

Firewall Cluster & Reverse Proxy

Installer les VMs

Il nous faut 5 machines virtuelles. 2x PFSense, 1x Debian avec interface et 2x Debian sans interface. Il faut donner à chaque VM une carte réseau avec *Internal Network* sélectionné et aux VMs de PFSense en plus une deuxième carte réseau avec *Bridged Adapter* sélectionné.

Configurer les interfaces

- 1) Lancer les PFSenses et configurer l'interface du LAN (em1), ici: 10.1.1.1/16 et 10.1.1.2/16.

(les WAN sont 192.168.10.1 et 192.168.10.2 ici, ancienne image)

```
VirtualBox Virtual Machine - Netgate Device ID: 461950283b8000VirtualBox Virtual Machine - Netgate Device ID: 298b32e5341f60b82bc3
*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense *** *** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***
WAN (wan)      -> em0      -> v4/DHCP4: 172.25.20.245/22  WAN (wan)      -> em0      -> v4/DHCP4: 172.25.21.139/22
LAN (lan)      -> em1      -> v4: 10.1.1.2/16           LAN (lan)      -> em1      -> v4: 10.1.1.1/16
0) Logout (SSH only)      9) pfTop      0) Logout (SSH only)      9) pfTop
```

- 2) Lancer le Reverse Proxy et ouvrir la console afin de taper `sudo nano /etc/network/interfaces` et ajouter emp0s3 :

```
GNU nano 7.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

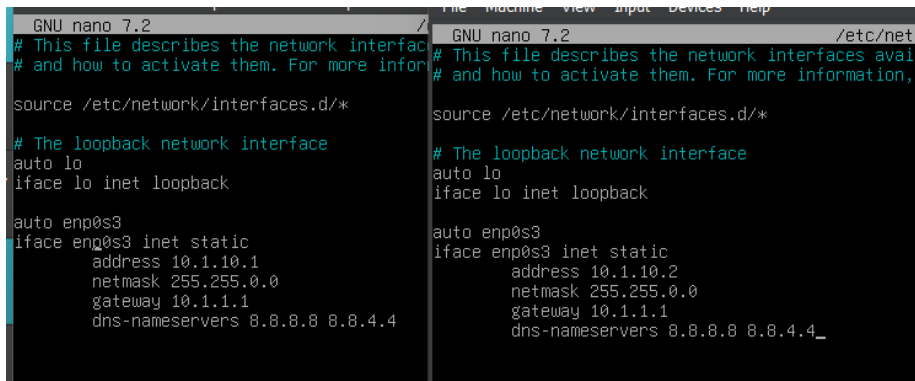
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto emp0s3
iface emp0s3 inet static
    address 10.1.9.1
    netmask 255.255.0.0
    gateway 10.1.1.1
    dns-nameservers 8.8.8.8 8.8.4.4
```

Puis `ctrl + x`, sauvegarder et redémarrer.

3) Lancer les Backend Servers et faire la même chose :



```
GNU nano 7.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see the man pages
of the /etc/network/interfaces file.

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
    address 10.1.10.1
    netmask 255.255.0.0
    gateway 10.1.1.1
    dns-nameservers 8.8.8.8 8.8.4.4

GNU nano 7.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see the man pages
of the /etc/network/interfaces file.

source /etc/network/interfaces.d/*

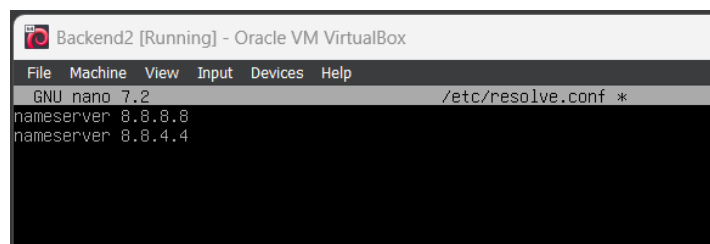
# The loopback network interface
auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
    address 10.1.10.2
    netmask 255.255.0.0
    gateway 10.1.1.1
    dns-nameservers 8.8.8.8 8.8.4.4
```

Puis `ctrl + x`, sauvegarder et redémarrer.

