

2 ALT SIO SISR	TP SERVEURS WEB	DATE : 15/01/24
----------------	--------------------	-----------------

## SOMMAIRE :

- 1) Serveur WEB 1
- 2) Serveur WEB 2 et Configuration connexion bdd via WEB 1
- 3) HA PROXY
- 4) HA PROXY #2 et Pacemaker

1)

Après avoir fait l'installation de la machine rocky, il faut taper cette commande pour installer les packages nécessaires pour le serveur web :

```
dnf install -y php-fpm mariadb-server php php-mysqld nginx
```

```
[root@localhost ~]# dnf install -y php-fpm mariadb-server php php-mysqld nginx
```

Ensuite il faut démarrer les services et les rendre actif avec cette commande :

```
Systemctl start ...
Systemctl enable ...
```

```
[root@localhost ~]# systemctl start php-fpm
[root@localhost ~]# systemctl enable php-fpm
Created symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service + /usr/lib/systemd/system/php-fpm.service.
[ 1009.471777] systemd-rc-local-generator[14153]: /etc/rc.d/rc.local is not marked executable, skipping.
[root@localhost ~]#
```

```
[root@localhost ~]# systemctl start nginx
[root@localhost ~]# systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service + /usr/lib/systemd/system/nginx.service.
[ 1111.723083] systemd-rc-local-generator[14376]: /etc/rc.d/rc.local is not marked executable, skipping.
[root@localhost ~]#
```

```
[root@localhost ~]# systemctl start mariadb
[root@localhost ~]# systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service + /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service + /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service + /usr/lib/systemd/system/mariadb.service.
[ 1083.043815] systemd-rc-local-generator[14329]: /etc/rc.d/rc.local is not marked executable, skipping.
[root@localhost ~]#
```

On ajoute les ports d'écoute pour accéder à la page d'accueil de nginx :

```
Firewall-cmd --add-port=80/tcp
```

Firewall-cmd --add-port=8080/tcp
Firewall-cmd --reload

```
[root@WEB1 ~]# firewall-cmd --reload
success
[root@WEB1 ~]# firewall-cmd --add-port=8080/tcp
success
[root@WEB1 ~]# firewall-cmd --add-port=80/tcp
success
[root@WEB1 ~]# firewall-cmd --reload
success
```

Voici le result :



Attaquons-nous à mariadb. Faire les commandes ci-dessous pour activer le service mariadb et voir son status :

systemctl enable --now mariadb.service
Systemctl status mariadb.service

```
[root@WEB1 ~]# systemctl enable --now mariadb.service
[ 2665.621425] systemd-rc-local-generator[4534]: /etc/rc.d/rc.local is not marked executable, skipping.
[root@WEB1 ~]# systemctl status mariadb.service
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
   Active: active (running) since Mon 2024-01-15 11:28:46 CET; 44min ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 939 (mariadb)
    Status: "Taking your SQL requests now..."
     Tasks: 0 (limit: 10020)
    Memory: 101.2M
       CPU: 1.540s
   CGroup: /system.slice/mariadb.service
           └─939 /usr/libexec/mariadb --basedir=/usr

Jan 15 11:28:43 WEB1 systemd[1]: Starting MariaDB 10.5 database server...
Jan 15 11:28:44 WEB1 mariadb-prepare-db-dir[9801]: Database MariaDB is probably initialized in /var/lib/mysql already, nothing is done.
Jan 15 11:28:44 WEB1 mariadb-prepare-db-dir[9801]: If this is not the case, make sure the /var/lib/mysql is empty before running mariadb-prepare-db-dir.
Jan 15 11:28:46 WEB1 systemd[1]: Started MariaDB 10.5 database server.
[root@WEB1 ~]#
```

On lance l'installation de mysql\_secure avec cette commande :

Mysql_secure_installation
---------------------------

```
[root@WEB1 ~]# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
... Success!

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.
```

```

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[root@WEB1 ~]#

```

Ensuite nous configurons mariadb comme ci :

Mysql -u root -p
Create database ECOM ;
Create user 'user'@'web1' identified by 'ECOM_59';
GRANT ALL PRIVILEGES ON ECOM.* TO 'user'@'web1';
FLUSH PRIVILEGES;
SHOW DATABASES;

```

[MariaDB [(none)]]> create database ECOM;
[MariaDB [(none)]]> CREATE USER 'noa'@'localhost' IDENTIFIED BY 'Fontaine_59';
Query OK, 0 rows affected (0.001 sec)

[MariaDB [(none)]]> GRANT ALL PRIVILEGES TO ECOM.* TO 'noa'@'localhost';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near 'TO
ECOM.* TO 'noa'@'localhost'' at line 1
[MariaDB [(none)]]> GRANT ALL PRIVILEGES ON ECOM.* TO 'noa'@'localhost';
Query OK, 0 rows affected (0.002 sec)

[MariaDB [(none)]]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.002 sec)

[MariaDB [(none)]]> █
[MariaDB [(none)]]> SHOW DATABASES;
+-----+
| Database |
+-----+
| ECOM      |
| information_schema |
| mysql     |
| performance_schema |
+-----+
4 rows in set (0.051 sec)

```

ATTENTION NE PAS OUBLIER LES ';' sinon erreur

---

On installe nano pour éditer les fichiers de configuration ou autre avec cette commande :

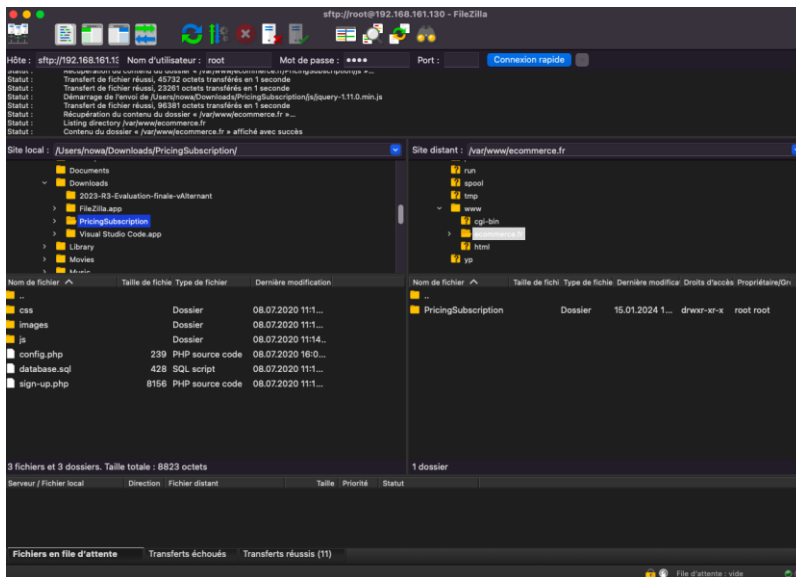
```
Dnf install -y nano
```

```
[noa@WEB1 nginx]$ sudo dnf install -y nano
```

On créer le répertoire ecommerce.fr dans /var/www/

```
Mkdir -p /var/www/ecommerce.fr/
```

Ensuite on ouvre Filezilla et on met le dossier fournis avec le site internet de ecommerce.fr dans le répertoire créer :



On se déplace ensuite dans le dossier de conf de php-fpm puis nous allons modifier celui-ci comme ci :

```
[[root@WEB1 etc]# cd /etc/php-fpm.d
[root@WEB1 php-fpm.d]# nano www.conf
```

```
; RPM: apache user chosen to provide access to the same directories as httpd
user = nginx
; RPM: Keep a group allowed to write in log dir.
group = nginx
```

Enregistrez et fermez.

On peut maintenant démarrer et activer le service php-fpm avec ces commandes :

Systemctl restart php-fpm
---------------------------

Systemctl enable php-fpm
--------------------------

```
[root@WEB1 php-fpm.d]# systemctl restart php-fpm
[root@WEB1 php-fpm.d]# systemctl enable php-fpm
[root@WEB1 php-fpm.d]# systemctl status php-fpm
● php-fpm.service - The PHP FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; enabled; preset: disabled)
   Active: active (running) since Mon 2024-01-15 18:32:51 CET; 15s ago
     Main PID: 2145 (php-fpm)
    Status: "Processes active: 0, idle: 5, Requests: 0, slow: 0, Traffic: 0req/sec"
      Tasks: 6 (limit: 10828)
     Memory: 12.0M
        CPU: 73ms
    CGroup: /system.slice/php-fpm.service
            └─2145 "php-fpm: master process (/etc/php-fpm.conf)"
              └─2147 "php-fpm: pool www"
                └─2148 "php-fpm: pool www"
                  └─2149 "php-fpm: pool www"
                    └─2150 "php-fpm: pool www"
                      └─2151 "php-fpm: pool www"

Jan 15 18:32:51 WEB1 systemd[1]: Starting The PHP FastCGI Process Manager...
Jan 15 18:32:51 WEB1 systemd[1]: Started The PHP FastCGI Process Manager.
lines 1-18/18 (END)
```

On se déplace ensuite dans le répertoire de configuration de nginx pour pouvoir créer le fichier de conf du site ecommerce.fr :

```
Cd /etc/nginx/conf.d
```

```
Nano site_ecommerce.fr.conf
```

```
[root@WEB1 ~]# cd /etc/nginx/conf.d/
[root@WEB1 conf.d]#
```

Et on met tous cela dans le fichier de conf créer

```
server { listen 80;

listen [::]:80;

root /var/www/site_commerce.fr/;
index index.html index.htm index.nginx-debian.html sign-up.php;

server_name ecommerce.fr ; location ~* \.php$ {

fastcgi_pass unix:/run/php-fpm/www.sock;
include fastcgi_params;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;

    fastcgi_param SCRIPT_NAME $fastcgi_script_name;
}

access_log /var/log/nginx/access_ecommerce.fr.log; error_log
/var/log/nginx/error_ecommerce.fr.log; location / {

try_files $uri $uri/ =404; }
```

--

Il faut maintenant faire la configuration dans le fichier de conf de php pour établir la connexion avec le moteur de base de données.

