

Equilibrage de charge avec Haproxy

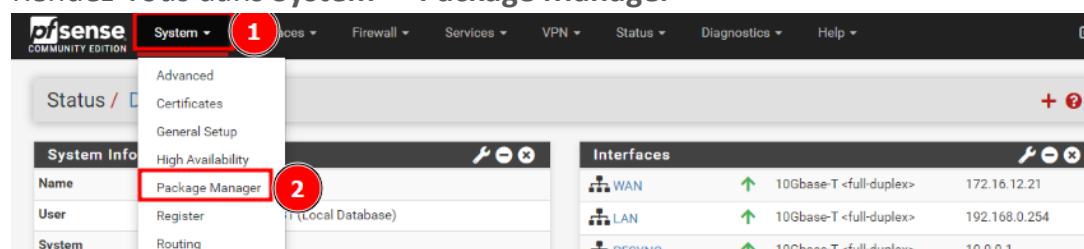
Version de PfSense : 2.7.0

1 – Installation de Haproxy	1
2 – Configuration de Haproxy - Backend	2
3 – Configuration de Haproxy – Frontend	4
4 – Configuration de Haproxy – options générales	6
5 – Création de la règle pare-feu.....	7
6 – Test du bon fonctionnement	8

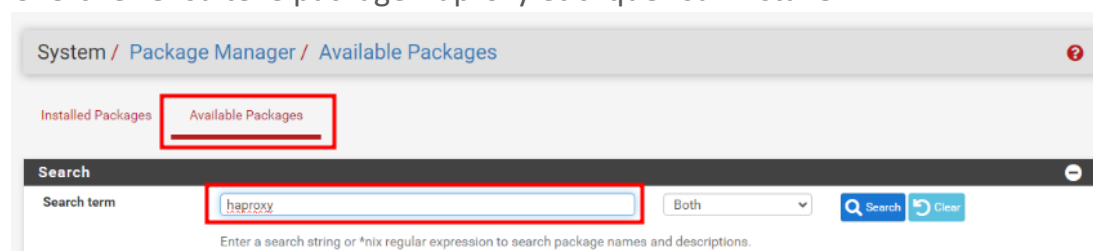
Nous allons voir dans cette procédure, comment faire de l'équilibrage de charge entre deux serveurs web grâce au package haproxy disponible sur PfSense.

