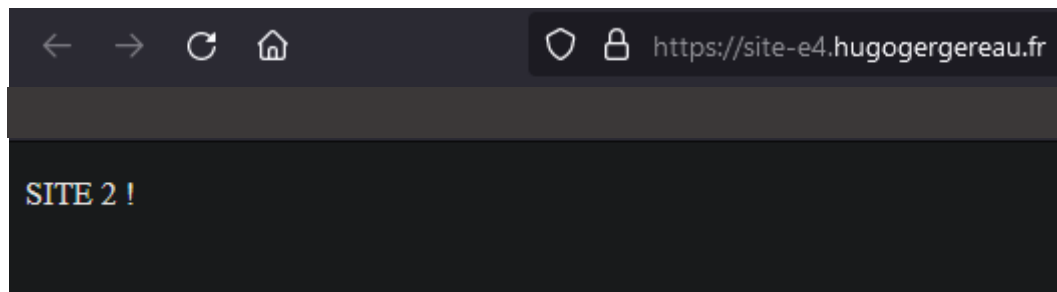


- 3) Maintenant on va simuler une panne sur le serveur N°1 :  
Pour simuler la panne je vais éteindre le vm N°1 et voir si le site N°2 prend le relais ! :



Une fois la vm stopper on actualise le site :

Et sans changer le lien du site c'est bien le serveur N°2 qui a pris le relais !



Voilà notre site web est maintenant redondant ce qui augmente la continuité de service !