Voici les commandes :

Cd /var/www/site_ecommerce.fr/PricingSubscription/
--

Nano config.php
-----------------

```
[[root@WEB1 ~]# cd /var/www/site_ecommerce.fr
[root@WEB1 site_ecommerce.fr]# nano config.php
```

Voici comment le compléter avec mes informations :

GNU nano 5.6.1	config.php	Modified
<pre>&lt;?php \$SETTINGS["mysql_user"]="noa"; \$SETTINGS["mysql_pass"]="Fontaine_59"; \$SETTINGS["hostname"]="localhost"; \$SETTINGS["mysql_database"]="ECOM"; \$SETTINGS["data_table"]="registrations"; \$SETTINGS["paypal_address"]="email@domain.com"; ?&gt;</pre>		

```
[[root@WEB1 site_ecommerce.fr]# mysql -u noa -p
[Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.5.22-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

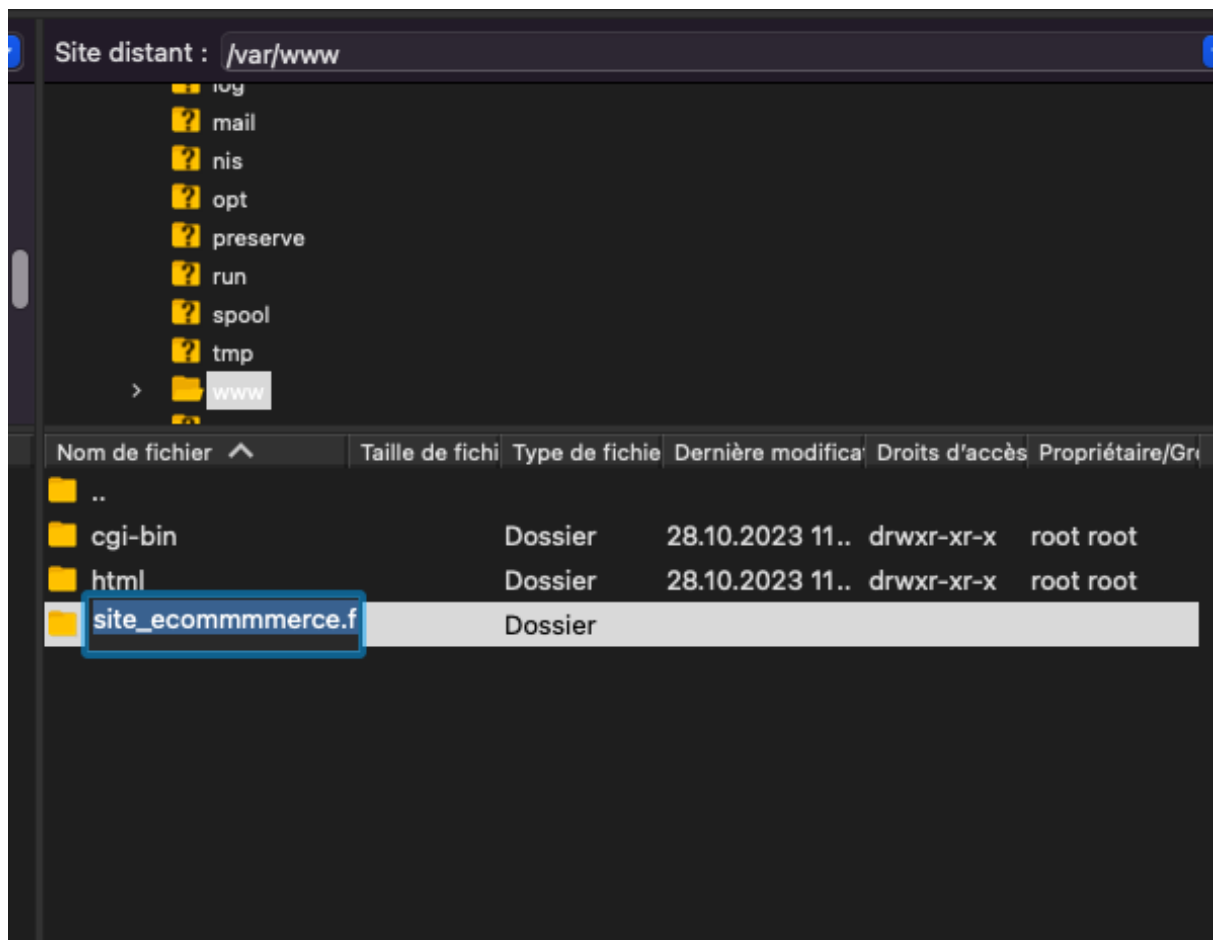
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
[MariaDB [ECOM]> source /var/www/site_ecommerce.fr/database.sql
Query OK, 0 rows affected (0.049 sec)

MariaDB [ECOM]> 
```

```
[MariaDB [ECOM]> exit;
Bye
[[root@WEB1 site_ecommerce.fr]# systemctl restart nginx
[[root@WEB1 site_ecommerce.fr]# systemctl enable nginx
[[root@WEB1 site_ecommerce.fr]# 
```





```
[root@WEB1 site_ecommmerce.fr]# mysql -u noa -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 10.5.22-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use ECOM
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

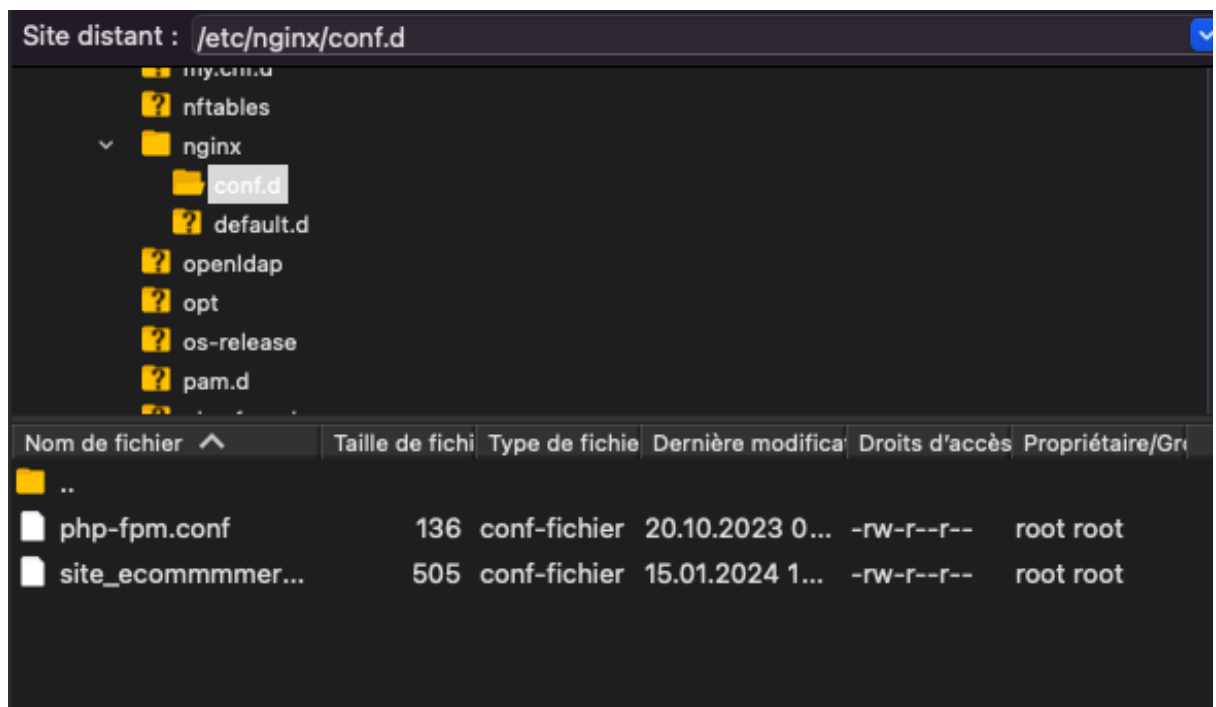
Database changed
MariaDB [ECOM]> source /var/www/site_ecommmerce.fr/database.sql
Query OK, 0 rows affected, 1 warning (0.001 sec)

MariaDB [ECOM]> EXIT;
Bye
[root@WEB1 site_ecommmerce.fr]#
```

```

[[root@WEB1 site_ecommmerce.fr]# systemctl restart mariadb.service
[[root@WEB1 site_ecommmerce.fr]# systemctl enable --now mariadb.service
[[root@WEB1 site_ecommmerce.fr]# systemctl restart php-fpm
[[root@WEB1 site_ecommmerce.fr]# systemctl enable php-fpm
[[root@WEB1 site_ecommmerce.fr]# systemctl restart php
Failed to restart php.service: Unit php.service not found.
[[root@WEB1 site_ecommmerce.fr]# systemctl restart nginx
[[root@WEB1 site_ecommmerce.fr]# systemctl enable nginx
[[root@WEB1 site_ecommmerce.fr]#

```



```

GNU nano 5.6.1 /etc/nginx/conf.d/site_ecommmerce.fr.conf Modified
server { listen 80;
listen [::]:80;
root /var/www/site_ecommmerce.fr/;
index index.html index.htm index.nginx-debian.html sign-up.php;
server_name ecommmerce.fr ; location ~* \.php$ {
fastcgi_pass unix:/run/php-fpm/www.sock;
include fastcgi_params;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
} fastcgi_param SCRIPT_NAME $fastcgi_script_name;
access_log /var/log/nginx/access_ecommerce.fr.log; error_log /var/log/nginx/err>
try_files $uri $uri/ =404; }}

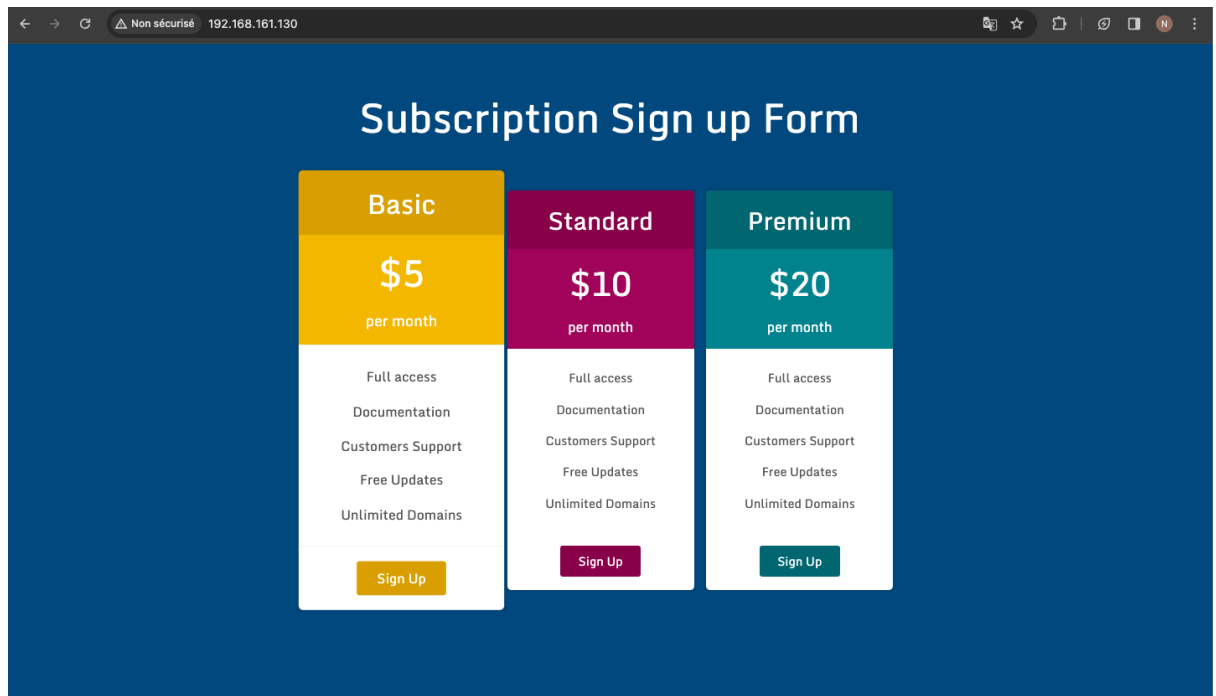
```

On restart les services comme indiquer dans la capture suivante et ça fonctionne :

```

2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:fd:31:14 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.161.130/24 brd 192.168.161.255 scope global dynamic noprefixroute ens160
        valid_lft 1774sec preferred_lft 1774sec
    inet6 fe80::20c:29ff:fefd:3114/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@WEB1 ~]# systemctl start nginx
[root@WEB1 ~]# systemctl enable nginx
[ 78.070370] systemd-rc-local-generator[1560]: /etc/rc.d/rc.local is not marked executable, skipping.
[root@WEB1 ~]# systemctl start mariadb.service
[root@WEB1 ~]# systemctl enable --now mariadb.service
[ 99.442182] systemd-rc-local-generator[1584]: /etc/rc.d/rc.local is not marked executable, skipping.
[root@WEB1 ~]# systemctl start php-fpm
[root@WEB1 ~]# systemctl enable php-fpm
[ 117.565217] systemd-rc-local-generator[1609]: /etc/rc.d/rc.local is not marked executable, skipping.