1 – Installation de Haproxy

Rendez-vous dans **System -> Package Manager**

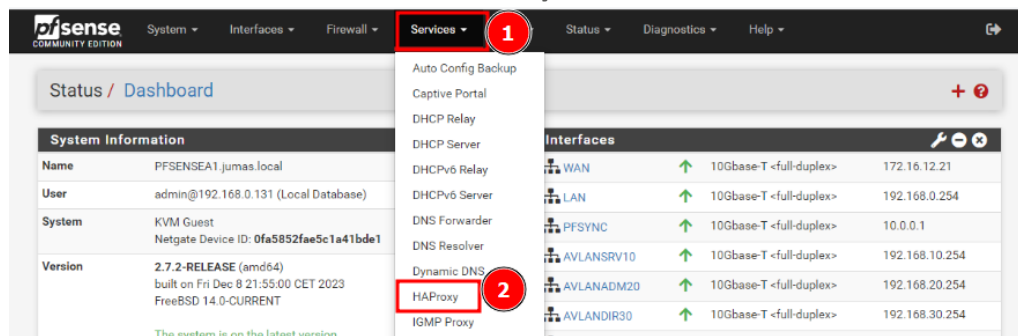


Cherchez ensuite le package Haproxy et cliquez sur installer

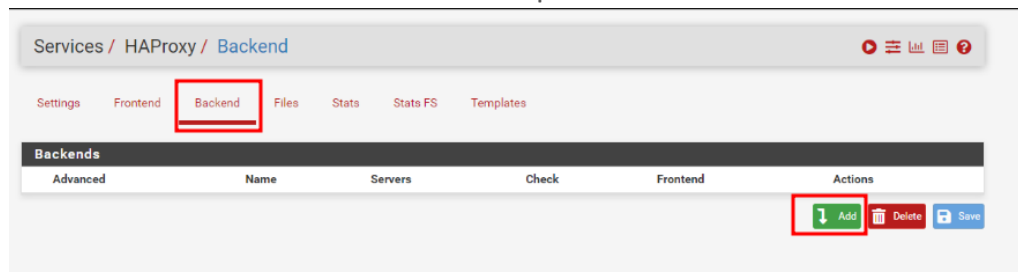


2 – Configuration de Haproxy- Backend

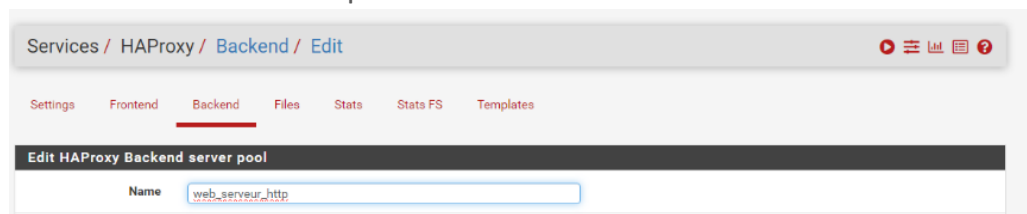
Rendez-vous dans **Services -> HAProxy**



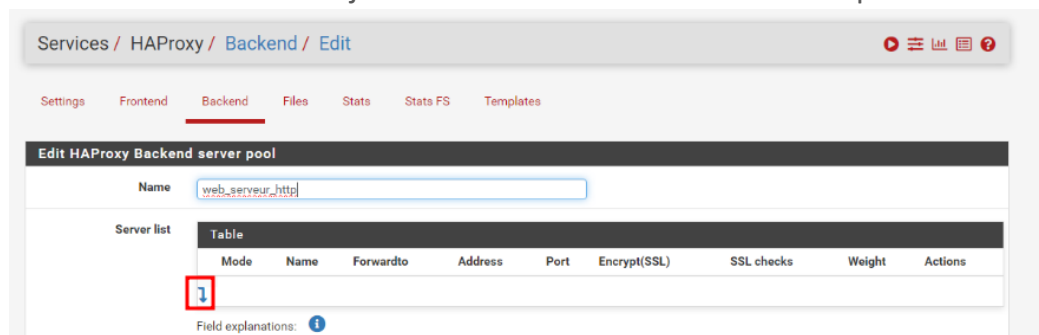
Allez maintenant dans **Backend** et cliquez sur **Add**



Donnez un nom à votre pool de serveur



Il va maintenant falloir ajouter nos deux serveurs web en cliquant sur la flèche



J'ajoute donc le premier serveur web

Edit HAProxy Backend server pool

Name: web_serveur_http

Server list

Mod	Name	Forwardto	Address	Port	Encrypt(SSL)	check
active	seur_web1	Address+Port	192.168.200.150	80		

Check certificate: ☐ SSL servers only. The server certificate will be verified against the CA and CRL certificate configured below.

Certificate check CN: SSL servers only, when set, must match the hostnames in the subject and subjectAlternateNames of the certificate provide

CA: SSL servers only. Select the CA authority to check the server certificate against.

CRL: SSL servers only. Select the CRL to check revoked certificates.

Client certificate: SSL servers only. This certificate will be sent if the server send a client certificate request.

Cookie: Persistence only. Used to identify server when cookie persistence is configured for the backend.

Max conn: Tuning. If the number of incoming concurrent requests goes higher than this value, they will be queued

Advanced: Advanced. Allows for adding custom HAProxy settings to the server. These are passed as written, use escaping where need

DNS template count: If set configures this server item as a template to provision servers from dns/srv responses.