Configurer les DNS

Dans la console du Reverse Proxy et les Backend Servers, taper `sudo nano /etc/resolv.conf`, oui, resolv et pas resolve. Puis ajouter cela :



```
Backend2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 7.2 /etc/resolv.conf *
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Puis `ctrl + x` et sauvegarder.

Configurer les firewalls

Ouvrir un navigateur dans le Reverse Proxy et taper `10.1.1.1`. Les identifiants par défaut sont

Utilisateur : `admin` | Mot de Passe : `pfsense`

Configurer les firewalls en laissant toutes les paramètres par défaut. Après avoir fini, cliquer sur *Firewall* et puis sur *Rules*.

Par défaut il existe déjà un passage entre le WAN et le LAN, donc il nous faut maintenant un passage du LAN aux Backend Servers. Pour cela il faut ajouter une nouvelle règle où la source est `10.1.9.1` et la destination `10.1.10.0`.

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	1 /1.72 MB	*	*	*	LAN Address	443 80	*	*		Anti- Lockout Rule	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	0/0 B	IPv4 TCP	10.1.9.1	*	10.1.10.0/24	*	*	none	Traffic from Reverse Proxy to Backend Servers	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	0 /426 B	IPv4 *	LAN net	*	*	*	*	none	Default allow LAN to any rule	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	0/0 B	IPv6 *	LAN net	*	*	*	*	none	Default allow LAN IPv6 to any rule	

Maintenant il nous faut un passage entre le firewall et le WAN. Pour cela il faut cliquer sur *Firewall* et puis sur *NAT*. Mettre le *Destination Port Range* et le *Target Port Range* sur *HTTPS* ou *HTTP* et la *Target IP* sur *10.1.9.1*.

<input type="checkbox"/>	<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	80 (HTTP)	10.1.9.1	80 (HTTP)	Traffic from Firewall to Reverse Proxy	
--------------------------	-------------------------------------	-----	-----	---	---	-------------	-----------	----------	-----------	--	--

Créer le Firewall Cluster

Maintenant il faut cliquer sur *System* et puis sur *High Availability Sync*.

Crocher *Synchronize States* et sauvegarder.

Revenir sur *Firewall* et cliquer sur *Virtual IPs*. Ajouter une nouvelle IP et sélectionner *CARP* pour *Type*, *LAN* pour *Interface*, *10.1.1.100* pour *Adress* et aussi un mot de passe qui doit être utilisé plus tard par le deuxième firewall.

Revenir encore sur *High Availability Sync* et taper sous *XMLRPC* l'IP, le nom d'utilisateur et le mot de passe du deuxième firewall. Crocher *Firewall Rules* et *NAT* afin de copier les configurations sur l'autre.





Dans la barre de navigation taper *10.1.1.2* et aller sur *Virtual IPs* et taper la même chose.

10.1.1.100/16	LAN	IP Alias	
---------------	-----	----------	--





Revenir sur `sudo nano /etc/network/interfaces` et modifier pour ajouter la nouvelle passerelle qui est l'ip virtuelle, puis la même chose avec les backends serveurs.