```



Nous allons maintenant ajouter dans le fichier host de notre ordinateur le site ecommerce.fr avec l'adresse ip du serveur web1. Voici comment faire :

```

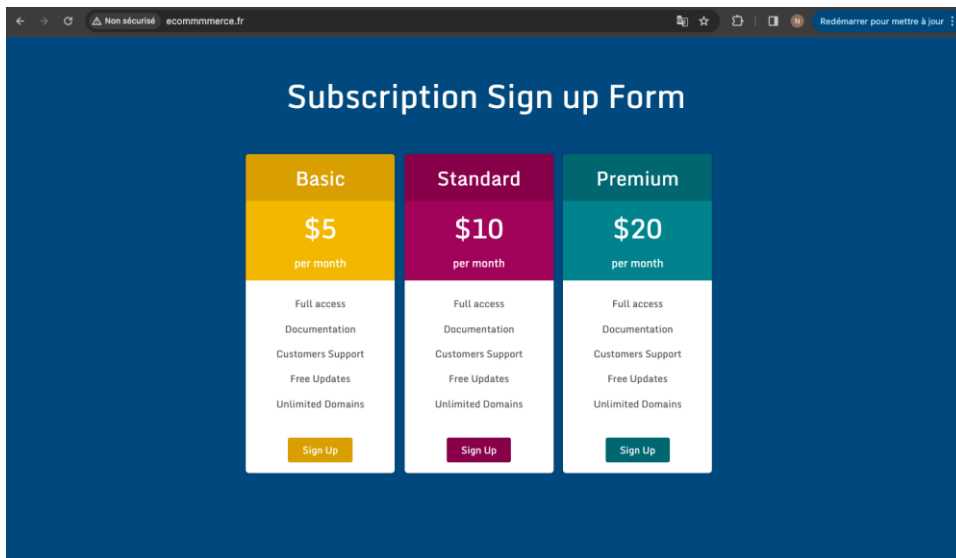
[Air-de-Noa:~ nowa$ nano /etc/hosts
[Air-de-Noa:~ nowa$ sudo nano /etc/hosts
[Password:

```

Ensuite il faut rajouter ceci dans le fichier de conf :

192.168.161.130 ecommerce.fr
------------------------------

Puis il faut sauvegarder et taper ecommerce.fr dans la barre de recherche et ça fonctionne.



2)

Maintenant nous allons faire pour le site UK.

Il faut suivre les étapes précédentes pour mettre le site en uk.

Voici déjà les logs présentes pour les sites :

```
[[root@WEB1 ~]# cd /var/log/nginx/
[[root@WEB1 nginx]# ls -lrt
total 72
-rw-r--r--. 1 root root 259 Jan 15 11:54 error.log-20240129
-rw-r--r--. 1 root root 40601 Jan 15 17:56 access.log-20240129
-rw-r--r--. 1 root root 587 Jan 16 18:48 error_ecommerce.fr.log-20240129
-rw-r--r--. 1 root root 4379 Jan 16 18:49 access_ecommerce.fr.log-20240129
-rw-r-----. 1 nginx root 0 Jan 29 08:21 access.log
-rw-r-----. 1 nginx root 0 Jan 29 08:21 error.log
-rw-r-----. 1 nginx root 5890 Jan 29 10:36 error_ecommerce.fr.log
-rw-r-----. 1 nginx root 7784 Jan 29 10:36 access_ecommerce.fr.log
[[root@WEB1 nginx]#
```

Maintenant il faut faire toutes les étapes montrées ci-dessous :

```
[[root@WEB1 ~]# cd /etc/nginx/conf.d
[[root@WEB1 conf.d]# nano site_ecommerce.uk.conf
```

Cd /etc/nginx/conf.d
Nano site_ecommerce.uk.conf
<pre>server { listen 80;  listen [::]:80;  root /var/www/site_commmmerce.uk/; index index.html index.htm index.nginx-debian.html sign-up.php;  server_name ecommerce.uk ; location ~* \.php\$ {</pre>

```

fastcgi_pass unix:/run/php-fpm/www.sock;
include fastcgi_params;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;

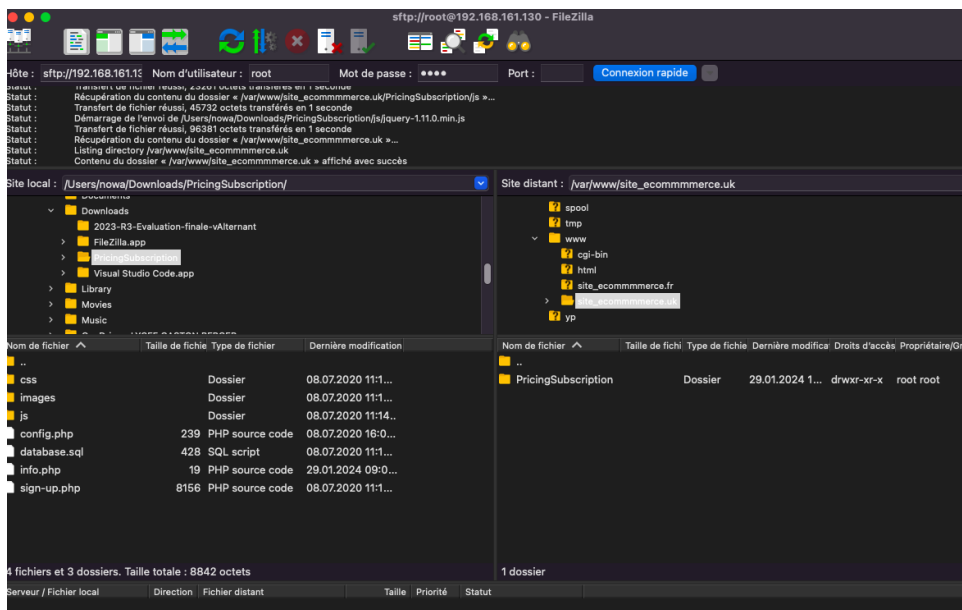
} fastcgi_param SCRIPT_NAME $fastcgi_script_name;

access_log /var/log/nginx/access_ecommerce.uk.log; error_log
/var/log/nginx/error_ecommerce.uk.log; location / {

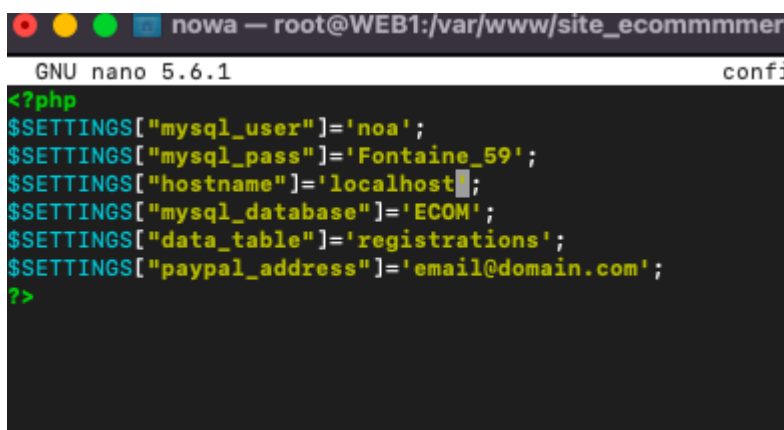
try_files $uri $uri/ =404; }}

```

On transfère le dossier du site dans un dossier créé site\_ecommerce.uk



On modifie le fichier config.php comme pour le site en .fr que nous avons fait :



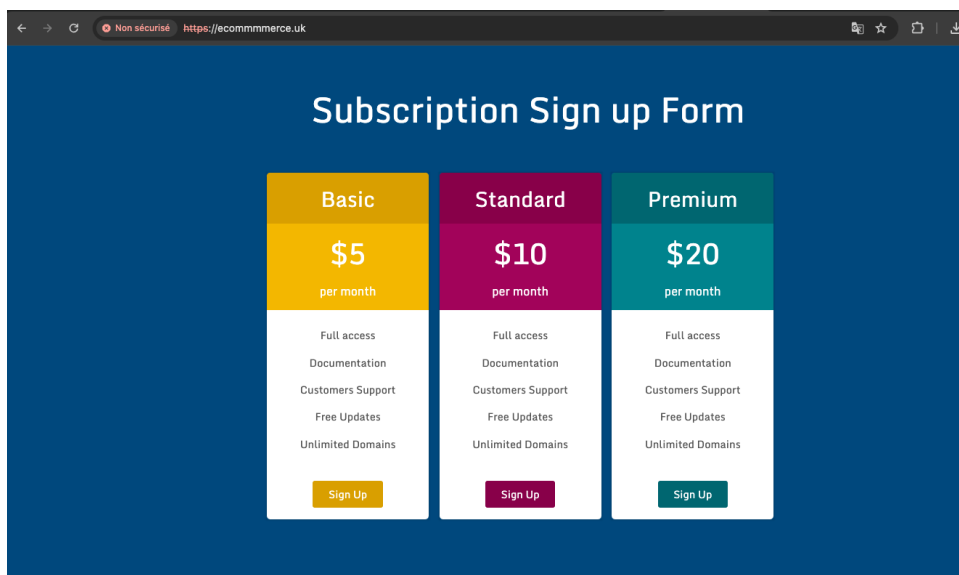
Faire ceci dans la bdd mysql :

```
[MariaDB [(none)]]> use ECOM
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
[MariaDB [ECOM]]> source /var/www/site_ecommerce.uk/database.sql
Query OK, 0 rows affected, 1 warning (0.004 sec)

MariaDB [ECOM]> █
```

```
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
##
127.0.0.1          localhost
255.255.255.255    broadcasthost
::1               localhost
192.168.161.130    ecommerce.fr
192.168.161.130    ecommerce.uk█
```



---

Intéressons nous au certificat SSL pour sécuriser nos sites.