Field explanations: [i](#)

J'ajoute ensuite le deuxième

Edit HAProxy Backend server pool

Name: web_serveur_http

Server list

Mod	Name	Forwardto	Address	Port	Encrypt(SSL)	check
active	seur_web2	Address+Port	192.168.200.140	80		

Check certificate: ☐ SSL servers only. The server certificate will be verified against the CA and CRL certificate configured below.

Certificate check CN: SSL servers only, when set, must match the hostnames in the subject and subjectAlternateNames of the certificate provide

CA: SSL servers only. Select the CA authority to check the server certificate against.

CRL: SSL servers only. Select the CRL to check revoked certificates.

Client certificate: SSL servers only. This certificate will be sent if the server send a client certificate request.

Cookie: Persistence only. Used to identify server when cookie persistence is configured for the backend.

Max conn: Tuning. If the number of incoming concurrent requests goes higher than this value, they will be queued

Advanced: Advanced. Allows for adding custom HAProxy settings to the server. These are passed as written, use escaping where need

DNS template count: If set configures this server item as a template to provision servers from dns/srv responses.

Field explanations: [i](#)

Maintenant, dans le menu **Loadbalancing options**, sélectionnez **Round robin**

Loadbalancing options (when multiple servers are defined)

Balance ☐ None
This allows writing your own custom balance settings into the advanced section. Or when you have no need for balancing with only 1 server.

☒ **Round robin**
Each server is used in turns, according to their weights. This is the smoothest and fairest algorithm when the server's processing time remains equally distributed. This algorithm is dynamic, which means that server weights may be adjusted on the fly for slow starts for instance.

☐ Static Round Robin
Each server is used in turns, according to their weights. This algorithm is as similar to roundrobin except that it is static, which means that changing a server's weight on the fly will have no effect. On the other hand, it has no design limitation on the number of servers, and when a server goes up, it is always immediately reintroduced into the farm, once the full map is recomputed. It also uses slightly less CPU to run (around -1%).

Dans le menu **Health checking**, sélectionnez comme ci-dessous

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 servers).

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: GET
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.

Une fois terminée, cliquez sur **Save** en bas de la page

Statistics

Stats Enabled ☐ Enables the haproxy statistics page (only used on "http" frontends)

Error files +

HSTS / Cookie protection +

Advanced settings +

Save

3 – Configuration de Haproxy – Frontend

Rendez-vous dans **Frontend** et cliquez sur **Add**

Settings Frontend Backend Files Stats Stats FS Templates

Frontends

Primary	Shared	On	Advanced	Name	Description	Address	Type	Backend	Actions
									<div> Add Delete Save </div>

Donnez un nom et une description

Edit HAProxy Frontend

Name: acces_web_serveur

Description: SRVA-WEB & SRVA-WEB2

Sélectionnez votre bonne interface WAN (dans mon cas c'est l'adresse VIP car j'ai un cluster de PfSense)

External address Define what ip:port combinations to listen on for incoming connections.

Listen address	Custom address	Port	SSL Offloading	Advanced	Actions
172.16.12.23 (VIP WAN)		80	<input type="checkbox"/>		

Ensuite, dans **Default backend**, sélectionnez le backend que nous avons créé auparavant

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.

