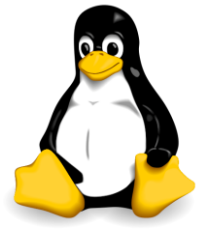
LESPAGNOL
Alexandre
2SISR

09/11/2023



Compte rendu (Mise en place d'un service WEB)

LESPAGNOL 09/11/2023
Alexandre
2SISR

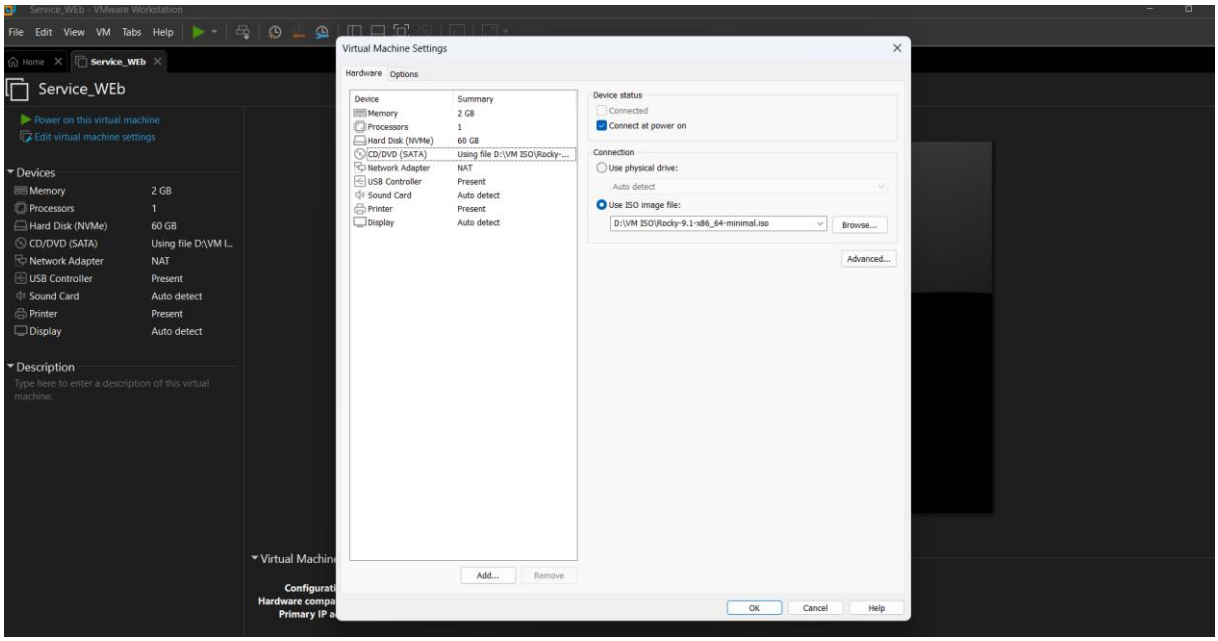
Sommaire :

LESPAGNOL
Alexandre
2SISR

09/11/2023

Installation machine service web

Choisir red hat 8 dans la version du linux



Installation d'un service web (NGINX)

Installation de nano :

```
localhost login: root
Password:
[root@localhost ~]# dnf install nano -y
Rocky Linux 9 - BaseOS                                          3.5 MB/s | 1.9 MB     00:00
Rocky Linux 9 - AppStream                                       9.5 MB/s | 7.1 MB     00:00
Rocky Linux 9 - Extras                                           35 kB/s |  11 kB     00:00
Dependencies resolved.
================================================================================
 Package              Architecture          Version                Repository      Size
================================================================================
Installing:
 nano                 x86_64                5.6.1-5.el9            baseos         694 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 694 k
Installed size: 2.7 M
Downloading Packages:
nano-5.6.1-5.el9.x86_64.rpm                                     3.3 MB/s | 694 kB     00:00
--------------------------------------------------------------------------------
Total                                                           1.5 MB/s | 694 kB     00:00
Rocky Linux 9 - BaseOS                                          1.7 MB/s | 1.7 kB     00:00
Importing GPG key 0x350D275D:
 Userid     : "Rocky Enterprise Software Foundation - Release key 2022 <releng@rockylinux.org>"
 Fingerprint: 21CB 256A E16F C54C 6E65 2949 702D 426D 350D 275D
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-Rocky-9
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : nano-5.6.1-5.el9.x86_64                               1/1
  Running scriptlet: nano-5.6.1-5.el9.x86_64                               1/1
  Verifying        : nano-5.6.1-5.el9.x86_64                               1/1

Installed:
  nano-5.6.1-5.el9.x86_64

Complete!
[root@localhost ~]#
```

Permettre les connexions root par ssh :

```
[root@localhost ~]# nano /etc/ssh/sshd_config_
```

```
 GNU nano 5.6.1                  /etc/ssh/sshd_config                    Modified

#LogLevel INFO
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes

^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo    M-A Set Mark   M-] To Bracket  M-Q Previous
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^_ Go To Line M-E Redo    M-6 Copy       ^Q Where Was    M-W Next
```

Installation package nginx :

```
[root@localhost ~]# dnf install nginx -y
Last metadata expiration check: 0:05:44 ago on Thu Nov  9 08:45:35 2023.
Dependencies resolved.
================================================================================
 Package                 Architecture  Version                     Repository        Size
================================================================================
Installing:
 nginx                   x86_64        1:1.20.1-14.el9_2.1         appstream         36 k
Installing dependencies:
 nginx-core              x86_64        1:1.20.1-14.el9_2.1         appstream        565 k
 nginx-filesystem        noarch        1:1.20.1-14.el9_2.1         appstream        8.5 k
 rocky-logos-httpd       noarch        90.14-1.el9                 appstream         24 k

Transaction Summary
================================================================================
Install  4 Packages

Total download size: 634 k
Installed size: 1.8 M
Downloading Packages:
(1/4): nginx-filesystem-1.20.1-14.el9_2.1.noarch.rpm          97 kB/s | 8.5 kB     00:00
(2/4): rocky-logos-httpd-90.14-1.el9.noarch.rpm              221 kB/s |  24 kB     00:00
(3/4): nginx-1.20.1-14.el9_2.1.x86_64.rpm                    314 kB/s |  36 kB     00:00
(4/4): nginx-core-1.20.1-14.el9_2.1.x86_64.rpm               5.1 MB/s | 565 kB     00:00
--------------------------------------------------------------------------------
Total                                                        1.3 MB/s | 634 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Running scriptlet: nginx-filesystem-1:1.20.1-14.el9_2.1.noarch            1/4
  Installing       : nginx-filesystem-1:1.20.1-14.el9_2.1.noarch            1/4
  Installing       : nginx-core-1:1.20.1-14.el9_2.1.x86_64                  2/4
  Installing       : rocky-logos-httpd-90.14-1.el9.noarch                   3/4
  Installing       : nginx-1:1.20.1-14.el9_2.1.x86_64                       4/4
  Running scriptlet: nginx-1:1.20.1-14.el9_2.1.x86_64                       4/4
  Verifying        : rocky-logos-httpd-90.14-1.el9.noarch                   1/4
  Verifying        : nginx-filesystem-1:1.20.1-14.el9_2.1.noarch            2/4
  Verifying        : nginx-1:1.20.1-14.el9_2.1.x86_64                       3/4
  Verifying        : nginx-core-1:1.20.1-14.el9_2.1.x86_64                  4/4

Installed:
  nginx-1:1.20.1-14.el9_2.1.x86_64              nginx-core-1:1.20.1-14.el9_2.1.x86_64
```

```
[root@localhost ~]# systemctl enable --now nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/
system/nginx.service.
```

Installation package php-fpm :

```
[root@localhost ~]# dnf install php-fpm
Last metadata expiration check: 0:07:00 ago on Thu Nov  9 08:45:35 2023.
Dependencies resolved.
================================================================================
 Package                 Architecture   Version                    Repository         Size
================================================================================
Installing:
 php-fpm                 x86_64         8.0.30-1.el9_2             appstream         1.6 M
Installing dependencies:
 httpd-filesystem        noarch         2.4.53-11.el9_2.5          appstream          14 k
 php-common              x86_64         8.0.30-1.el9_2             appstream         665 k

Transaction Summary
================================================================================
Install  3 Packages

Total download size: 2.3 M
Installed size: 16 M
Is this ok [y/N]: y
Downloading Packages:
(1/3): httpd-filesystem-2.4.53-11.el9_2.5.noarch.rpm         291 kB/s |  14 kB     00:00
(2/3): php-common-8.0.30-1.el9_2.x86_64.rpm                  4.0 MB/s | 665 kB     00:00
(3/3): php-fpm-8.0.30-1.el9_2.x86_64.rpm                     8.5 MB/s | 1.6 MB     00:00
--------------------------------------------------------------------------------
Total                                                        4.8 MB/s | 2.3 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : php-common-8.0.30-1.el9_2.x86_64                       1/3
  Running scriptlet: httpd-filesystem-2.4.53-11.el9_2.5.noarch              2/3
  Installing       : httpd-filesystem-2.4.53-11.el9_2.5.noarch              2/3
  Installing       : php-fpm-8.0.30-1.el9_2.x86_64                          3/3
  Running scriptlet: php-fpm-8.0.30-1.el9_2.x86_64                          3/3
  Verifying        : httpd-filesystem-2.4.53-11.el9_2.5.noarch              1/3
  Verifying        : php-fpm-8.0.30-1.el9_2.x86_64                          2/3
  Verifying        : php-common-8.0.30-1.el9_2.x86_64                       3/3

Installed:
  httpd-filesystem-2.4.53-11.el9_2.5.noarch        php-common-8.0.30-1.el9_2.x86_64
  php-fpm-8.0.30-1.el9_2.x86_64

Complete!
[root@localhost ~]#
```

```
[root@localhost ~]# systemctl enable --now php-fpm
Created symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service → /usr/lib/system
d/system/php-fpm.service.
```

Installation package mariadb-server :