On installe OpenSSL avec cette commande :

```
Dnf install -y openssl
```

```

root@GHEB1 ~# dnf install -y openssl
rocky Linux 9 - BaseOS                               3.6 kB/s | 4.1 kB | 00:01
rocky Linux 9 - AppStream                             5.3 kB/s | 4.5 kB | 00:00
rocky Linux 9 - Extras                                3.1 kB/s | 2.9 kB | 00:00
Package openssl-1:3.0.7-25.el9.x86_64 is already installed.
Dependencies resolved.
=====
Package                                Architecture           Version                Repository             Size
=====
Upgrading:
openssl                                x86_64                 1:3.0.7-25.el9_3      baseos                 1.2 M
openssl-libs                           x86_64                 1:3.0.7-25.el9_3      baseos                 2.1 M
=====
Transaction Summary
=====
Upgrade 2 Packages

Total download size: 3.3 M
Downloading Packages:
(1/2): openssl-libs-3.0.7-25.el9_3.x86_64.rpm        2.0 MB/s | 2.1 MB | 00:01
(2/2): openssl-3.0.7-25.el9_3.x86_64.rpm             1.1 MB/s | 1.2 MB | 00:01
-----
Total                                              2.3 MB/s | 3.3 MB | 00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Upgrading : openssl-libs-1:3.0.7-25.el9_3.x86_64          1/4
Upgrading : openssl-1:3.0.7-25.el9_3.x86_64              2/4
Cleanup   : openssl-1:3.0.7-24.el9.x86_64                3/4
Cleanup   : openssl-libs-1:3.0.7-24.el9.x86_64           4/4
Running scriptlet: openssl-libs-1:3.0.7-24.el9.x86_64     4/4
Verifying  : openssl-libs-1:3.0.7-25.el9_3.x86_64        1/4
Verifying  : openssl-libs-1:3.0.7-24.el9.x86_64          2/4
Verifying  : openssl-1:3.0.7-25.el9_3.x86_64             3/4
Verifying  : openssl-1:3.0.7-24.el9.x86_64               4/4
=====
Upgraded:
openssl-1:3.0.7-25.el9_3.x86_64                        openssl-libs-1:3.0.7-25.el9_3.x86_64

Complete!
root@GHEB1 ~# mkdir /etc/ssl/private
root@GHEB1 ~#

```

Puis on crée le répertoire privé dans `/etc/ssl/` avec cette commande :

```
Mkdir /etc/ssl/private
```

Puis nous allons limité les droits grâce à cette commande :

```
Chmod 700 /etc/ssl/private
```

Maintenant il faut taper cette commande pour créer le certificat auto-signée :

```
OpenSSL req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt
```

```
root@WEB1:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt
.....
.....
.....
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

Il faut remplir ensuite les questions demandées :

```

-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:FRANCE
String too long, must be at most 2 bytes long
Country Name (2 letter code) [XX]:fr
State or Province Name (full name) []:NORD
Locality Name (eg, city) [Default City]:LILLE
Organization Name (eg, company) [Default Company Ltd]:GASTONBERGER
Organizational Unit Name (eg, section) []:INFO
Common Name (eg, your name or your server's hostname) []:WEB1
Email Address []:noafontaine.pro@gmail.com
[root@WEB1 ~]#

```

Faisons cette commande pour créer les groupes DEFFIE-HELLMAN

```
[root@WEB1 ~]# openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
Generating DH parameters, 2048 bit long safe prime
```

Il faut modifier le fichier de conf dans /etc/nginx/conf.d/site\_ecommerce.fr.conf

```
GNU nano 5.6.1 /etc/nginx/conf.d/site_ecommerce.fr.conf
server {
    listen 443 ssl;
    listen [::]:443 ssl;
    ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
    ssl_dhparam /etc/ssl/certs/dhparam.pem;

    root /var/www/site_ecommerce.fr;

    index index.html index.htm index.nginx-debian.html sign-up.php;

    server_name ecommerce.fr;

    location ~* \.php$ {
        fastcgi_pass unix:/run/php-fpm.sock;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param SCRIPT_NAME $fastcgi_script_name;
    }

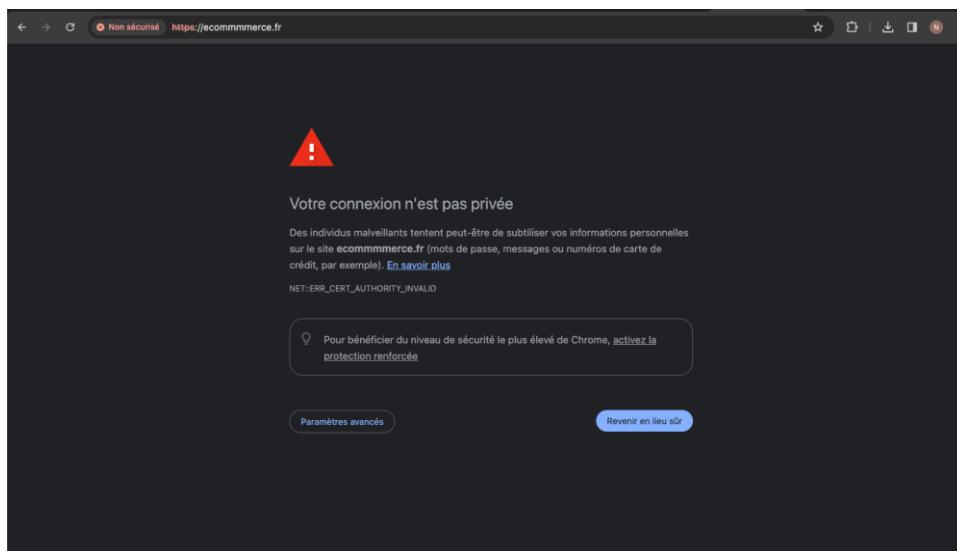
    access_log /var/log/nginx/access_ecommerce.fr.log;
    error_log /var/log/nginx/error_ecommerce.fr.log;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

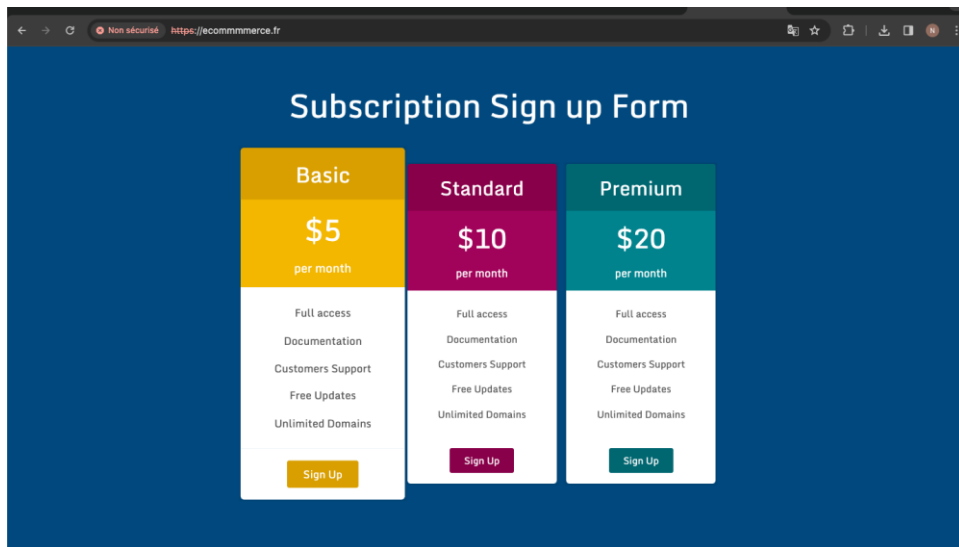
On ajoute le port https qui est 443 dans le firewall puis on relance les différents services :

```
[root@WEB1 ~]# firewall-cmd --add-port=443/tcp --permanent
success
[root@WEB1 ~]# firewall-cmd --reload
success
```

```
[root@WEB1 ~]# systemctl restart nginx mariadb php-fpm_
```

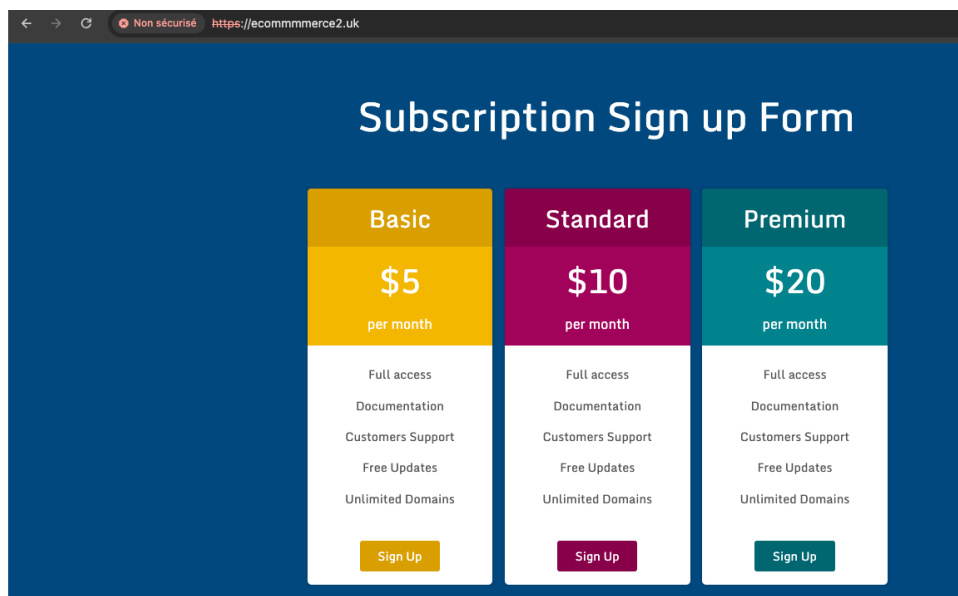


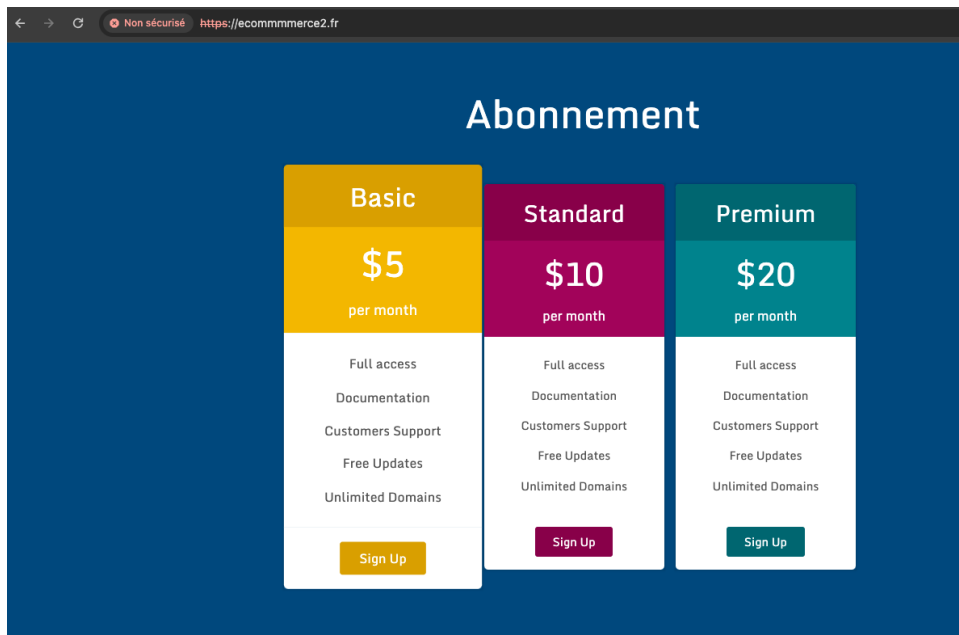




Nous allons voir la configuration du serveur HAPROXY mais tout d'abord j'ai dupliqué mon serveur WEB 1 et enlever dans la copie mariadb pour pouvoir se connecter sur la bdd du serveur web 1 depuis le serveur web 2.