Name	Expression	CS	Not	Value	Actions																		
<p>- 'CS' makes the string matches 'Case Sensitive' so www.domain.tld wil not be the same as WWW.domain.TLD</p> <p>- 'Not' makes the match if the value given is not matched</p> <p>Example:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Expression</th> <th>CS</th> <th>Not</th> <th>Value</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>Backend1acl</td> <td>Host matches</td> <td></td> <td></td> <td>www.yourdomain.tld</td> <td></td> </tr> <tr> <td>addHeaderAc</td> <td>SSL Client certificate valid</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>ac's with the same name will be 'combined' using OR criteria.</p> <p>For more information about ACLs please see HAProxy Documentation Section 7 - Using ACLs</p> <p>NOTE Important change in behaviour, since package version 0.32</p> <p>-ac's are no longer combined with logical AND operators, list multiple acl's below where needed.</p> <p>-ac's alone no longer implicitly generate use_backend configuration. Add 'actions' below to accomplish this behaviour.</p>						Name	Expression	CS	Not	Value	Actions	Backend1acl	Host matches			www.yourdomain.tld		addHeaderAc	SSL Client certificate valid				
Name	Expression	CS	Not	Value	Actions																		
Backend1acl	Host matches			www.yourdomain.tld																			
addHeaderAc	SSL Client certificate valid																						

Actions Use these to select the backend to use or perform other actions like calling a lua script, blocking certain requests or others available.

Action	Parameters	Condition acl names	Actions												
<p>Example:</p> <table border="1"> <thead> <tr> <th>Action</th> <th>Parameters</th> <th>Condition</th> </tr> </thead> <tbody> <tr> <td>Use Backend</td> <td>Website1Backend</td> <td>Backend1acl</td> </tr> <tr> <td>http-request header set</td> <td>Headername: X-HEADER-ClientCertValid</td> <td>addHeaderAc</td> </tr> <tr> <td></td> <td>New logformat value: YES</td> <td></td> </tr> </tbody> </table>				Action	Parameters	Condition	Use Backend	Website1Backend	Backend1acl	http-request header set	Headername: X-HEADER-ClientCertValid	addHeaderAc		New logformat value: YES	
Action	Parameters	Condition													
Use Backend	Website1Backend	Backend1acl													
http-request header set	Headername: X-HEADER-ClientCertValid	addHeaderAc													
	New logformat value: YES														

Default Backend web_serveur_http

If a backend is selected with actions above or in other shared frontends, no default is needed and this can be left to 'None'.

N'oubliez pas de sauvegarder vos modifications une fois terminée

NOTE: paste text into this box that you would like to pass behind each bind option.

Advanced pass thru

NOTE: paste text into this box that you would like to pass thru in the frontend.

4 – Configuration de Haproxy – options générales

Allez dans **Settings** et définissez un nombre de connexions maximum

Settings | Frontend | Backend | Files | Stats | Stats FS | Templates

General settings

☒ Enable HAProxy

Installed version: 2.8.3-86e043a

Maximum connections per process.

Sets the maximum per-process number of concurrent connections to X.
NOTE: setting this value too high will result in HAProxy not being able to allocate enough memory.
 Current 'System Tunables' settings:
 'kern.maxfiles': 129070
 'kern.maxfilesperproc': 116163
 Full memory usage will only show after all connections have actually been used.

When setting a high amount of allowed simultaneous connections you will need to add and/or increase the following two 'System Tunables' kern.maxfiles and kern.maxfilesperproc. For HAProxy alone set these to at least the number of allowed connections * 2 + 31. So for 100,000 connections these need to be 200,031 or more to avoid trouble, take into account that handles are also used by other processes when setting kern.maxfiles.

Connections	Memory usage
1	50 kB
1,000	48 MB
10,000	488 MB
100,000	4.8 GB

Calculated for plain HTTP connections, using ssl offloading will increase this.

Dans **Stats tab**, sélectionnez un port

Stats tab, 'internal' stats port

Internal stats port EXAMPLE: 2200

Sets the internal port to be used for the stats tab. This is bound to 127.0.0.1 so will not be directly exposed on any LAN/WAN/other interface. It is used to internally pass through the stats page. Leave this setting empty to remove the "HAProxyLocalStats" item from the stats page and save a little on resources.