LESPAGNOL
Alexandre
2SISR

09/11/2023

```
[root@localhost ~]# dnf install mariadb-server -y
Last metadata expiration check: 0:08:05 ago on Thu Nov  9 08:45:35 2023.
Dependencies resolved.
================================================================================================
 Package                          Arch        Version                     Repository       Size
================================================================================================
Installing:
 mariadb-server                   x86_64      3:10.5.22-1.el9_2           appstream       9.6 M
Upgrading:
 libselinux                       x86_64      3.5-1.el9                   baseos           85 k
 libselinux-utils                 x86_64      3.5-1.el9                   baseos          159 k
 libsemanage                      x86_64      3.5-1.el9                   baseos          117 k
 libsepol                         x86_64      3.5-1.el9                   baseos          314 k
 policycoreutils                  x86_64      3.5-1.el9                   baseos          204 k
 python3-libselinux               x86_64      3.5-1.el9                   appstream       187 k
Installing dependencies:
 checkpolicy                      x86_64      3.5-1.el9                   appstream       345 k
 mariadb                          x86_64      3:10.5.22-1.el9_2           appstream       1.6 M
 mariadb-common                   x86_64      3:10.5.22-1.el9_2           appstream        27 k
 mariadb-connector-c              x86_64      3.2.6-1.el9_0               appstream       195 k
 mariadb-connector-c-config       noarch      3.2.6-1.el9_0               appstream       9.8 k
 mariadb-errmsg                   x86_64      3:10.5.22-1.el9_2           appstream       211 k
 mysql-selinux                    noarch      1.0.5-1.el9_0               appstream        35 k
 perl-AutoLoader                  noarch      5.74-480.el9                appstream        21 k
 perl-B                           x86_64      1.80-480.el9                appstream       179 k
 perl-Carp                        noarch      1.50-460.el9                appstream        29 k
 perl-Class-Struct                noarch      0.66-480.el9                appstream        22 k
 perl-DBD-MariaDB                 x86_64      1.21-16.el9_0               appstream       151 k
 perl-DBI                         x86_64      1.643-9.el9                 appstream       700 k
 perl-Data-Dumper                 x86_64      2.174-462.el9               appstream        55 k
 perl-Digest                      noarch      1.19-4.el9                  appstream        25 k
 perl-Digest-MD5                  x86_64      2.58-4.el9                  appstream        36 k
 perl-DynaLoader                  x86_64      1.47-480.el9                appstream        26 k
 perl-Encode                      x86_64      4:3.08-462.el9              appstream       1.7 M
 perl-Errno                       x86_64      1.30-480.el9                appstream        15 k
 perl-Exporter                    noarch      5.74-461.el9                appstream        31 k
 perl-Fcntl                       x86_64      1.13-480.el9                appstream        20 k
```

```
[root@localhost ~]# systemctl enable --now 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/system
d/system/mariadb.service.
```

Installation du package php :

```
[root@localhost ~]# dnf install php -y
Last metadata expiration check: 0:09:14 ago on Thu Nov  9 08:45:35 2023.
Dependencies resolved.
================================================================================
 Package                 Architecture   Version                   Repository        Size
================================================================================
Installing:
 php                     x86_64         8.0.30-1.el9_2            appstream        7.7 k
Installing dependencies:
 apr                     x86_64         1.7.0-11.el9              appstream        123 k
 apr-util                x86_64         1.6.1-20.el9_2.1          appstream         94 k
 apr-util-bdb            x86_64         1.6.1-20.el9_2.1          appstream         12 k
 httpd-core              x86_64         2.4.53-11.el9_2.5         appstream        1.4 M
 httpd-tools             x86_64         2.4.53-11.el9_2.5         appstream         81 k
 libxslt                 x86_64         1.1.34-9.el9              appstream        240 k
 mailcap                 noarch         2.1.49-5.el9              baseos            32 k
 oniguruma               x86_64         6.9.6-1.el9.5             appstream        217 k
Installing weak dependencies:
 apr-util-openssl        x86_64         1.6.1-20.el9_2.1          appstream         14 k
 httpd                   x86_64         2.4.53-11.el9_2.5         appstream         47 k
 mod_http2               x86_64         1.15.19-4.el9_2.4         appstream        149 k
 mod_lua                 x86_64         2.4.53-11.el9_2.5         appstream         61 k
 php-cli                 x86_64         8.0.30-1.el9_2            appstream        3.1 M
 php-mbstring            x86_64         8.0.30-1.el9_2            appstream        468 k
 php-opcache             x86_64         8.0.30-1.el9_2            appstream        509 k
 php-pdo                 x86_64         8.0.30-1.el9_2            appstream         81 k
 php-xml                 x86_64         8.0.30-1.el9_2            appstream        131 k

Transaction Summary
================================================================================
Install  18 Packages

Total download size: 6.7 M
Installed size: 27 M
Downloading Packages:
(1/18): mailcap-2.1.49-5.el9.noarch.rpm                  183 kB/s |  32 kB     00:00
(2/18): mod_lua-2.4.53-11.el9_2.5.x86_64.rpm             248 kB/s |  61 kB     00:00
(3/18): httpd-tools-2.4.53-11.el9_2.5.x86_64.rpm         326 kB/s |  81 kB     00:00
(4/18): apr-util-openssl-1.6.1-20.el9_2.1.x86_64.rpm     929 kB/s |  14 kB     00:00
(5/18): apr-util-bdb-1.6.1-20.el9_2.1.x86_64.rpm         818 kB/s |  12 kB     00:00
(6/18): apr-util-1.6.1-20.el9_2.1.x86_64.rpm             3.3 MB/s |  94 kB     00:00
(7/18): httpd-2.4.53-11.el9_2.5.x86_64.rpm               395 kB/s |  47 kB     00:00
(8/18): mod_http2-1.15.19-4.el9_2.4.x86_64.rpm           4.5 MB/s | 149 kB     00:00
(9/18): apr-1.7.0-11.el9.x86_64.rpm                      4.0 MB/s | 123 kB     00:00
(10/18): libxslt-1.1.34-9.el9.x86_64.rpm                 6.2 MB/s | 240 kB     00:00
(11/18): oniguruma-6.9.6-1.el9.5.x86_64.rpm              6.2 MB/s | 217 kB     00:00
```

Installation du package php-mysqlnd :

```
[root@localhost ~]# dnf install php-mysqlnd -y
Last metadata expiration check: 0:10:14 ago on Thu Nov  9 08:45:35 2023.
Dependencies resolved.
================================================================================
 Package              Architecture     Version                  Repository          Size
================================================================================
Installing:
 php-mysqlnd          x86_64           8.0.30-1.el9_2           appstream          148 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 148 k
Installed size: 450 k
Downloading Packages:
php-mysqlnd-8.0.30-1.el9_2.x86_64.rpm                    1.3 MB/s | 148 kB     00:00
--------------------------------------------------------------------------------
Total                                                    405 kB/s | 148 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : php-mysqlnd-8.0.30-1.el9_2.x86_64                      1/1
  Running scriptlet: php-mysqlnd-8.0.30-1.el9_2.x86_64                      1/1
  Verifying        : php-mysqlnd-8.0.30-1.el9_2.x86_64                      1/1

Installed:
  php-mysqlnd-8.0.30-1.el9_2.x86_64

Complete!
[root@localhost ~]#
```

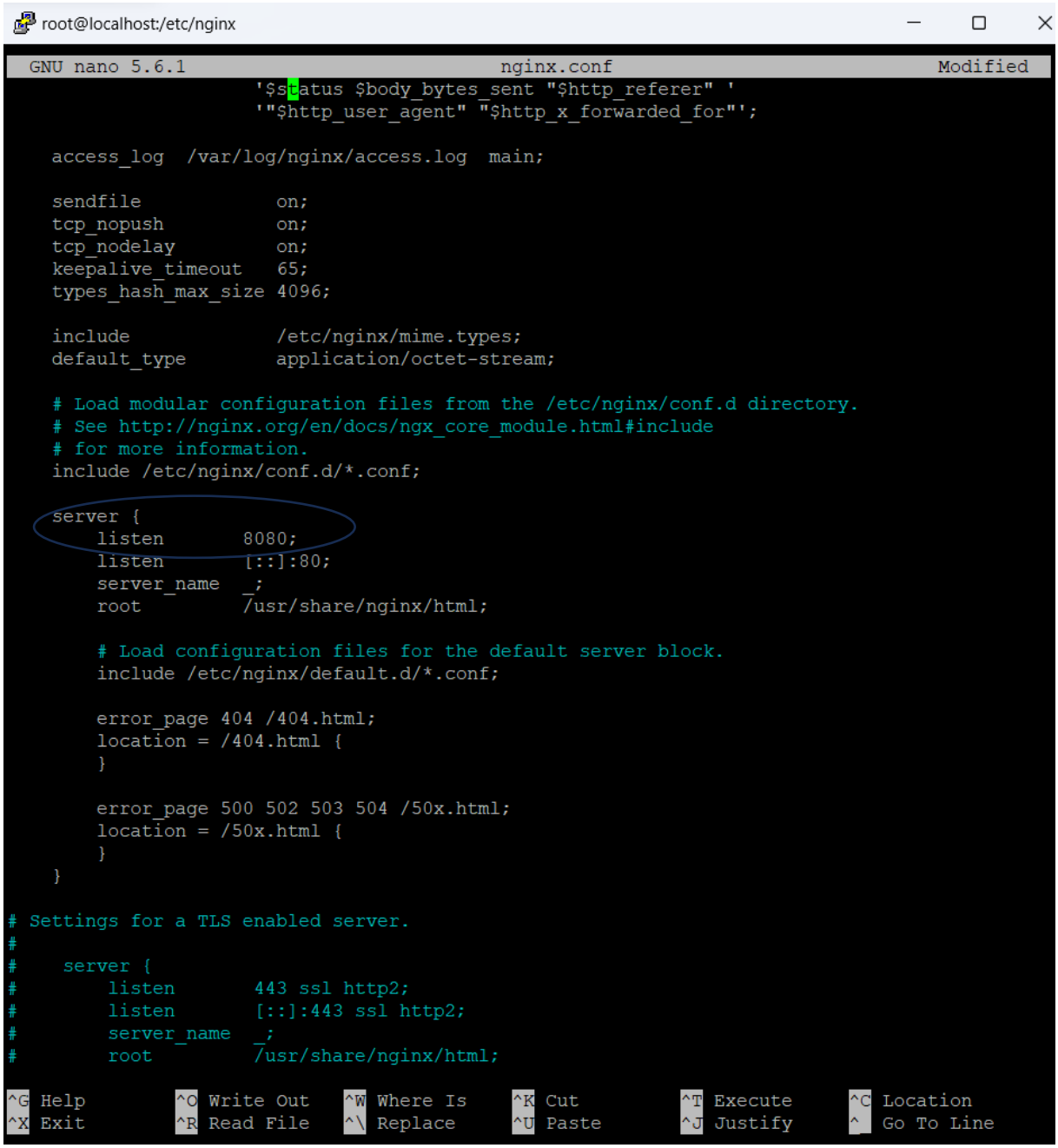
Autoriser le service http :

```
[root@localhost ~]#  firewall-cmd --add-service=http --permanent
success
```

```
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]# firewall-cmd --list-service
cockpit dhcpv6-client http ssh
```