```
auto enp0s3
iface enp0s3 inet static
    address 10.1.9.1
    netmask 255.255.0.0
    gateway 10.1.1.100
    dns-nameservers 8.8.8.8 8.8.4.4
```





Maintenant ajouter une deuxième ip virtuelle sur le WAN sur le réseau du WAN :

Virtual IP Address				
Virtual IP address	Interface	Type	Description	Actions
192.168.102.3/24	WAN	IP Alias		 
10.1.1.100/16	LAN	IP Alias		 

Pour faire passer le trafic de l'ip virtuelle WAN, ajouter un *Port Forwarding* du WAN :

<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	192.168.102.3	80 (HTTP)	10.1.9.1	80 (HTTP)	  
--------------------------	-------------------------------------	---	-----	-----	---	---	---------------	-----------	----------	-----------	---

Puis une règle pour rediriger le tout trafic HTTP vers l'ip virtuelle.

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	<input checked="" type="checkbox"/>	0/0 B	IPv4 TCP	*	*	192.168.102.3	80 (HTTP)	*	none		   

Installer Nginx et créer le Reverse Proxy

Taper dans la console de tous les VMs Debian :

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install nginx
```

```
sudo systemctl enable nginx
```

 (pas besoin mais ça lance Nginx automatiquement quand la machine démarre)

```
sudo start nginx
```

Maintenant sur le Reverse Proxy :

sudo nano /etc/nginx/sites-available/reverse-proxy

Mettre ça :

```
# Backend 1 (monsite.learn.local)
server {
    listen 80;
    server_name monsite.learn.local;
    location / {
        proxy_pass http://10.1.10.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

# Backend 2 (site.prod.local)
server {
    listen 80;
    server_name site.prod.local;
    location / {
        proxy_pass http://10.1.10.2;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Ctrl + X et sauvegarder.

sudo ln -s /etc/nginx/sites-available/reverse_proxy /etc/nginx/sites-enabled/

sudo nginx -t (pour tester les configs)

sudo systemctl reload nginx

sudo nano /etc/hosts

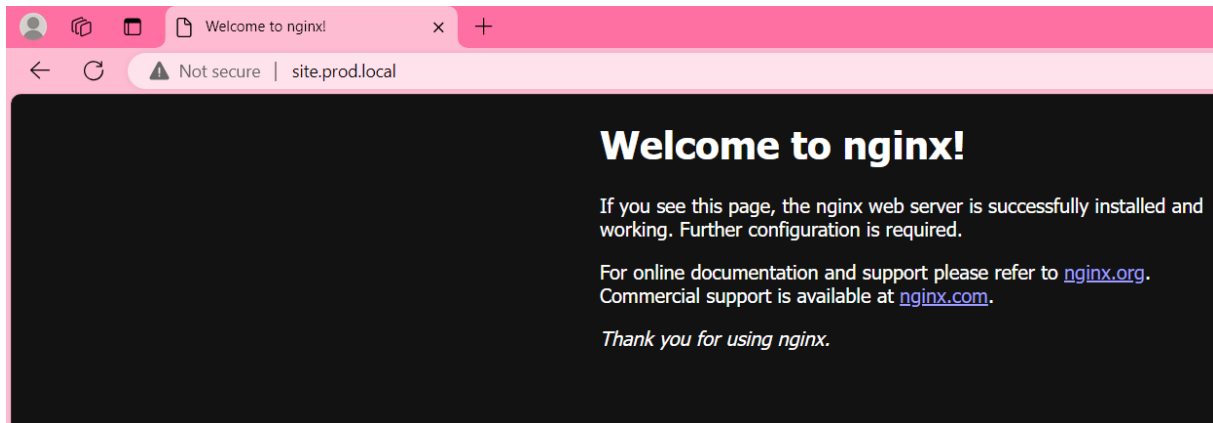
Ajouter les deux domaines :

```
GNU nano 7.2 /etc/hosts *
127.0.0.1    localhost
127.0.1.1    aldin
10.1.9.1 monsite.learn.local
10.1.9.1 site.prod.local
# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

Maintenant sur la machine hôte, ouvrir `C:\Windows\System32\drivers\etc\hosts` et ajouter l'adresse ip virtuelle et les domaines :

```
192.168.102.3    monsite.learn.local
192.168.102.3    site.prod.local
```

Le site est maintenant accessible par l'hôte.



Si le premier firewall est éteint, le site sera toujours disponible par le deuxième.