On peut voir que j'ai changé la redirection pour les différenciés :





Maintenant dans WEB 1 j'ouvre le port 3306/tcp via le firewall :

```
[root@WEB1 ~]# firewall-cmd --add-port=3306/tcp --permanent
success
[root@WEB1 ~]# firewall-cmd --reload
success
[root@WEB1 ~]# _
```

Nous allons maintenant décommenter une ligne dans le fichier de conf de mariadb :

```
[root@WEB1 ~]# nano /etc/my.cnf.d/mariadb-server.cnf
```

Celle-ci :

Bind-address : 0.0.0.0
------------------------

```
[galera]
# Mandatory settings
#wsrep_on=ON
#wsrep_provider=
#wsrep_cluster_address=
#binlog_format=row
#default_storage_engine=InnoDB
#innodb_autoinc_lock_mode=2
#
# Allow server to accept connections on all interfaces.
#
bind-address=0.0.0.0
#
# Optional setting
#wsrep_slave_threads=1
#innodb_flush_log_at_trx_commit=0
# this is only for embedded server
[embedded]
```

Ensuite dans WEB 1, Il faut faire ceci pour que l'on puisse se connecter via web 2

```

MariaDB [(none)]> GRANT ALL on ECOM.* to user@'192.168.177.130' IDENTIFIED BY 'u
ser';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> █

```

Maintenant on peut se connecter à distance à la base de données :

```

[root@WEB2 ~]# mysql -u user -h 192.168.177.130 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.5.22-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database                |
+-----+
| ECOM                     |
| information_schema      |
+-----+
2 rows in set (0.304 sec)

MariaDB [(none)]> █

```

3)

On installe sur la machine haproxy haproxy puis on copie colle le fichier de conf dans le même répertoire pour avoir une sauvegarde :

```

[root@HAPROXY ~]# dnf install -y haproxy █

```

```

[root@HAPROXY ~]# cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy2.cfg █

```

Ensuite allez dans le fichier de configuration puis le modifié comme suit :

```

GNU nano 5.6.1 /etc/haproxy/haproxy.cfg
ssl-default-server-ciphers PROFILE=SYSTEM

#-----
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#-----
defaults
  mode tcp
  listen stats
  bind *:8080
  mode http
  stats enable
  stats hide-version
  stats uri /stats
  stats admin if LOCALHOST
  stats auth haproxy:haproxy

#-----
# main frontend which proxys to the backends
#-----
frontend main
  bind *:80
  acl url_static      path_beg       -i /static /images /javascript /stylesheets
  acl url_static      path_end       -i .jpg .gif .png .css .js
  # use_backend static      if url_static
  default_backend      app

#-----
# static backend for serving up images, stylesheets and such
#-----
backend static
  balance roundrobin
  server      static 127.0.0.1:4331 check

#-----
# round robin balancing between the various backends
#-----
backend app
  balance roundrobin
  server web01 192.168.161.130:443 ssl verify none check
  server web02 192.168.161.137:443 ssl verify none check

```

On active rsyslog :

```

[root@HAPROXY ~]# systemctl start rsyslog
[root@HAPROXY ~]# systemctl enable rsyslog
[ 1249.387945] systemd-rc-local-generator[14381]: /etc/rc.d/rc.local is not marked executable, skipping.
[root@HAPROXY ~]# systemctl status rsyslog
● rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-02-12 18:31:15 CET; 20min ago
     Docs: man:rsyslogd(8)
           https://www.rsyslog.com/doc/
   Main PID: 832 (rsyslogd)
     Tasks: 3 (limit: 10375)
    Memory: 6.1M
       CPU: 430ms
   CGroup: /system.slice/rsyslog.service
           └─832 /usr/sbin/rsyslogd -n

Feb 12 18:31:14 HAPROXY systemd[1]: Starting System Logging Service...
Feb 12 18:31:15 HAPROXY systemd[1]: Started System Logging Service.
Feb 12 18:31:15 HAPROXY rsyslogd[832]: [origin software="rsyslogd" swVersion="8.2102.0-101.e19_0.1" x-pid="832" x-info="https://www.rsyslog.com/doc"]
Feb 12 18:31:15 HAPROXY rsyslogd[832]: imjournal: journal files changed, reloading... [v8.2102.0-101.e19_0.1 try https://www.rsyslog.com/doc]
[root@HAPROXY ~]#

```

Ensuite il faut modifier le fichier de conf de rsyslog comme suit :

```

module(load="imudp") # needs to be done just once
input(type="imudp" port="514")

```

Il suffit juste de décommenter les deux lignes montrées ici ^

SELINUX veille sur tout et n'importe quoi il faut donc lui donner une policy. Voici comment faire :

```

[root@HAPROXY ~]# setsebool -P haproxy_connect_any 1

```

```

[root@HAPROXY ~]# setsebool -P haproxy_connect_any 1
[ 1457.762541] SELinux: Converting 330 SID table entries...
[ 1457.774511] SELinux: policy capability network_peer_controls=1
[ 1457.775479] SELinux: policy capability open_perms=1
[ 1457.776271] SELinux: policy capability extended_socket_class=1
[ 1457.777033] SELinux: policy capability always_check_network=0
[ 1457.777787] SELinux: policy capability cgroup_seclabel=1
[ 1457.778647] SELinux: policy capability mnp_nosuid_transition=1
[ 1457.779655] SELinux: policy capability genfs_seclabel_symlinks=0
[root@HAPROXY ~]#

```

Maintenant on redémarre haproxy et rsyslog :

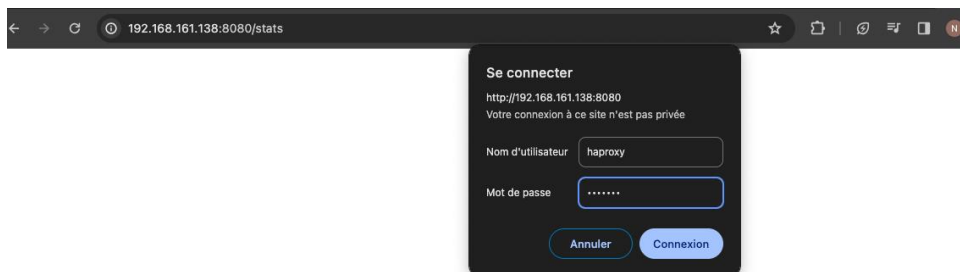
```
[root@HAPROXY ~]# systemctl restart rsyslog haproxy
```

On n'oublie pas bien sûr d'ouvrir le port 8080 via le firewall sinon ça fonctionne pas :

```
[root@HAPROXY ~]# firewall-cmd --permanent --add-port=8080/tcp
success
[root@HAPROXY ~]# firewall-cmd --reload
success
[root@HAPROXY ~]#
```

Voici le résultat :

On renseigne les infos de log rempli dans le fichier de conf de haproxy :



Et voici :

HAProxy  
Statistics Report for pid 1466

> General process information

pid = 1466 (process #1, rlimit = 1, nthread = 2)  
uptime = 5d 0h33m37s  
systemd.service.name = unitd.service = 8033  
maxsock = 65535, maxconn = 4092, maxqueue = 0  
current conn = 1, current pipe = 100, conn rate = 0/sec, bit rate = 5.166 kbps  
Running tasks: 0/16, cda = 100 %

active UP  
active UP going down  
active DOWN going up  
active or backup DOWN  
active or backup DOWN for maintenance (MAINT)  
active or backup SOFT STOPPED for maintenance  
Note: "NOLOG" or "NOLOG" = UP with load balancing disabled.

Display options:  
• Scope: [ ]  
• Hide DOWN servers  
• Refresh: [ ]  
• Color: [ ]  
• JSON: [ ]

External resources:  
• Proxy: [ ]  
• Unitd: [ ]  
• Online: [ ]

Global		Session rate		Sessions		Bytes		Errors		Warnings		Status		Server	
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Weight
Frontend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Backend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Global		Session rate		Sessions		Bytes		Errors		Warnings		Status		Server	
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Weight
Frontend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Backend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Global		Session rate		Sessions		Bytes		Errors		Warnings		Status		Server	
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Weight
Frontend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Backend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Global		Session rate		Sessions		Bytes		Errors		Warnings		Status		Server	
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Weight
Frontend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Backend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

C'est bien beau mais il faut le sécuriser maintenant. Voici les étapes à faire :

On se déplace dans le répertoire ci :

```
[root@HAPROXY ~]# cd /etc/pki/tls/certs
```

On installe openssl puis nous faisons la très longue commande pour obtenir un certificat auto-signé :

```

[root@haproxy ~]# rpm -q openssl
openssl-1:3.0.1-20.el9_3.x86_64
Last metadata expiration check: 8:21:16 ago on Mon Feb 12 10:45:24 2024.
Package openssl-1:3.0.1-20.el9_3.x86_64 is already installed.
Dependencies resolved.
=====
Package                Architecture      Version           Repository        Size
-----
Upgrading:
openssl                x86_64            1:3.0.7-25.el9_3  basos             1.2 M
openssl-libs            x86_64            1:3.0.7-25.el9_3  basos             2.1 M
=====
Transaction Summary
-----
Upgrade 2 Packages

Total download size: 3.3 M
Downloading Packages:
(1/2): openssl-libs-3.0.7-25.el9_3.x86_64.rpm 262 kB/s | 2.1 MB 00:00
(2/2): openssl-3.0.7-25.el9_3.x86_64.rpm       31 kB/s | 1.2 MB 00:30
-----
Total                                     87 kB/s | 3.3 MB 00:30
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                : 1/1
  Upgrading                : openssl-libs-1:3.0.7-25.el9_3.x86_64 1/4
  Upgrading                : openssl-1:3.0.7-25.el9_3.x86_64      2/4
  Cleanup                  : openssl-1:3.0.1-20.el9_3.x86_64      3/4
  Cleanup                  : openssl-libs-1:3.0.1-20.el9_3.x86_64 4/4
  Running scriptlet: openssl-libs-1:3.0.1-20.el9_3.x86_64      4/4
  Verifying                : openssl-libs-1:3.0.1-20.el9_3.x86_64 1/4
  Verifying                : openssl-libs-1:3.0.1-20.el9_3.x86_64 2/4
  Verifying                : openssl-1:3.0.7-25.el9_3.x86_64     3/4
  Verifying                : openssl-1:3.0.1-20.el9_3.x86_64     4/4

Upgraded:
openssl-1:3.0.7-25.el9_3.x86_64      openssl-libs-1:3.0.7-25.el9_3.x86_64

Complete!
[root@haproxy ~]# rpm -q openssl req --589 -nodes -newkey rsa:2048 -keyout /etc/pki/tls/certs/haproxy.pem -out /etc/pki/tls/certs/haproxy.pem -days 365