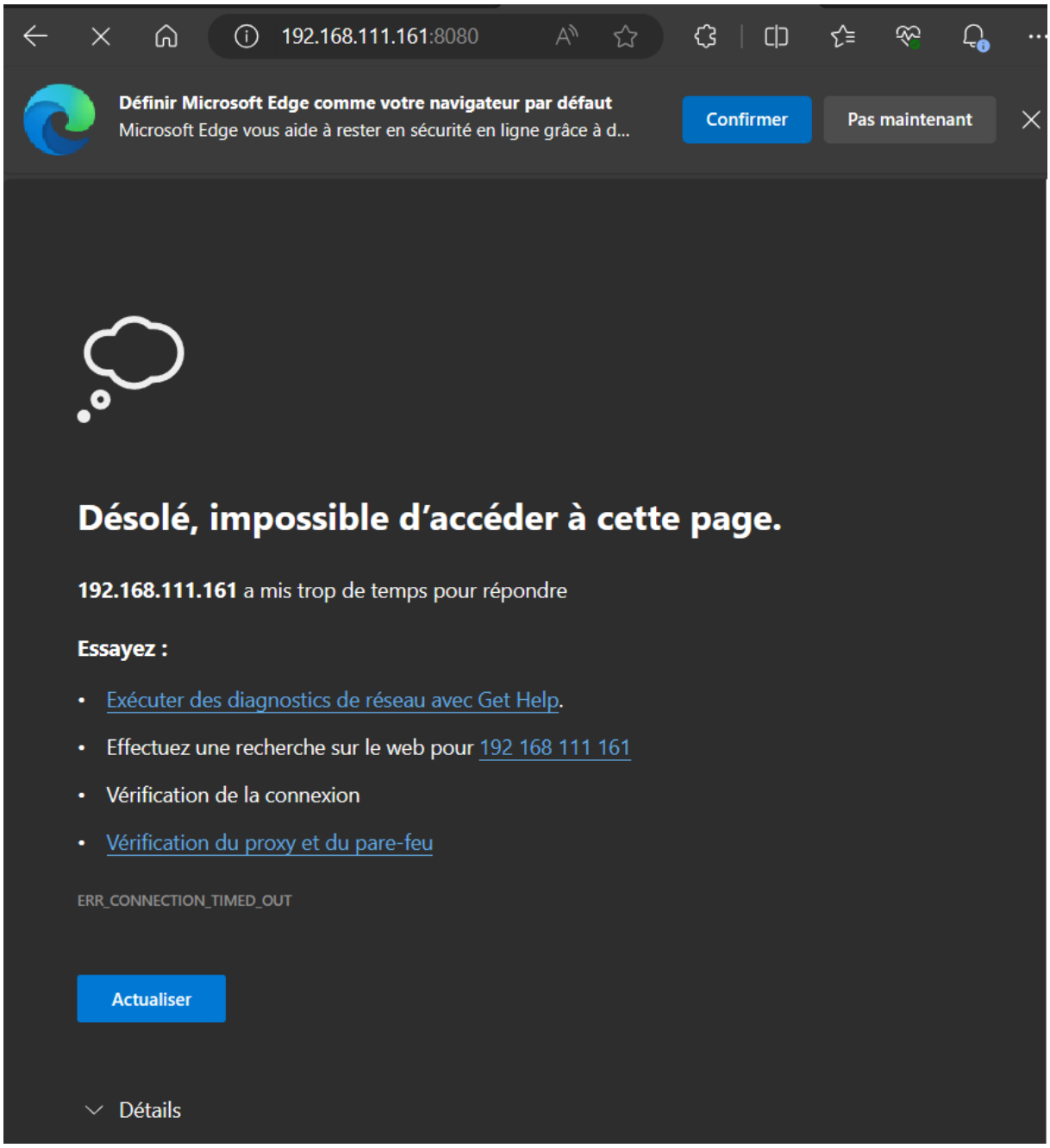
Modifions le port d'écoute de nginx (par défaut 80 pour le protocole html), Il faut se rendre dans le fichier de configuration nginx et modifier le paramètre :

```
[root@localhost ~]# cd /etc/nginx
[root@localhost nginx]# ls
conf.d                fastcgi_params.default  nginx.conf          uwsgi_params.default
default.d             koi-utf                 nginx.conf.default  win-utf
fastcgi.conf          koi-win                 scgi_params
fastcgi.conf.default  mime.types              scgi_params.default
fastcgi_params        mime.types.default      uwsgi_params
[root@localhost nginx]# nano nginx.conf
```



On relance le service et testons une connexion avec le port :

```
[root@localhost nginx]# service nginx restart
Redirecting to /bin/systemctl restart nginx.service
```



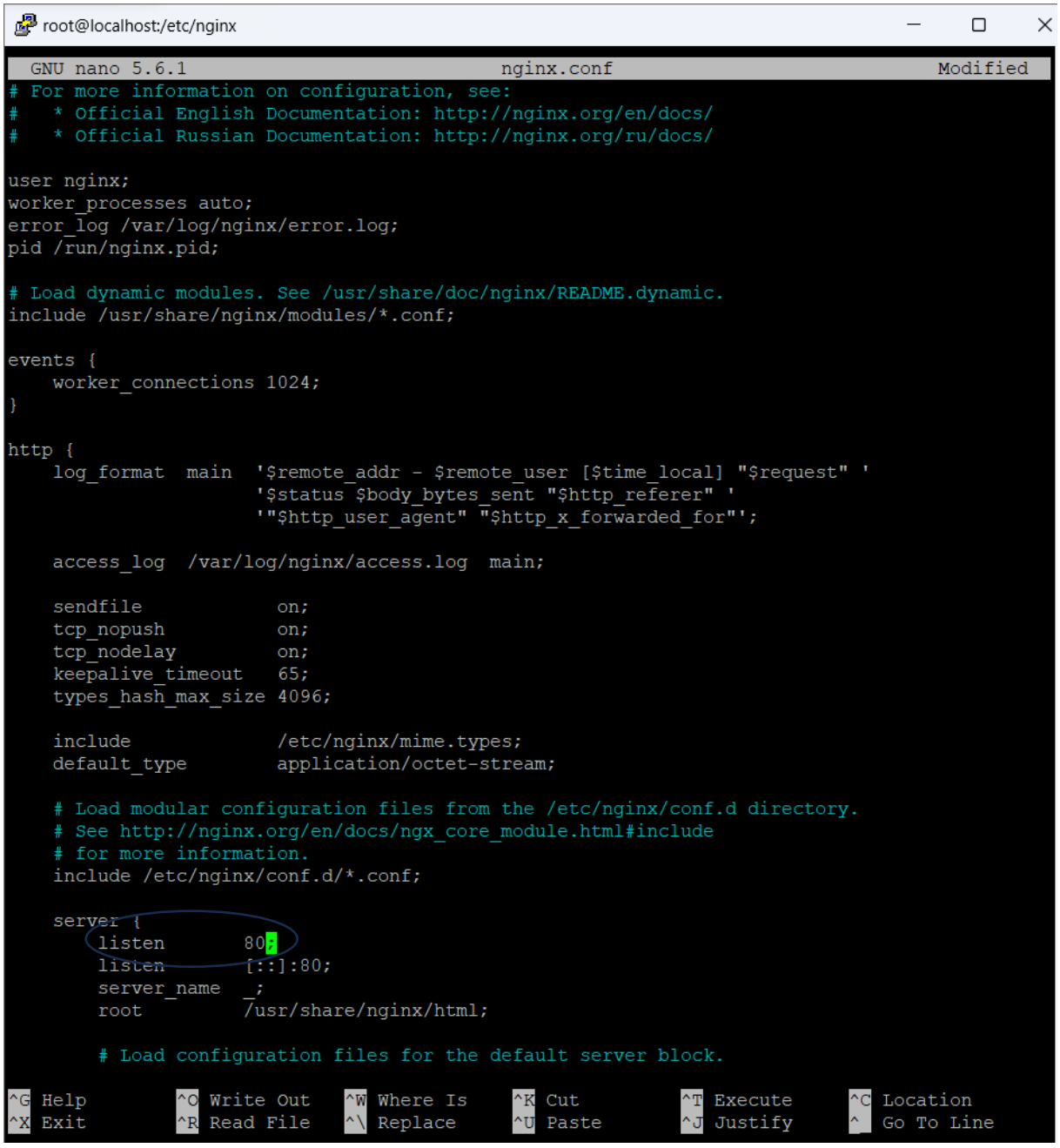
Cela ne marche pas car le firewall bloque ceci il faut donc ceci :

```
[root@localhost nginx]# firewall-cmd --add-port=8080/tcp --permanent
success
```

```
[root@localhost nginx]# firewall-cmd --reload
success
```

Remettons le port http par défaut, c'est-à-dire 80 et on redémarre le service. service nginx restart :

```
[root@localhost nginx]# cd /etc/nginx
[root@localhost nginx]# nano nginx.conf
```




```
[root@localhost nginx]# service nginx restart
Redirecting to /bin/systemctl restart nginx.service
```

Installation moteur BDD

```
[root@localhost ~]# systemctl enable --now 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/system
d/system/mariadb.service.
```

Faire la configuration de base :

```
root@localhost:~
[root@localhost ~]# 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.

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
```

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

Reload privilege tables now? [Y/n] n
 ... skipping.

Cleaning up...

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

Thanks for using MariaDB!
```

Se connecter :

```
[root@localhost ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
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)]>
```

Créér une bdd :

```
MariaDB [(none)]> create database web;
Query OK, 1 row affected (0.001 sec)
```

Créer un utilisateur :

```
MariaDB [(none)]> CREATE USER 'USER-web'@'localhost' IDENTIFIED BY 'web';
Query OK, 0 rows affected (0.001 sec)
```

Attribution des droits :

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON BDD.* TO 'USER-web'@'localhost';
Query OK, 0 rows affected (0.001 sec)
```

Forcer la prise en compte :

```
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)
```

Permet de voir les BDDs :

```
MariaDB [(none)]> Show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| web                |
+--------------------+
4 rows in set (0.000 sec)
```

Voir les tables :

```
MariaDB [(none)]> use web;
Database changed
MariaDB [web]> show tables;
Empty set (0.000 sec)
```

On quitte mariadb :