Internal stats refresh rate Seconds, Leave this setting empty to not refresh the page automatically. EXAMPLE: 10

Sticktable page refresh rate Seconds, Leave this setting empty to not refresh the page automatically. EXAMPLE: 10

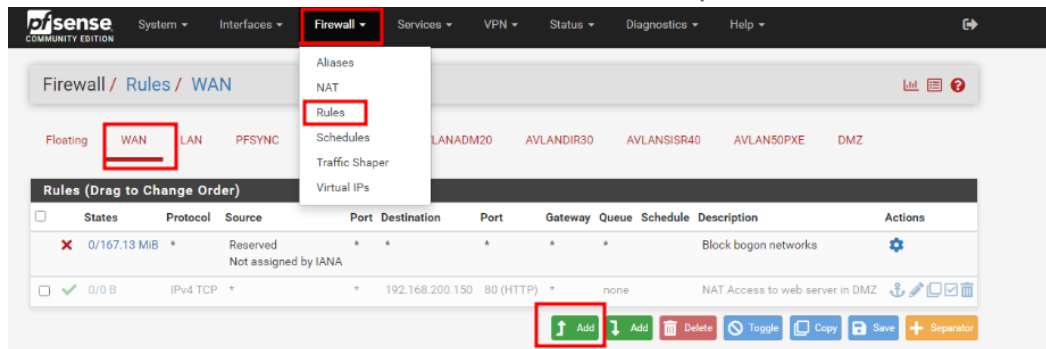
Si jamais vous utilisez un cluster de PfSense, vous pouvez cocher l'option HAProxy Sync.

Configuration synchronization

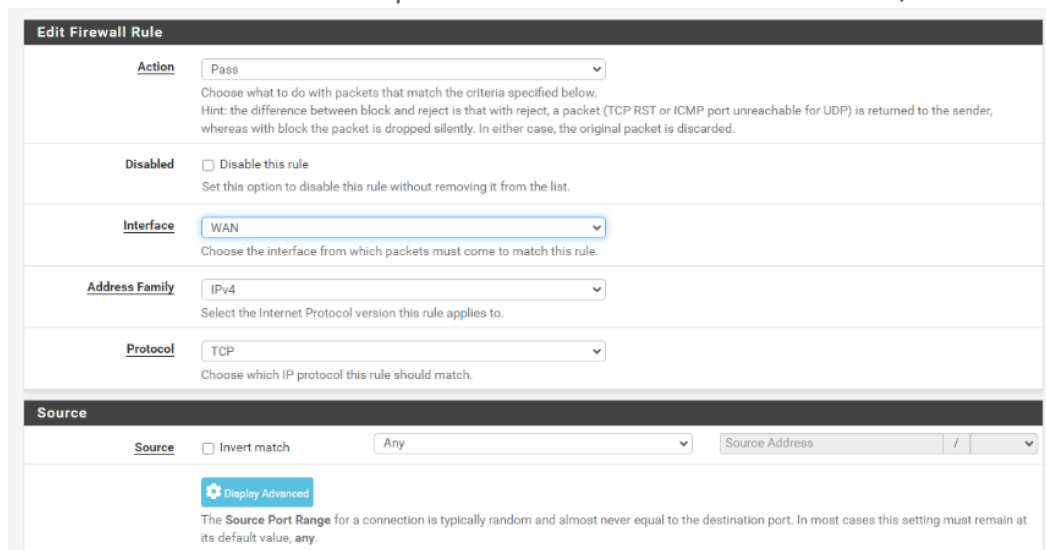
HAProxy Sync ☒ Sync HAProxy configuration to backup CARP members via XMLRPC.
 Note: The synchronisation host and password are those configured in pfSense main "System: High Availability Sync" settings.

5 – Création de la règle pare-feu

Rendez-vous dans **Firewall -> Rules -> WAN** et cliquez sur **Add**



Pour la source, vous pouvez soit décider de mettre une IP en particulier ou alors si vous êtes dans mon cas et que vos serveurs sont dans une DMZ, alors laissez any



Pour la source, sélectionnez votre interface WAN