```

Ensuite nous devons re modifier le fichier de conf de haproxy. Voici comment :

Il faut rajouter les lignes dans le global maxsslconn 256 et tune.ssl.default-dh-param 2048 et dans le front changer le port du bind et rajouter le fichier du certificat ssl auto-signée :

```

GNU nano 5.6.1 /etc/haproxy/haproxy.cfg
user      haproxy
group     haproxy
daemon

maxsslconn 256
tune.ssl.default-dh-param 2048

# turn on stats unix socket
stats socket /var/lib/haproxy/stats

# utilize system-wide crypto-policies
ssl-default-bind-ciphers PROFILE=SYSTEM
ssl-default-server-ciphers PROFILE=SYSTEM

#-----
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#-----
defaults
  mode tcp
  listen stats
  bind *:8080
  mode http
  stats enable
  stats hide-version
  stats uri /stats -> l'url
  stats admin if LOCALHOST
  stats auth haproxy:haproxy

#-----
# main frontend which proxys to the backends
#-----
frontend main
  bind *:443 ssl crt /etc/pki/tls/certs/haproxy.pem

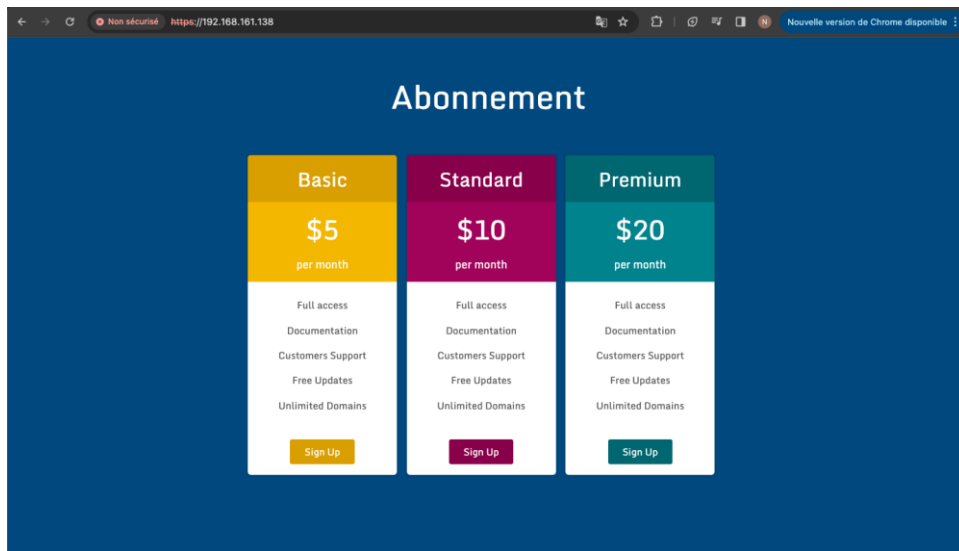
  acl url_static      path_beg       -i /static /images /javascript /stylesheets
  acl url_static      path_end       -i .jpg .gif .png .css .js

  use_backend static   if url_static
  default_backend      app

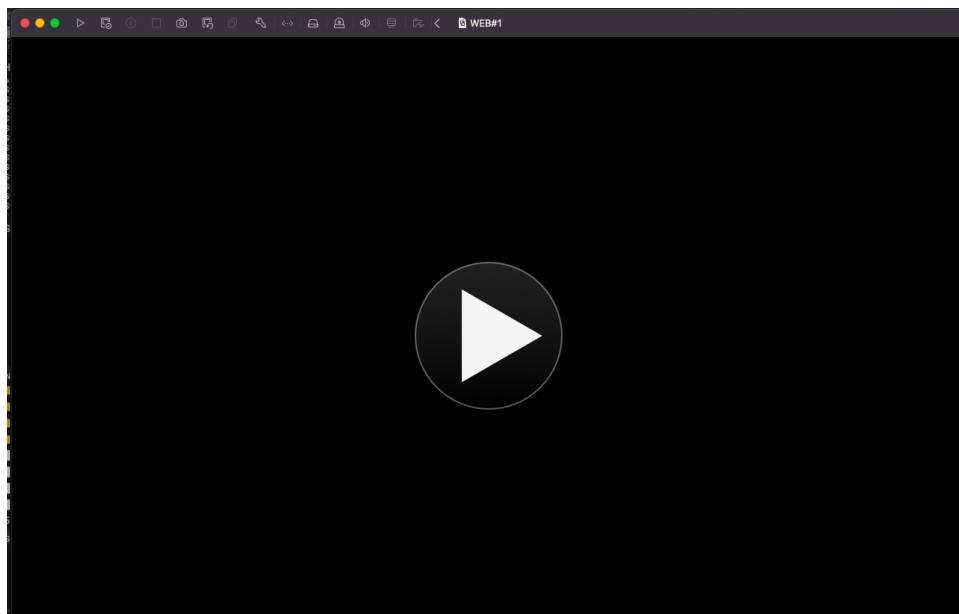
#-----
# static backend for serving up images, stylesheets and such
#-----
backend static

```

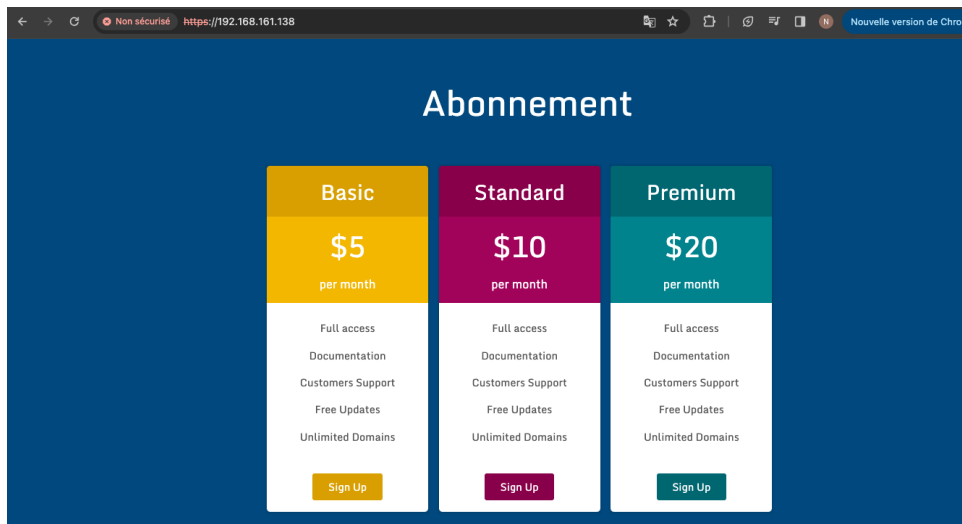
On redémarre le service haproxy puis on fait le test :



Si on était le serveur WEB 1



Ça fonctionne toujours :



4)

J'ai cloné mon premier HAPROXY pour en avoir deux.

Je modifie le fichier hosts de toutes mes VMs pour qu'ils puissent résoudre les noms des autres serveurs

HAPROXY #2 :

```
GNU nano 5.6.1 /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.161.138 HAPROXY#1
192.168.161.137 WEB2
192.168.161.130 WEB1
```

HAPROXY #1 :

```
GNU nano 5.6.1 /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.161.141 HAPROXY#2
192.168.161.130 WEB1
192.168.161.137 WEB2
```

WEB1 :

```
GNU nano 5.6.1 /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.161.141 HAPROXY#2
192.168.161.138 HAPROXY#1
192.168.161.137 WEB2
```

WEB2 :



```
GNU nano 5.6.1 /etc
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.161.141 HAPROXY#2
192.168.161.138 HAPROXY#1
192.168.161.130 WEB1
```

Je fais un curl pour le test sur les deux haproxy :

HAPROXY#2 :

```
[root@HAPROXY2 ~]# curl WEB1
<!--The Rocky Linux Project nor the <strong>RESF</strong> have
"hacked" this webserver: This test page is included with the
distribution.</li>
<ul>
<p>For more information about Rocky Linux, please visit the
<a href="https://rockylinux.org/"><strong>Rocky Linux
website</strong></a>.
</p>
</div>
<div class='col-sm-12 col-md-6 col-md-6 col-md-offset-12'>
<div class='section'>

<h2>I am the admin, what do I do?</h2>

<p>You may now add content to the webroot directory for your
software.</p>

<p><strong>For systems using the
<a href="https://httpd.apache.org/">Apache Webserver</strong></a>:
You can add content to the directory <code>/var/www/html/</code>.
Until you do so, people visiting your website will see this page. If
you would like this page to not be shown, follow the instructions in:
<code>/etc/httpd/conf.d/welcome.conf</code>.</p>

<p><strong>For systems using
<a href="https://nginx.org">Nginx</strong></a>:
You can add your content in a location of your
choice and edit the <code>root</code> configuration directive
in <code>/etc/nginx/nginx.conf</code>.</p>

<div id="logos">
<a href="https://rockylinux.org/" id="rocky-poweredby"></a> <!-- Rocky -->
 <!-- webserver -->
</div>
</div>
</div>

<footer class="col-sm-12">
<a href="https://apache.org">Apache</a> is a registered trademark of <a href="https://apache.org">the Apache Sof
e Foundation</a> in the United States and/or other countries.<br />
<a href="https://nginx.org">NGINX</a> is a registered trademark of <a href="https://">F5 Networks, Inc.</a>.
</Footer>

</body>
</html>
[root@HAPROXY2 ~]# curl WEB2_
```

```

<li>The Rocky Linux Project nor the <strong>RESF</strong> have
"hacked" this webserver: This test page is included with the
distribution.</li>
</ul>
<p>For more information about Rocky Linux, please visit the
<a href="https://rockylinux.org/"><strong>Rocky Linux
website</strong></a>.
</p>
</div>
<div class='col-sm-12 col-md-6 col-md-6 col-md-offset-12'>
<div class='section'>

    <h2>I am the admin, what do I do?</h2>

<p>You may now add content to the webroot directory for your
software.</p>

<p><strong>For systems using the
<a href="https://httpd.apache.org/">Apache Webserver</strong></a>:
You can add content to the directory <code>/var/www/html</code>.
Until you do so, people visiting your website will see this page. If
you would like this page to not be shown, follow the instructions in:
<code>/etc/httpd/conf.d/welcome.conf</code>.</p>

<p><strong>For systems using
<a href="https://nginx.org">Nginx</strong></a>:
You can add your content in a location of your
choice and edit the <code>root</code> configuration directive
in <code>/etc/nginx/nginx.conf</code>.</p>

<div id="logos">
    <a href="https://rockylinux.org/" id="rocky-poweredby"></a> <!-- Rocky -->
     <!-- webserver -->
</div>
</div>
</div>

<footer class="col-sm-12">
<a href="https://apache.org">Apache&trade;</a> is a registered trademark of <a href="https://apache.org">the Apache So
e Foundation</a> in the United States and/or other countries.<br />
<a href="https://nginx.org">NGINX&trade;</a> is a registered trademark of <a href="https://">F5 Networks, Inc.</a>.
</footer>

</body>
</html>
root@HAPROXY2 ~]#