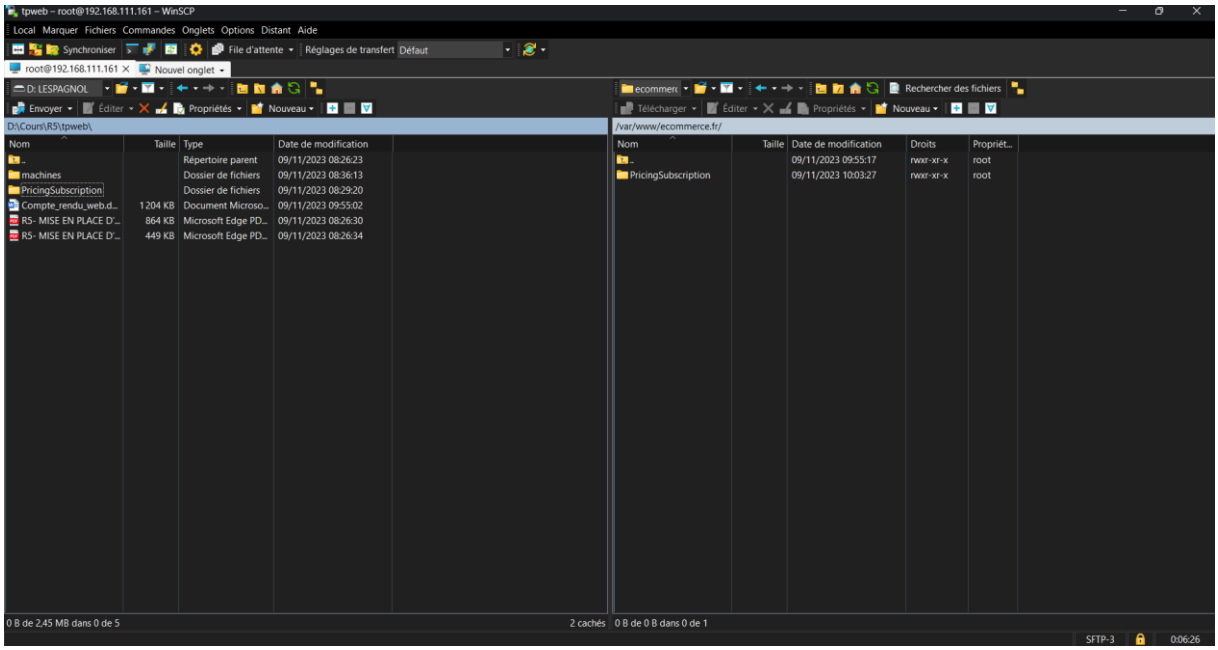
```
MariaDB [web]> exit
Bye
```

Déploiement du site FR

Nous allons déployer le site FR. Nous allons créer le répertoire ecommerce.fr/ dans l'arborescence /var/www/ :

```
[root@localhost ~]# mkdir -p /var/www/ecommerce.fr/
```

Puis avec le package fournis, vous déposerez l'ensemble des fichiers dans le répertoire ecommerce.fr :



On remarque que cela a bien mis notre dossier où nous le souhaitons :

```
[root@localhost ~]# cd /var/www/
[root@localhost www]# ls
cgi-bin  ecommerce.fr  html
[root@localhost www]# cd ecommerce.fr
[root@localhost ecommerce.fr]# ls
PricingSubscription
[root@localhost ecommerce.fr]#
```

Maintenant, nous allons démarrer modifier les paramètres de php-fpm RDV dans : /etc/php-fpm.d/www.conf (Vous noterez la valeur listen, par défaut : listen = /run/php-fpm/www.sock) Puis modifier le fichier avec les informations ci-contre :

root@localhost:/var/www/ecommerce.fr

```
  GNU nano 5.6.1                        /etc/php-fpm.d/www.conf
; Start a new pool named 'www'.
; the variable $pool can be used in any directive and will be replaced by the
; pool name ('www' here)
[www]

; Per pool prefix
; It only applies on the following directives:
; - 'access.log'
; - 'slowlog'
; - 'listen' (unixsocket)
; - 'chroot'
; - 'chdir'
; - 'php_values'
; - 'php_admin_values'
; When not set, the global prefix (or @php_fpm_prefix@) applies instead.
; Note: This directive can also be relative to the global prefix.
; Default Value: none
;prefix = /path/to/pools/$pool

; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the default user's group
;       will be used.
; RPM: apache user chosen to provide access to the same directories as httpd
user = apache
; RPM: Keep a group allowed to write in log dir.
group = apache

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
;   'ip.add.re.ss:port'    - to listen on a TCP socket to a specific IPv4 address on
;                            a specific port;
;   '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific IPv6 address on
;                            a specific port;
;   'port'                 - to listen on a TCP socket to all addresses
;                            (IPv6 and IPv4-mapped) on a specific port;
;   '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
listen = /run/php-fpm/www.sock

; Set listen(2) backlog.
; Default Value: 511
;listen.backlog = 511

; Set permissions for unix socket, if one is used. In Linux, read/write

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^_ Go To Line
```

Ensuite, vous pouvez activer le démarrage automatique et démarrer dés maintenant le service :

```
[root@localhost ~]# systemctl enable --now php-fpm
Created symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service → /usr/lib/system
d/system/php-fpm.service.
```

Nous allons pouvoir maintenant créer le fichier de configuration nginx

Il faut le créer dans le répertoire de configuration /etc/nginx /conf.d/ :

```
[root@localhost ~]# nano /etc/nginx/conf.d/site_commerce.fr.conf
```

root@localhost:~

```
 GNU nano 5.6.1              /etc/nginx/conf.d/site_commerce.fr.conf              Modified
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, configurer le fichier de configuration php pour établir la connexion avec le moteur de bdd.

Il faut modifier le fichier conf.php se (que vous venez de déposer dans /var/www/site_commerce.fr) Vous devez renseigner les informations avec les informations lors de la création de la base et de l'utilisateur :

```
[root@localhost ~]# cd /var/www/ecommerce.fr/PricingSubscription/
[root@localhost PricingSubscription]# ls
config.php  css  database.sql  images  js  sign-up.php
[root@localhost PricingSubscription]# nano config.php
```

root@localhost:/var/www/ecommerce.fr/PricingSubscription

```
 GNU nano 5.6.1                    config.php                          Modified
<?php
$SETTINGS["mysql_user"]='USER-web';
$SETTINGS["mysql_pass"]='web';
$SETTINGS["hostname"]='localhost';
$SETTINGS["mysql_database"]='web';
$SETTINGS["data_table"]='registrations';
$SETTINGS["paypal_address"]='email@domain.com';
?>
```

Nous allons maintenant exécuter le script SQL fournit (database.sql). Pour cela, il faut se connecter sur le moteur BDD et sélectionner la base :

```
[root@localhost PricingSubscription]# ls
config.php  css  database.sql  images  js  sign-up.php
[root@localhost PricingSubscription]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
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 web
Database changed
```

```
MariaDB [web]> source /var/www/ecommerce.fr/PricingSubscription/database.sql
Query OK, 0 rows affected (0.002 sec)
```

```
MariaDB [web]> exit
Bye
```

```
[root@localhost ~]# cd /etc/nginx/conf.d/
[root@localhost conf.d]# ls
php-fpm.conf  site_commerce.fr.conf
[root@localhost conf.d]# nano site_commerce.fr.conf
```

root@localhost:/etc/nginx/conf.d

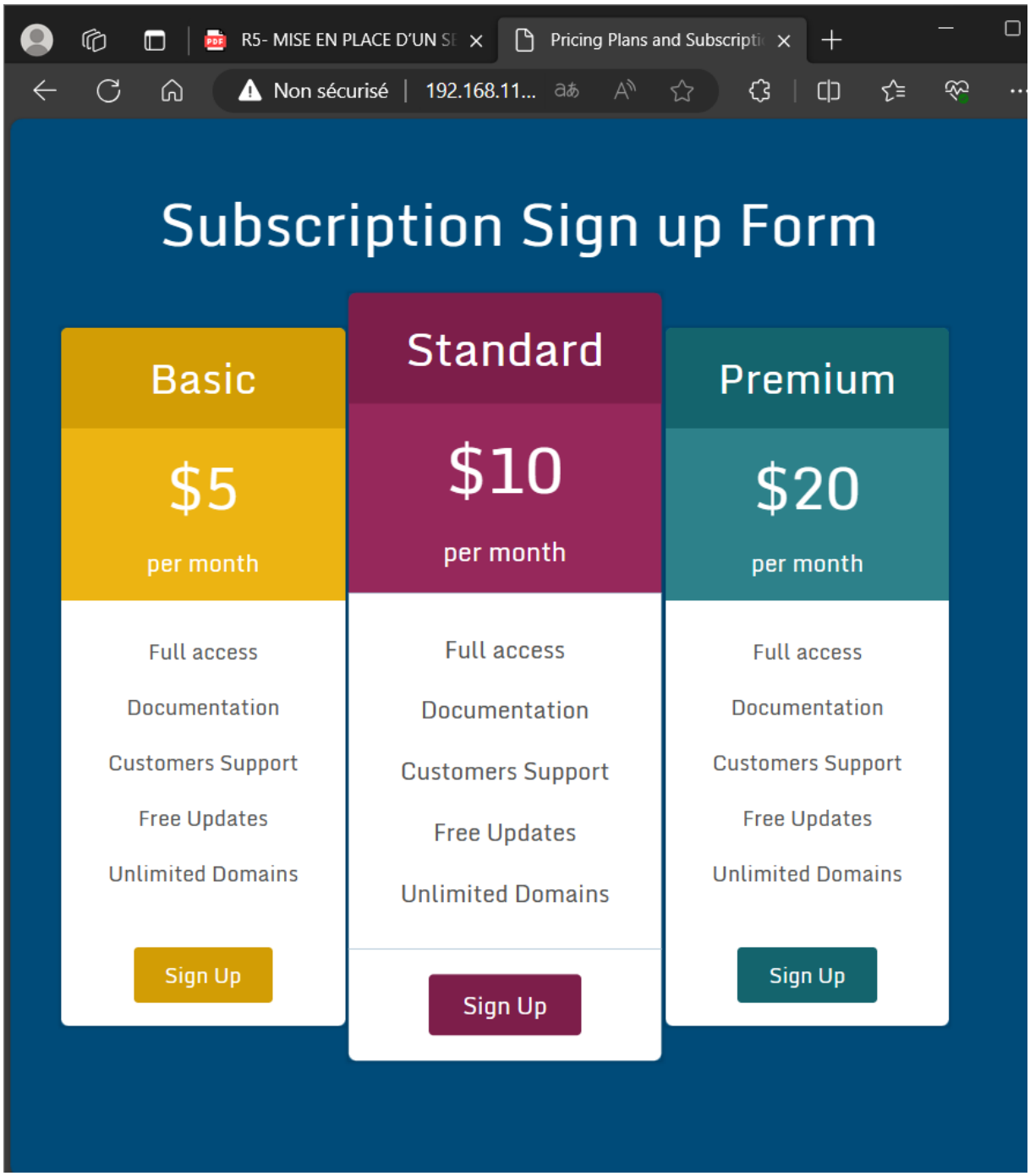
```
  GNU nano 5.6.1                    site_commerce.fr.conf                    Modified
server {
        listen 80;
        listen [::]:80;


        root /var/www/ecommerce.fr/PricingSubscription/;
        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;
        }
}
```

```
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^_ Go To Line
```

On doit faire une relance du nginx afin de faire nos tests :

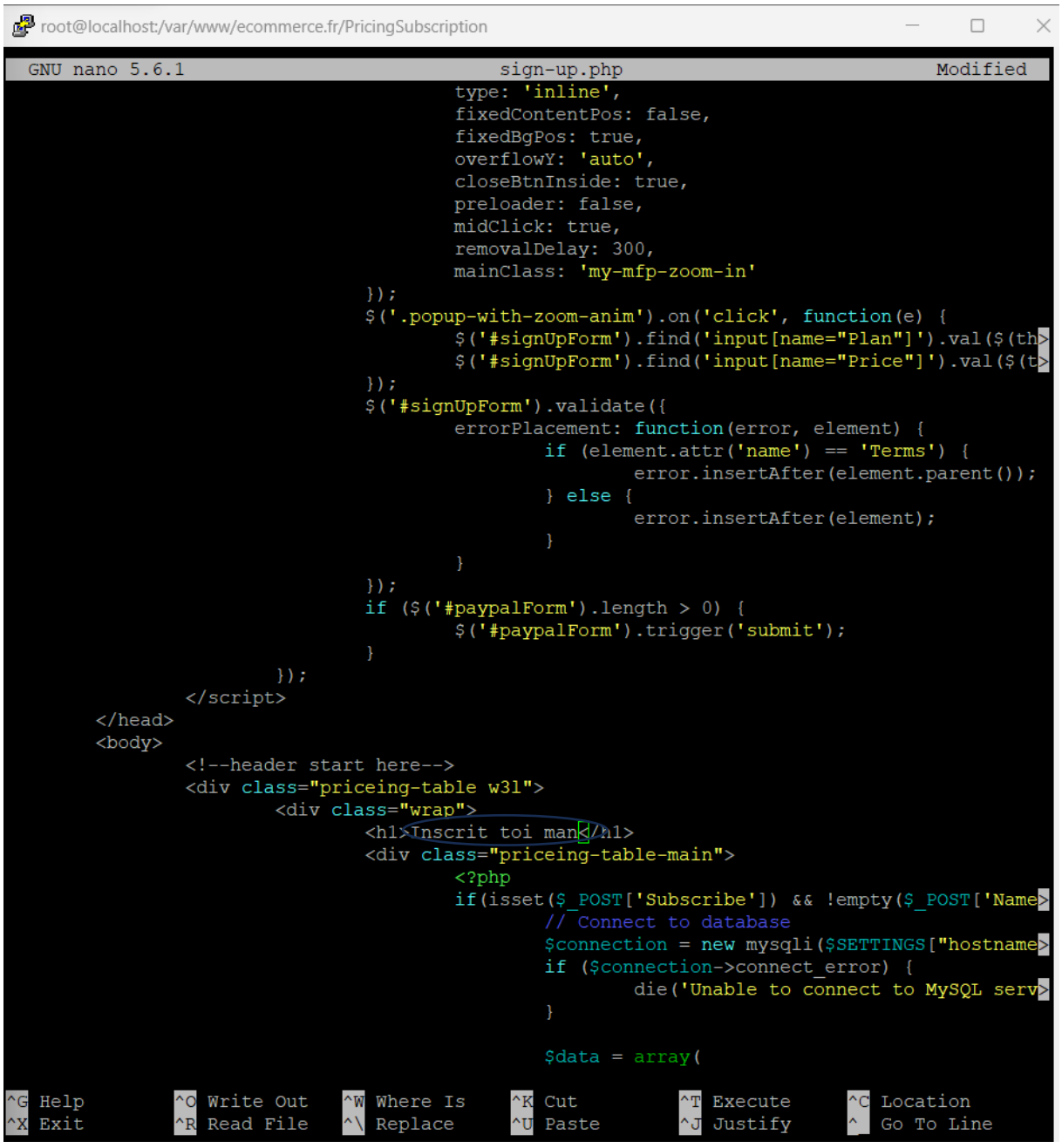
```
[root@localhost ~]# service nginx restart
Redirecting to /bin/systemctl restart nginx.service
```

Si tout fonctionne, lorsque vous tapez @duserveur, vous devrez avoir l'affichage ci-contre :

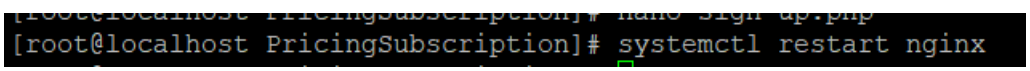


Maintenant à vous de jouer pour personnaliser la page en FR (le fichier de configuration est le suivant : sign-up.php :

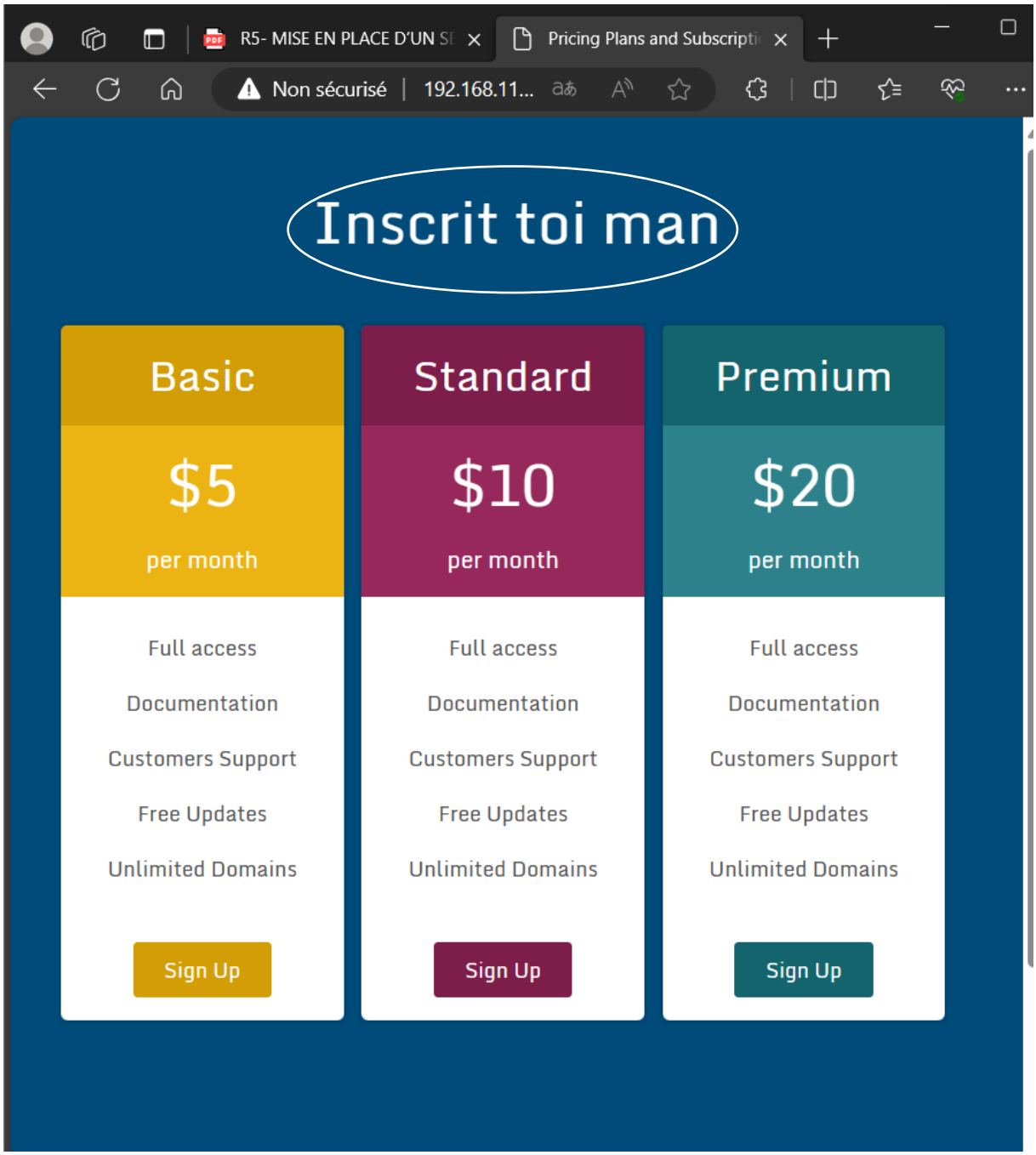
```
[root@localhost ~]# cd /var/www/
[root@localhost www]# ls
cgi-bin  ecommerce.fr  html
[root@localhost www]# cd ecommerce.fr/
[root@localhost ecommerce.fr]# ls
PricingSubscription
[root@localhost ecommerce.fr]# cd PricingSubscription/
[root@localhost PricingSubscription]# ls
config.php  css  database.sql  images  js  sign-up.php
[root@localhost PricingSubscription]# nano sign-up.php
```



On redémarre nginx :



On remarque que nos modifications ont bien été pris en compte en rafraichissant notre page web :



Déploiement du site UK

Nous allons maintenant déployer le 2eme site UK :

Nous allons déployer el site FR. Nous allons créer le répertoire ecommerce.fr/ dans l'arborescence /var/www/ :

```
[root@localhost ~]# mkdir -p /var/www/ecommerce.uk/
```

Puis avec le package fournis, vous déposerez l'ensemble des fichiers dans le répertoire ecommerce.uk :

```
[root@localhost ~]# cp /var/www/ecommerce.fr/PricingSubscription/ /var/www/ecommerce.uk/ -R
[root@localhost ~]# cd /var/www/ecommerce.uk/
[root@localhost ecommerce.uk]# ls
PricingSubscription
[root@localhost ecommerce.uk]# cd PricingSubscription/
[root@localhost PricingSubscription]# ls
config.php  css  database.sql  images  js  sign-up.php
[root@localhost PricingSubscription]#
```

On modifie notre fichier de configuration sign-up.php vu que l'on a copié celui du fr :

```
[root@localhost PricingSubscription]# nano sign-up.php
```

root@localhost:/var/www/ecommerce.uk/PricingSubscription

```
 GNU nano 5.6.1                          sign-up.php                          Modified
                <link href="https://fonts.googleapis.com/css?family=Monda" rel="stylesheet">

                <script src="js/jquery-1.11.0.min.js"></script>
                <script src="js/jquery.magnific-popup.js" type="text/javascript"></script>
                <script src="js/jquery.validate.min.js" type="text/javascript"></script>
                <script>
                        $(document).ready(function() {
                                $('.popup-with-zoom-anim').magnificPopup({
                                        type: 'inline',
                                        fixedContentPos: false,
                                        fixedBgPos: true,
                                        overflowY: 'auto',
                                        closeBtnInside: true,
                                        preloader: false,
                                        midClick: true,
                                        removalDelay: 300,
                                        mainClass: 'my-mfp-zoom-in'
                                });
                                $('.popup-with-zoom-anim').on('click', function(e) {
                                        $('#signUpForm').find('input[name="Plan"]').val($(th>
                                        $('#signUpForm').find('input[name="Price"]').val($(t>
                                });
                                $('#signUpForm').validate({
                                        errorPlacement: function(error, element) {
                                                if (element.attr('name') == 'Terms') {
                                                        error.insertAfter(element.parent());
                                                } else {
                                                        error.insertAfter(element);
                                                }
                                        }
                                });
                                if ($('#paypalForm').length > 0) {
                                        $('#paypalForm').trigger('submit');
                                }
                        });
                </script>
        </head>
        <body>
                <!--header start here-->
                <div class="priceing-table w3l">
                        <div class="wrap">
                                <h1>Subscription Sign Up Form</h1>
                                <div class="priceing-table-main">
                                        <?php