```

## HAPROXY#1:

```
[root@HAPROXY ~]# curl WEB1
```

```

<li>The Rocky Linux Project nor the <strong>RESF</strong> have
"hacked" this webserver: This test page is included with the
distribution.</li>
</ul>
<p>For more information about Rocky Linux, please visit the
<a href="https://rockylinux.org/"><strong>Rocky Linux
website</strong></a>.
</p>
</div>
<div class='col-sm-12 col-md-6 col-md-6 col-md-offset-12'>
<div class='section'>

    <h2>I am the admin, what do I do?</h2>

<p>You may now add content to the webroot directory for your
software.</p>

<p><strong>For systems using the
<a href="https://httpd.apache.org/">Apache Webserver</strong></a>:
You can add content to the directory <code>/var/www/html</code>.
Until you do so, people visiting your website will see this page. If
you would like this page to not be shown, follow the instructions in:
<code>/etc/httpd/conf.d/welcome.conf</code>.</p>

<p><strong>For systems using
<a href="https://nginx.org">Nginx</strong></a>:
You can add your content in a location of your
choice and edit the <code>root</code> configuration directive
in <code>/etc/nginx/nginx.conf</code>.</p>

<div id="logos">
    <a href="https://rockylinux.org/" id="rocky-poweredby"></a> <!-- Rocky -->
     <!-- webserver -->
</div>
</div>
</div>

<footer class="col-sm-12">
<a href="https://apache.org">Apache&trade;</a> is a registered trademark of <a href="https://apac
e Foundation</a> in the United States and/or other countries.<br />
<a href="https://nginx.org">NGINX&trade;</a> is a registered trademark of <a href="https://">F5 N
</footer>

</body>
</html>
root@HAPROXY ~]#

```

```
[root@HAPROXY ~]# curl WEB2
```

```

<li>The Rocky Linux Project nor the <strong>RESF</strong> have
"hacked" this webserver: This test page is included with the
distribution.</li>
</ul>
<p>For more information about Rocky Linux, please visit the
<a href="https://rockylinux.org/"><strong>Rocky Linux
website</strong></a>.
</p>
</div>
</div>
<div class='col-sm-12 col-md-6 col-md-6 col-md-offset-12'>
<div class='section'>

<h2>I am the admin, what do I do?</h2>

<p>You may now add content to the webroot directory for your
software.</p>

<p><strong>For systems using the
<a href="https://httpd.apache.org/">Apache Webserver</strong></a>:
You can add content to the directory <code>/var/www/html/</code>.
Until you do so, people visiting your website will see this page. If
you would like this page to not be shown, follow the instructions in:
<code>/etc/httpd/conf.d/welcome.conf</code>.</p>

<p><strong>For systems using
<a href="https://nginx.org">Nginx</strong></a>:
You can add your content in a location of your
choice and edit the <code>root</code> configuration directive
in <code>/etc/nginx/nginx.conf</code>.</p>

<div id="logos">
<a href="https://rockylinux.org/" id="rocky-poweredby"></a> <!-- Rocky -->
 <!-- webserver -->
</div>
</div>
</div>

<footer class="col-sm-12">
<a href="https://apache.org">Apache</a> is a registered trademark of <a href="https://a
Foundation/> in the United States and/or other countries.<br />
<a href="https://nginx.org">NGINX</a> is a registered trademark of <a href="https://>F
</footer>

</body>
</html>
root@HAPROXY ~]#

```

Faisons l'installation du pacemaker :

Avec cette commande :

```

[root@HAPROXY2 ~]# dnf --enablerepo=highavailability -y install pacemaker pcs
Rocky Linux 9 - High Availability
169

```

Complete!

```

[root@HAPROXY ~]# dnf --enablerepo=highavailability -y install pacemaker pcs

```

On l'active au démarrage :

```

[root@HAPROXY2 ~]# systemctl enable --now pcsd
Created symlink /etc/systemd/system/multi-user.target.wants/pcsd.service + /usr/lib/systemd/system/pcsd.service.
[ 3048.699256] systemd-rc-local-generator[4061]: /etc/rc.d/rc.local is not marked executable, skipping.

```

```

[root@HAPROXY ~]# systemctl enable --now pcsd

```

Maintenant on crée le cluster

```

[root@HAPROXY ~]# passwd hacluster
Changing password for user hacluster.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@HAPROXY ~]#

```

On réalise l'ouverture dans le firewall :

```
[root@HAPROXY ~]# firewall-cmd --add-service=high-availability --permanent
success
[root@HAPROXY ~]# firewall-cmd --reload
success
[root@HAPROXY ~]#
```

Il faut aussi ajouter les services http et https pour que ça fonctionne !!!

Maintenant on fait la commande ci pour l'authentification des nodes et le setup :

```
[root@HAPROXY ~]# pcs host auth 192.168.161.138 192.168.161.141
Username: hacluster
Password:
192.168.161.138: Authorized
192.168.161.141: Authorized
[root@HAPROXY ~]# pcs cluster setup ha_cluster 192.168.161.138 192.168.161.141
No addresses specified for host '192.168.161.138', using '192.168.161.138'
No addresses specified for host '192.168.161.141', using '192.168.161.141'
Destroying cluster on hosts: '192.168.161.138', '192.168.161.141'...
192.168.161.138: Successfully destroyed cluster
192.168.161.141: Successfully destroyed cluster
Requesting remove 'pcsd settings' from '192.168.161.138', '192.168.161.141'
192.168.161.138: successful removal of the file 'pcsd settings'
192.168.161.141: successful removal of the file 'pcsd settings'
Sending 'corosync authkey', 'pacemaker authkey' to '192.168.161.138', '192.168.161.141'
192.168.161.141: successful distribution of the file 'corosync authkey'
192.168.161.141: successful distribution of the file 'pacemaker authkey'
192.168.161.138: successful distribution of the file 'corosync authkey'
192.168.161.138: successful distribution of the file 'pacemaker authkey'
Sending 'corosync.conf' to '192.168.161.138', '192.168.161.141'
192.168.161.138: successful distribution of the file 'corosync.conf'
192.168.161.141: successful distribution of the file 'corosync.conf'
Cluster has been successfully set up.
```

On start les clusters :

```
[root@HAPROXY ~]# pcs cluster start --all
192.168.161.138: Starting Cluster...
192.168.161.141: Starting Cluster...
[root@HAPROXY ~]#
```

On les rend actif :

```
[root@HAPROXY ~]# pcs cluster enable --all
[ 1233.828594] systemd-rc-local-generator[1978]: /etc/rc.d/rc.local is not marked executable, skipping.
[ 1234.250652] systemd-rc-local-generator[2003]: /etc/rc.d/rc.local is not marked executable, skipping.
192.168.161.138: Cluster Enabled
192.168.161.141: Cluster Enabled
[root@HAPROXY ~]#
```

On regarde les statuts et tout va bien :

```
[root@HAPROXY ~]# pcs cluster status
Cluster Status:
Cluster Summary:
* Stack: corosync (Pacemaker is running)
* Current DC: 192.168.161.141 (version 2.1.6-10.1.el9_3-6fdc9deea29) - partition with quorum
* Last updated: Thu Mar 14 15:11:27 2024 on 192.168.161.138
* Last change: Thu Mar 14 15:10:44 2024 by hacluster via crmd on 192.168.161.141
* 2 nodes configured
* 0 resource instances configured
Node List:
* Online: [ 192.168.161.138 192.168.161.141 ]

PCSD Status:
192.168.161.141: Online
192.168.161.138: Online
[root@HAPROXY ~]#
```

Voici des commandes qui peuvent être utile :

```

[root@HAPROXY ~]# pcs cluster corosync
totem {
    version: 2
    cluster_name: ha_cluster
    transport: knet
    crypto_cipher: aes256
    crypto_hash: sha256
    cluster_uuid: 14d682447a9940d9ab451c326667f827
}

nodelist {
    node {
        ring0_addr: 192.168.161.138
        name: 192.168.161.138
        nodeid: 1
    }

    node {
        ring0_addr: 192.168.161.141
        name: 192.168.161.141
        nodeid: 2
    }
}

quorum {
    provider: corosync_votequorum
    two_node: 1
}

logging {
    to_logfile: yes
    logfile: /var/log/cluster/corosync.log
    to_syslog: yes
    timestamp: on
}
[root@HAPROXY ~]#

```

Statut du cluster :

```

[root@HAPROXY ~]# pcs status cluster
Cluster Status:
Cluster Summary:
* Stack: corosync (Pacemaker is running)
* Current DC: 192.168.161.141 (version 2.1.6-10.1.el9_3-6fdc9deea29) - partition with quorum
* Last updated: Thu Mar 14 15:12:23 2024 on 192.168.161.138
* Last change: Thu Mar 14 15:10:44 2024 by hacluster via crmd on 192.168.161.141
* 2 nodes configured
* 0 resource instances configured
Node List:
* Online: [ 192.168.161.138 192.168.161.141 ]

PCSD Status:
192.168.161.138: Online
192.168.161.141: Online
[root@HAPROXY ~]#

```

Statut du node :

```
[root@HAPROXY ~]# pcs status node

Usage: pcs status ...
    [status] [--full] [--hide-inactive]
        View all information about the cluster and resources (--full provides
        more details, --hide-inactive hides inactive resources).

    resources [<resource id | tag id>] [node=<node>] [--hide-inactive]
        Show status of all currently configured resources. If --hide-inactive
        is specified, only show active resources. If a resource or tag id is
        specified, only show status of the specified resource or resources in
        the specified tag. If node is specified, only show status of resources
        configured for the specified node.

    cluster
        View current cluster status.

    corosync
        View current membership information as seen by corosync.

    quorum
        View current quorum status.

    qdevice <device model> [--full] [<cluster name>]
        Show runtime status of specified model of quorum device provider. Using
        --full will give more detailed output. If <cluster name> is specified,
        only information about the specified cluster will be displayed.

    booth
        Print current status of booth on the local node.

    nodes [corosync | both | config]
        View current status of nodes from pacemaker. If 'corosync' is
        specified, view current status of nodes from corosync instead. If
        'both' is specified, view current status of nodes from both corosync &
        pacemaker. If 'config' is specified, print nodes from corosync &
        pacemaker configuration.

    pcsd [<node>]...
        Show current status of pcsd on nodes specified, or on all nodes
        configured in the local cluster if no nodes are specified.

    xml
        View xml version of status (output from crm_mon -r -1 -X).

[root@HAPROXY ~]#
```

Permet de vérifier s'il y a des erreurs

```
[root@HAPROXY ~]# crm_verify -L -U
(unpack_resources) error: Resource start-up disabled since no STONITH resources have been defined
(unpack_resources) error: Either configure some or disable STONITH with the stonith-enabled option
(unpack_resources) error: NOTE: Clusters with shared data need STONITH to ensure data integrity
crm_verify: Errors found during check: config not valid
[root@HAPROXY ~]#
```

Pour corriger les erreurs :

```
[root@HAPROXY ~]# pcs property set stonith-enabled=false
[root@HAPROXY ~]# pcs property set no-quorum-policy=ignore
[root@HAPROXY ~]#
```

Pu rien :

```
[root@HAPROXY ~]# pcs property set stonith-enabled=false
[root@HAPROXY ~]# crm_verify -L -U
[root@HAPROXY ~]#
```

Cette commande permettra de voir les propriétés :

```
[root@HAPROXY ~]# pcs property
Cluster Properties: cib-bootstrap-options
  cluster-infrastructure=corosync
  cluster-name=ha_cluster
  dc-version=2.1.6-10.1.el9_3-6fdc9deea29
  have-watchdog=false
  no-quorum-policy=ignore
  stonith-enabled=false
[root@HAPROXY ~]#
```

On voit qu'on a des ip différentes grâce à la commande-ci :

```
[root@HAPROXY ~]# pcs resource create virtual_ip2 ocf:heartbeat:IPaddr2 ip=192.168.161.143 cidr_netmask=24 op monitor interval=30s
[root@HAPROXY ~]# pcs status resources
* virtual_ip (ocf:heartbeat:IPaddr2):          Started 192.168.161.141
* virtual_ip2 (ocf:heartbeat:IPaddr2):         Started 192.168.161.138
[root@HAPROXY ~]# ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:e1:55:52 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.161.138/24 brd 192.168.161.255 scope global dynamic noprefixroute ens160
        valid_lft 1039sec preferred_lft 1039sec
    inet 192.168.161.143/24 brd 192.168.161.255 scope global secondary ens160
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe01:5552/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@HAPROXY ~]#
```

Maintenant nous allons créer les clusters de ressource :

```
[root@HAPROXY ~]# pcs resource create haproxy ocf:heartbeat:haproxy binpath=/usr/sbin/haproxy conffile=/etc/haproxy/haproxy.cfg op monitor interval=10s --force
```

Ici, on voit les agents grâce à la commande montrer dans le screen :

```
[root@HAPROXY ~]# pcs resource agents ocf:heartbeat
apache
conntrackd
corosync-qnetd
crypt
CTDB
db2
Delay
dhcpcd
Dummy
ethmonitor
exportfs
Filesystem
galera
garbd
iface-vlan
IPaddr2
IPsrcaddr
iSCSILogicalUnit
iSCSITarget
LVM-activate
lvmlockd
MailTo
mysql
nagios
named
nfsnotify
nfsserver
nginx
NodeUtilization
openstack-cinder-volume
openstack-floating-ip
openstack-info
openstack-virtual-ip
oraasm
oracle
oralsnr
pgsql
podman
portblock
postfix
rabbitmq-cluster
redis
Route
rsyncd
SendArp
slapd
Squid
storage-mon
sybaseASE
symlink
tomcat
vdo-vol
VirtualDomain
Xinetd
[root@HAPROXY ~]#
```

On affecte ici la ressource au cluster :

```
[root@HAPROXY ~]# pcs resource group add HAProxyGroup virtual_ip
haproxy
[root@HAPROXY ~]#
```

On ajoute des contraintes mais je n'ai pas réussi :

```
[root@HAPROXY ~]# pcs constraint order virtual_ip then haproxy
Error: Cannot create an order constraint for resources in the same group
[root@HAPROXY ~]#
```

Ici on voit les informations du cluster :



```

[root@HAPROXY ~]# pcs status
Cluster name: ha_cluster
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: 192.168.161.141 (version 2.1.6-10.1.el9_3-6fdc9de
ea29) - partition with quorum
  * Last updated: Thu Mar 14 15:56:04 2024 on 192.168.161.138
  * Last change: Thu Mar 14 15:54:30 2024 by root via cibadmin o
n 192.168.161.138
  * 2 nodes configured
  * 7 resource instances configured

Node List:
  * Online: [ 192.168.161.138 192.168.161.141 ]

Full List of Resources:
  * virtual_ip2 (ocf:heartbeat:IPaddr2):      Started 192.168.
161.138
  * haproxy3 (ocf:heartbeat:haproxy):         Stopped (not ins
talled)
  * haproxy12 (ocf:heartbeat:haproxy):        Stopped (not ins
talled)
  * haproxy5 (ocf:heartbeat:haproxy5):        Stopped (not ins
talled)
  * haproxy6 (ocf:heartbeat:haproxy6):        Stopped (not ins
talled)
  * Resource Group: HAproxyGroup:
    * virtual_ip (ocf:heartbeat:IPaddr2):      Started
192.168.161.141
    * haproxy (ocf:heartbeat:haproxy):        Stopped (not ins
talled)

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
[root@HAPROXY ~]#

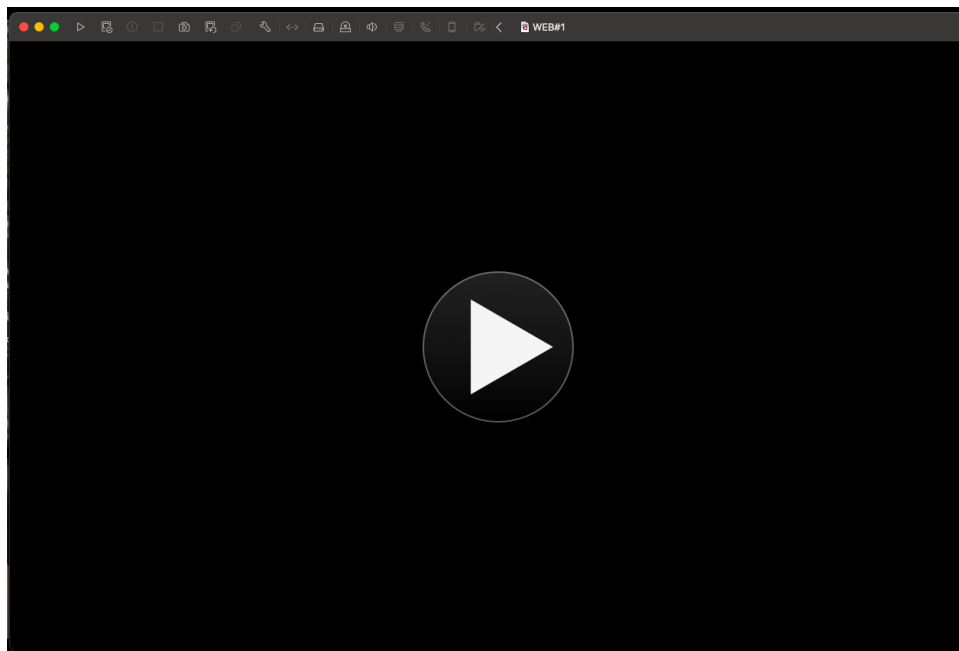
```

Voici le test :

The screenshot shows a terminal window on the left and a web browser on the right. The terminal window displays the output of the 'pcs status' command, showing the cluster name 'ha\_cluster', the stack 'corosync', and the current DC '192.168.161.141'. It also lists the nodes '192.168.161.138' and '192.168.161.141' as online. The full list of resources shows 'virtual\_ip2' started on '192.168.161.138' and 'haproxy3', 'haproxy12', 'haproxy5', and 'haproxy6' stopped. The daemon status shows 'corosync', 'pacemaker', and 'pcsd' as active/enabled.

The web browser displays a pricing page titled 'Abonnement' with three plans: Basic (\$5/month), Standard (\$10/month), and Premium (\$20/month). Each plan includes 'Full access', 'Documentation', 'Customers Support', 'Free Updates', and 'Unlimited Domains'. The 'Sign Up' button is visible for each plan.

Avec VM éteinte :



```
[root@HAPROXY ~]# pcs constraint order virtual_ip then haproxy
Error: Cannot create an order constraint for resources in the same group
[root@HAPROXY ~]# pcs status
Cluster name: ha-cluster
Cluster Summary:
  * Stack: corosync (pacemaker is running)
  * Current DC: 192.168.161.141 (version 2.1.4-10.1.el9_3-6f6c9de
ea29) - partition with quorum
  * Last updated: Thu Mar 14 15:54:0A 2024 on 192.168.161.138
  * Last change: Thu Mar 14 15:54:30 2024 by root via cibadmin o
n 192.168.161.138
  * 2 nodes configured
  * 7 resource instances configured

Node List:
  * Online: [ 192.168.161.138 192.168.161.141 ]

Full list of Resources:
  * virtual_ip2 (ocf:heartbeat:IPaddr2): Started 192.168.
161.138
  * haproxy3 (ocf:heartbeat:haproxy): Stopped (not ins
talled)
  * haproxy12 (ocf:heartbeat:haproxy): Stopped (not ins
talled)
  * haproxy6 (ocf:heartbeat:haproxy6): Stopped (not ins
talled)
  * haproxy6 (ocf:heartbeat:haproxy6): Stopped (not ins
talled)
  * Resource Group: HaproxyGroup:
    * virtual_ip (ocf:heartbeat:IPaddr2): Started
192.168.161.141
    * haproxy (ocf:heartbeat:haproxy): Stopped (not ins
talled)

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
quorum: active/enabled
[root@HAPROXY ~]# ip -o -s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    group default qlen 1000
    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: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq st
ate UP group default qlen 1000
    link/ether 00:0c:29:a1:58:52 brd ffff:ffff:ffff:ffff
    altname ens160
    inet 192.168.161.120/24 brd 192.168.161.255 scope global dyna
mic noprefixroute ens160
        valid_lft 3461sec preferred_lft 3461sec
    inet 192.168.161.142/24 brd 192.168.161.255 scope global seco
ndary ens160
        valid_lft forever preferred_lft forever
    inet6 fe80::c0c:29:a1:58:52/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@HAPROXY ~]#
```

## Abonnement

Basic	Standard	Premium
\$5	\$10	\$20
per month	per month	per month
Full access	Full access	Full access
Documentation	Documentation	Documentation
Customers Support	Customers Support	Customers Support
Free Updates	Free Updates	Free Updates
Unlimited Domains	Unlimited Domains	Unlimited Domains
Sign Up	Sign Up	Sign Up