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^_ Go To Line
```

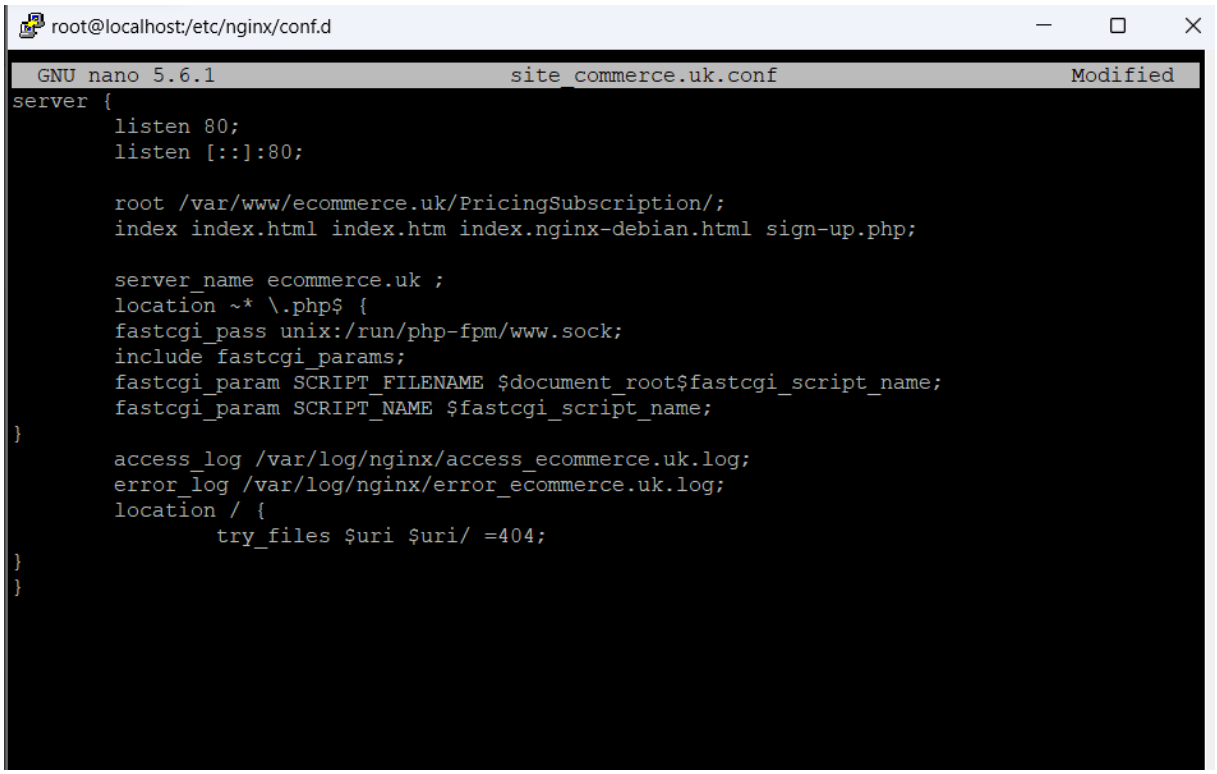
Création de la nouvelle bdd :

```
MariaDB [(none)]> create database webuk;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| web                |
| webuk              |
+--------------------+
5 rows in set (0.000 sec)

MariaDB [(none)]> CREATE USER 'USER-WEB-UK'@'localhost' IDENTIFIED BY 'web';
ERROR 1396 (HY000): Operation CREATE USER failed for 'USER-WEB-UK'@'localhost'
MariaDB [(none)]> CREATE USER 'USERWEBUK'@'localhost' IDENTIFIED BY 'web';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON webuk.* TO 'USERWEBUK'@'localhost';
Query OK, 0 rows affected (0.001 sec)

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

MariaDB [(none)]> show dadabases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds t
o your MariaDB server version for the right syntax to use near 'dadabases' at line 1
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| web                |
| webuk              |
+--------------------+
5 rows in set (0.000 sec)
```

Nous allons pouvoir maintenant créer le fichier de configuration nginx. Il faut le créer dans le répertoire de configuration /etc/nginx /conf.d/ Son nom sera site_commerce.uk.conf :

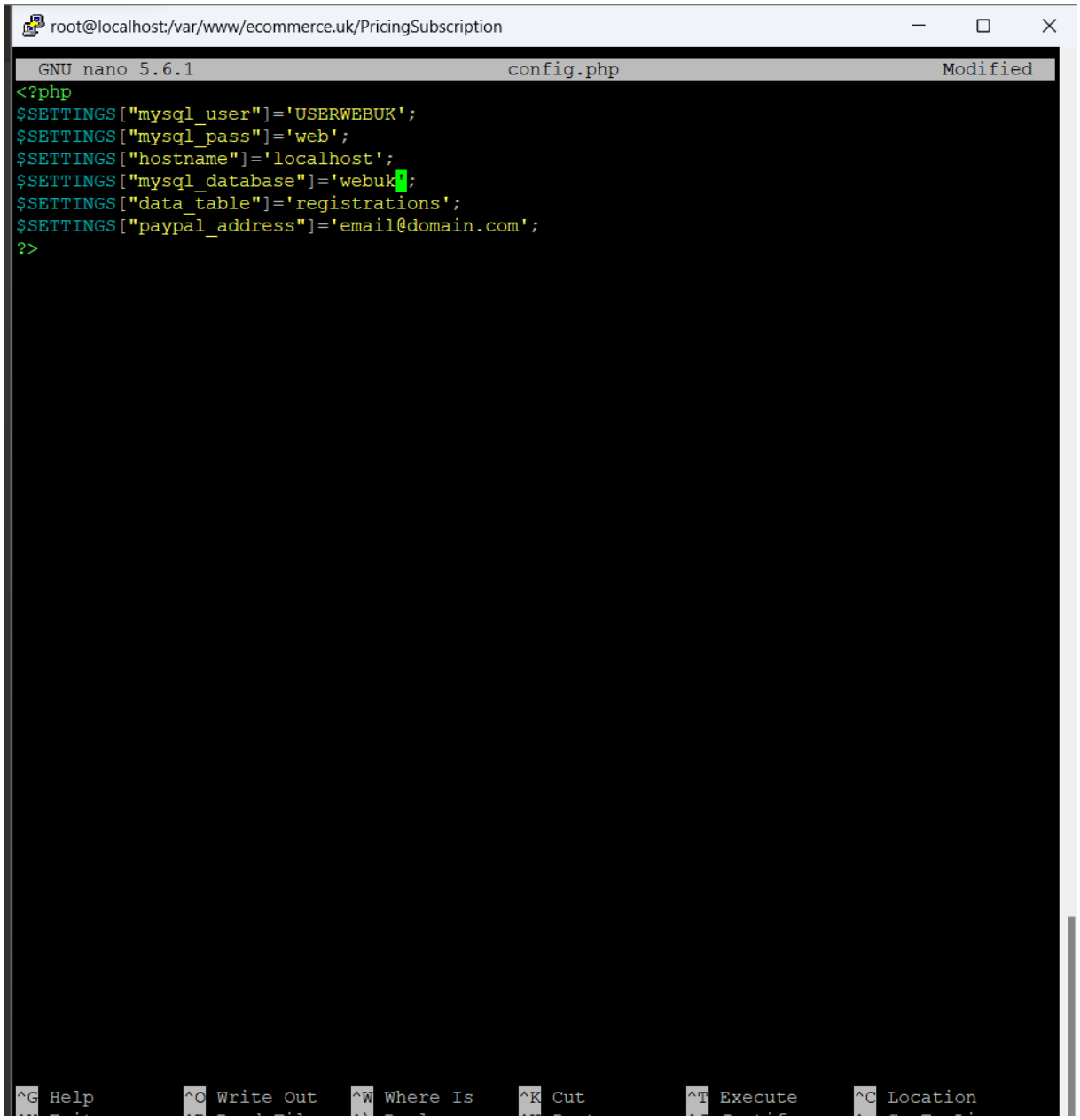
```
[root@localhost ~]# cd /etc/nginx/
[root@localhost nginx]# ls
conf.d                fastcgi_params.default  nginx.conf          uwsgi_params.default
default.d             koi-utf                 nginx.conf.default  win-utf
fastcgi.conf          koi-win                 scgi_params
fastcgi.conf.default  mime.types              scgi_params.default
fastcgi_params        mime.types.default      uwsgi_params
[root@localhost nginx]# nano conf.d
[root@localhost nginx]# cd conf.d
[root@localhost conf.d]# ls
php-fpm.conf  site_commerce.fr.conf
[root@localhost conf.d]# nano site_commerce.uk.conf
```

On modifie le fichier de conf :



Il faut modifier le fichier conf.php se (que vous venez de déposer dans /var/www/site_commerce.fr) Vous devez renseigner les informations avec les informations lors de la création de la base et de l'utilisateur :

```
[root@localhost conf.d]# cd
[root@localhost ~]# cd /var/www/
[root@localhost www]# ls
cgi-bin  ecommerce.fr  ecommerce.uk  html
[root@localhost www]# cd ecommerce.uk/
[root@localhost ecommerce.uk]# cd PricingSubscription/
[root@localhost PricingSubscription]# ls
config.php  css  database.sql  images  js  sign-up.php
[root@localhost PricingSubscription]# nano database.sql
[root@localhost PricingSubscription]# nano c
config.php  css/
[root@localhost PricingSubscription]# nano config.php
```

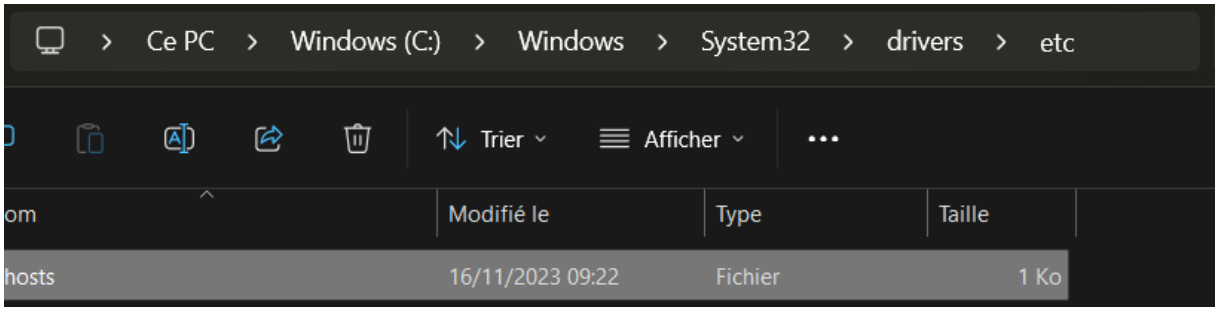


Nous allons maintenant exécuter le script SQL fournit (database.sql). Pour cela, il faut se connecter sur le moteur BDD et sélectionner la base. Nous pouvons exécuter le script en faisant source emplacement_du_script :

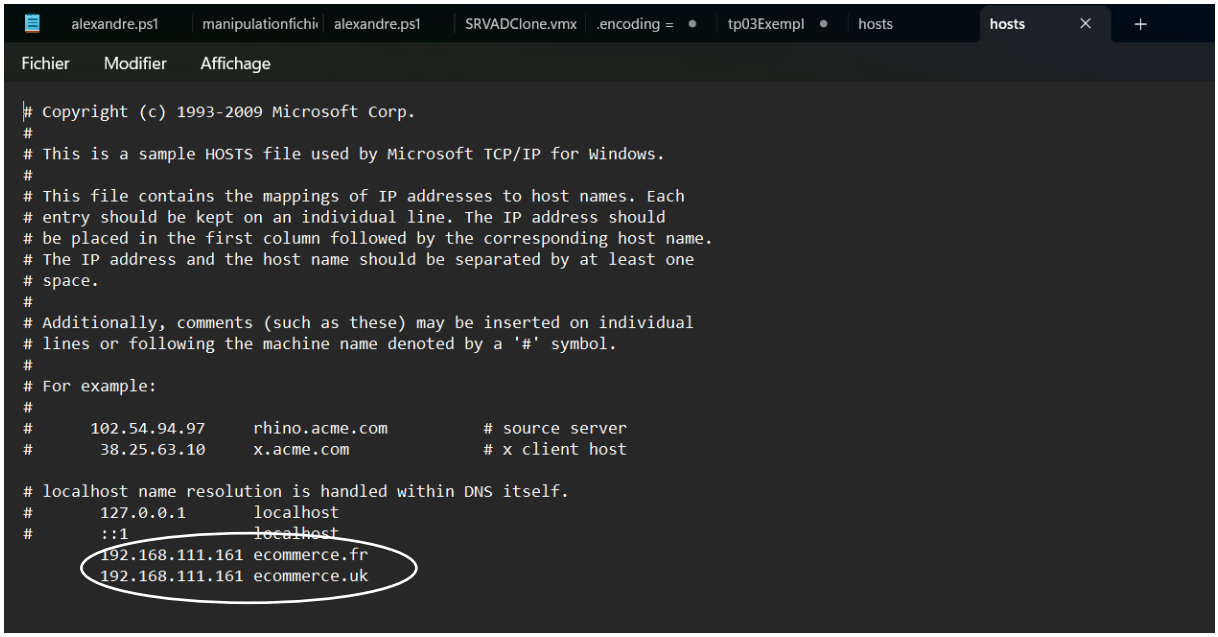
```
MariaDB [(none)]> use webuk
Database changed
MariaDB [webuk]> source /var/www/ecommerce.uk/PricingSubscription/database.sql;
Query OK, 0 rows affected (0.008 sec)
```

```
[root@localhost ~]# systemctl restart mariadb
[root@localhost ~]# systemctl restart nginx
```

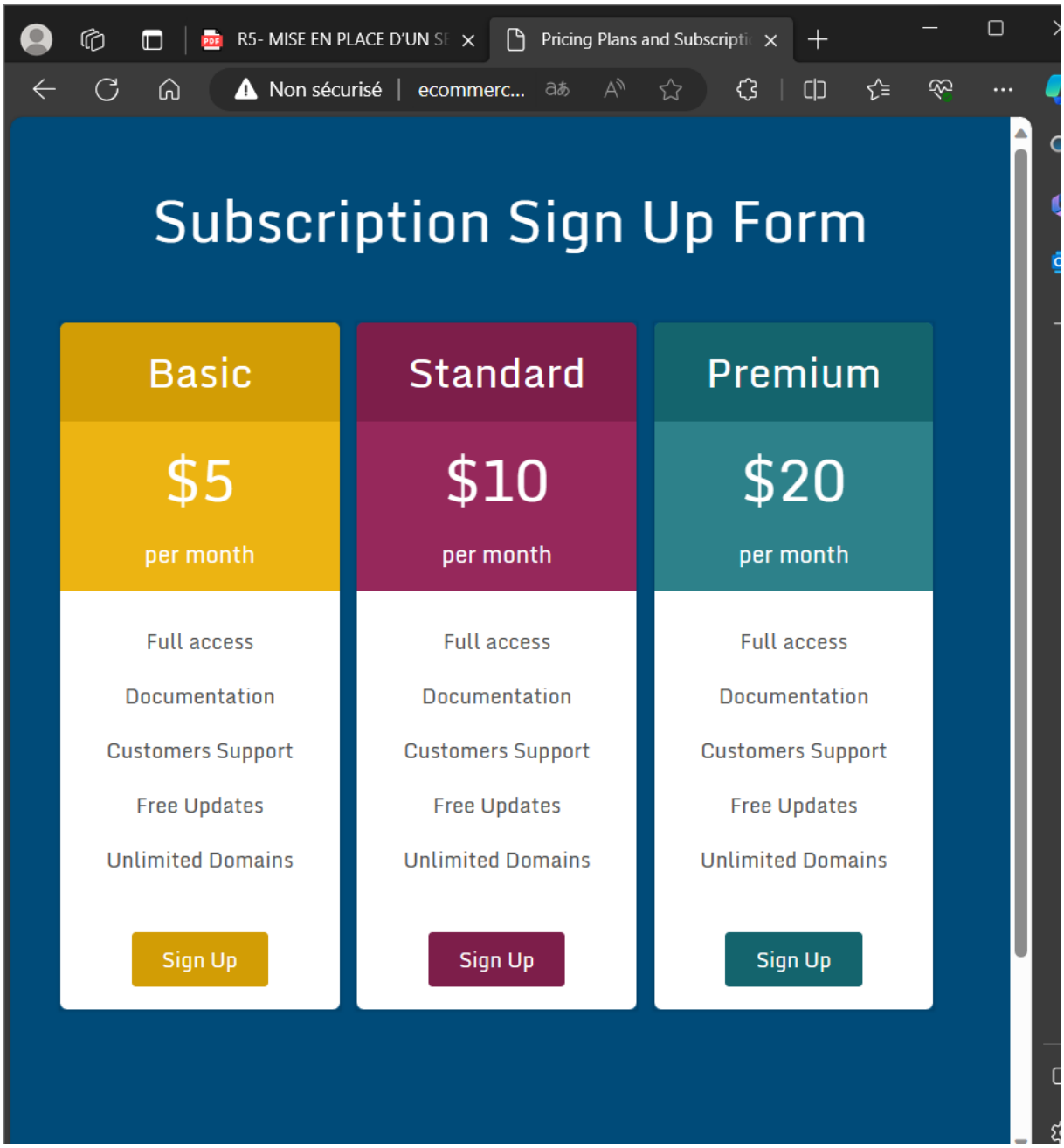
Pour changer les résolution dns local :



On édit le fichier host en tant administrateur et on associe notre ip en nom de domaine :



Et voilà nous avons bien notre site en uk:


R5- MISE EN PLACE D'UN SI
Pricing Plans and Subscripti
Non sécurisé | ecommerc...
Subscription Sign Up Form
Basic
$5
per month
Full access
Documentation
Customers Support
Free Updates
Unlimited Domains
Sign Up
Standard
$10
per month
Full access
Documentation
Customers Support
Free Updates
Unlimited Domains
Sign Up
Premium
$20
per month
Full access
Documentation
Customers Support
Free Updates
Unlimited Domains
Sign Up

On peut voir pour chaque site, les logs d'accès :

```
[root@localhost conf.d]# cd /var/log/nginx/
[root@localhost nginx]# pwd
/var/log/nginx
[root@localhost nginx]# ls
access.log              access_ecommerce.uk.log  error_ecommerce.fr.log
access_ecommerce.fr.log  error.log               error_ecommerce.uk.log
[root@localhost nginx]# ls -lrt
total 24
-rw-r--r--. 1 root root  528 Nov  9 09:26 error.log
-rw-r--r--. 1 root root 1905 Nov  9 09:26 access.log
-rw-r--r--. 1 root root    0 Nov  9 10:36 error_ecommerce.fr.log
-rw-r--r--. 1 root root    0 Nov 16 09:18 error_ecommerce.uk.log
-rw-r--r--. 1 root root 4827 Nov 16 09:22 access_ecommerce.fr.log
-rw-r--r--. 1 root root 4830 Nov 16 09:27 access_ecommerce.uk.log
```

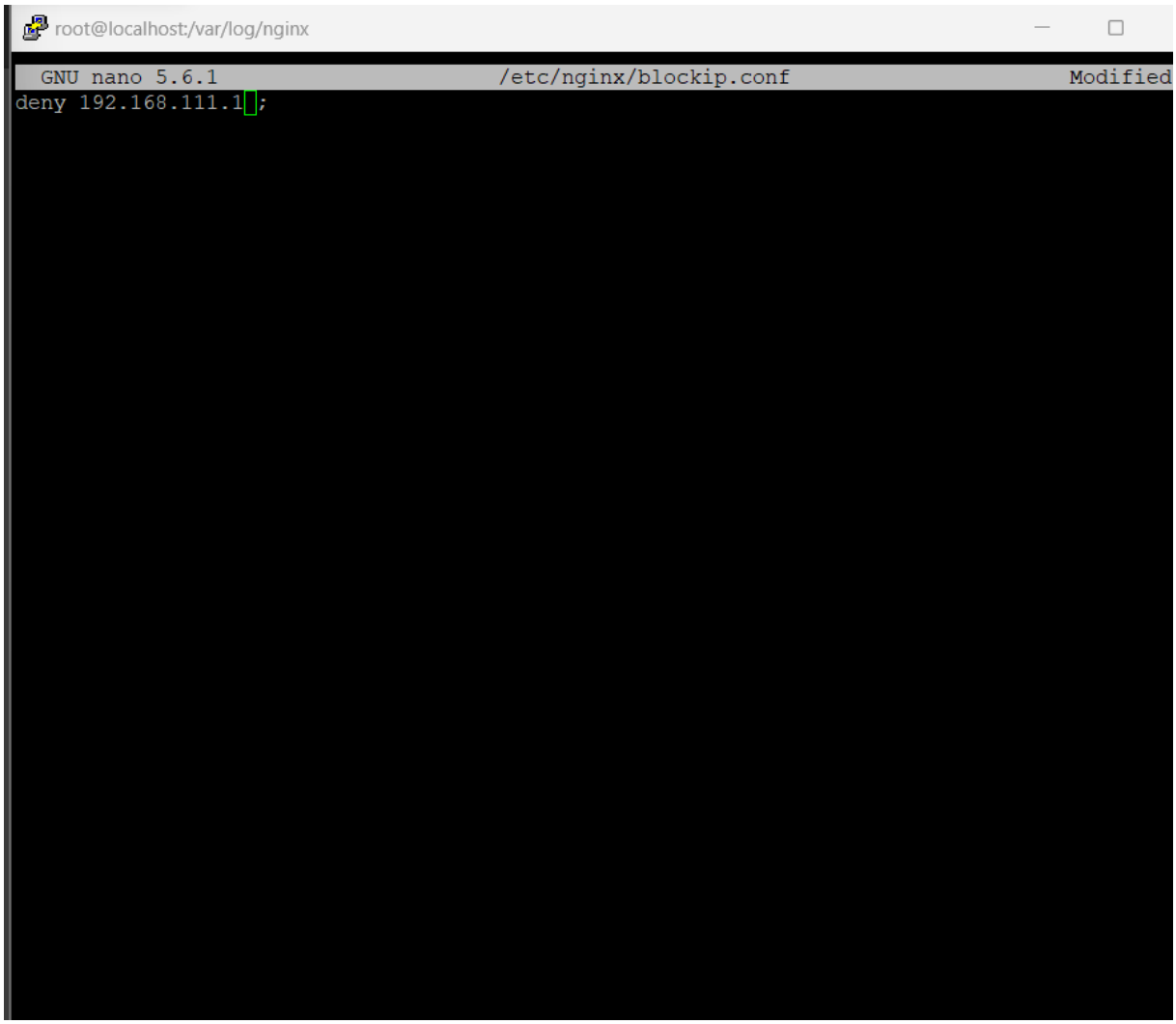
Nos sites sont donc bien opérationnels maintenant on va faire un snapshot de notre vm.

Restriction accès IP

Certains pirates s'amusent à tester les vulnérabilités de notre site internet. La bonne nouvelle, nous avons les adresses IP des pirates. Nous allons ajouter une restriction d'ip sur le site ecommerce.fr :

On va créer dans le répertoire /etc/nginx , un fichier se nommant blockip.conf et on va rentrer dans ce fichier créer DENY et notre adresse IP de notre pc physique :

```
[root@localhost ~]# cd /etc/nginx/
```



On va ajouter dans le fichier de configuration du site_commerce.fr.conf de prendre en compte ce fichier :

```
[root@localhost ~]# cd /etc/nginx/
[root@localhost nginx]# ls
blockip.conf          fastcgi_params          mime.types.default   uwsgi_params
conf.d                fastcgi_params.default  nginx.conf           uwsgi_params.default
default.d             koi-utf                 nginx.conf.default   win-utf
fastcgi.conf          koi-win                 scgi_params
fastcgi.conf.default  mime.types              scgi_params.default
[root@localhost nginx]# cd conf.d/
[root@localhost conf.d]# ls
php-fpm.conf  site_commerce.fr.conf  site_commerce.uk.conf
[root@localhost conf.d]# nano site_commerce.fr.conf
[root@localhost conf.d]#
```

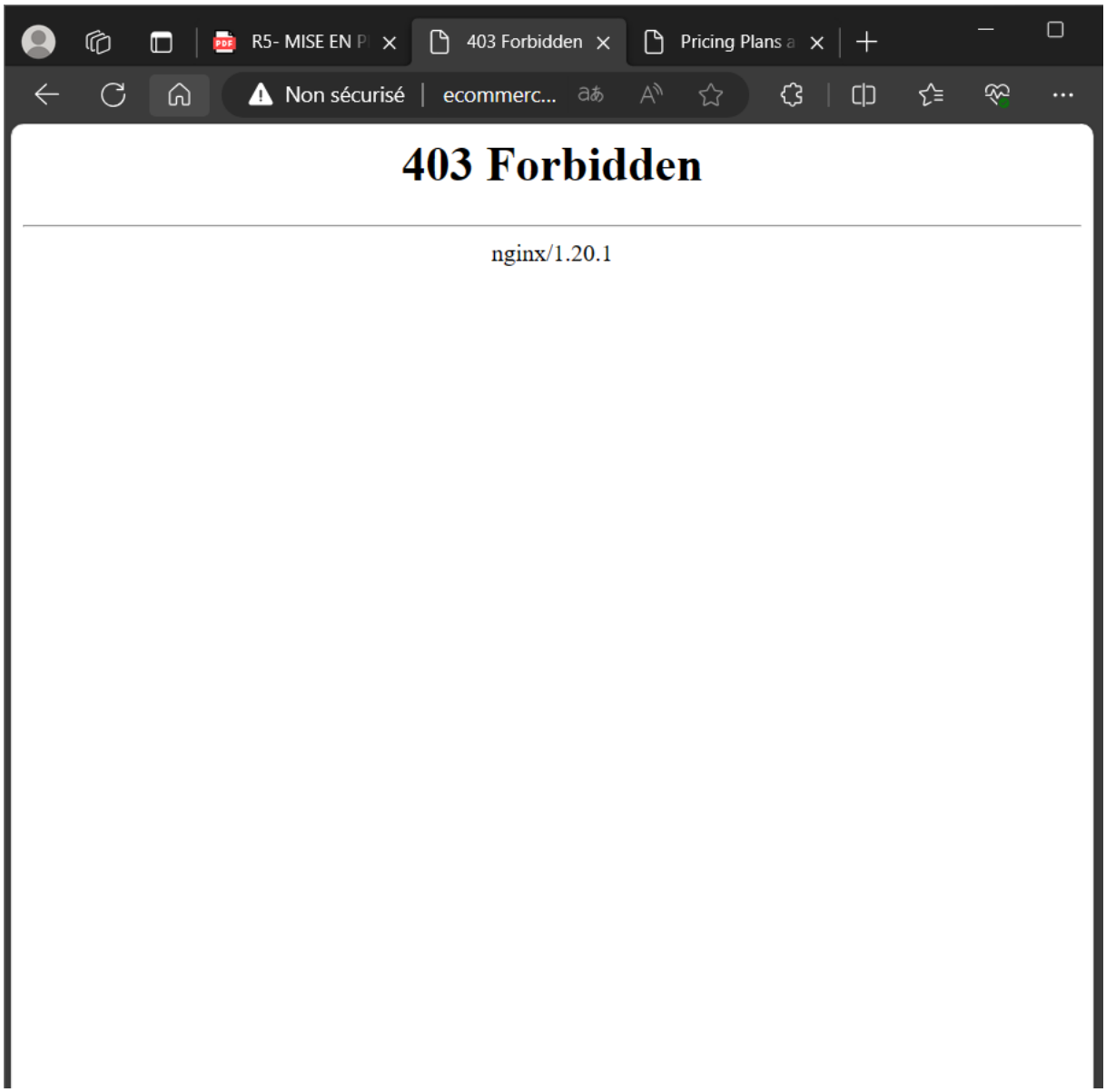
root@localhost:/etc/nginx/conf.d

```
  GNU nano 5.6.1                    site_commerce.fr.conf                    Modified
server {
        listen 80;
        listen [::]:80;


        root /var/www/ecommerce.fr/PricingSubscription/;
        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;
                include blockip.conf;
        }
}
```

On recharge nginx :

```
[root@localhost ~]# systemctl restart nginx
```

Et on remarque que notre adresse est bloqué pour ce site web :



Analyse des logs

Les accès et les erreurs des sites sont présents dans le répertoire : /var/log/nginx. Les erreurs sont dans error.log Les accès dans le fichier access.log :

```
[root@localhost ~]# cd /var/log/nginx/
[root@localhost nginx]# ls
access.log              access_ecommerce.uk.log  error_ecommerce.fr.log
access_ecommerce.fr.log  error.log               error_ecommerce.uk.log
```

Cependant la lecture des logs peut être complexe au début :

```
[root@localhost nginx]# cat access.log
192.168.111.1 - - [09/Nov/2023:09:16:18 +0100] "GET / HTTP/1.1" 200 7620 "-" "Mozilla/5.0 (Wi
ndows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537
.36 Edg/119.0.0.0" "-"
192.168.111.1 - - [09/Nov/2023:09:16:18 +0100] "GET /icons/poweredby.png HTTP/1.1" 200 15443
"http://192.168.111.161/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTM
L, like Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0" "-"
192.168.111.1 - - [09/Nov/2023:09:16:18 +0100] "GET /poweredby.png HTTP/1.1" 200 368 "http://
192.168.111.161/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0" "-"
192.168.111.1 - - [09/Nov/2023:09:16:19 +0100] "GET /favicon.ico HTTP/1.1" 404 3332 "http://1
92.168.111.161/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like G
ecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0" "-"
192.168.111.1 - - [09/Nov/2023:09:26:42 +0100] "GET / HTTP/1.1" 200 7620 "-" "Mozilla/5.0 (Wi
ndows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537
.36 Edg/119.0.0.0" "-"
192.168.111.1 - - [09/Nov/2023:09:26:42 +0100] "GET /icons/poweredby.png HTTP/1.1" 200 15443
"http://192.168.111.161:8080/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0" "-"
192.168.111.1 - - [09/Nov/2023:09:26:42 +0100] "GET /poweredby.png HTTP/1.1" 200 368 "http://
192.168.111.161:8080/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0" "-"
192.168.111.1 - - [09/Nov/2023:09:26:42 +0100] "GET /favicon.ico HTTP/1.1" 404 3332 "http://1
92.168.111.161:8080/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, l
ike Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0" "-"
[root@localhost nginx]#
```

Pour nous simplifier la lecture, nous utiliserons Goacesss , permettant de transformer ces logs dans une interface plus exploitable. Pour cela, il est nécessaire d'activer le repository epel-release. :

```
[root@localhost nginx]# dnf install epel-release -y
Last metadata expiration check: 1:29:08 ago on Thu Nov 16 08:38:12 2023.
Dependencies resolved.
================================================================================
 Package                 Architecture       Version               Repository          Size
================================================================================
Installing:
 epel-release            noarch             9-7.el9               extras              19 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 19 k
Installed size: 26 k
Downloading Packages:
epel-release-9-7.el9.noarch.rpm                          383 kB/s |  19 kB     00:00
--------------------------------------------------------------------------------
Total                                                     55 kB/s |  19 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : epel-release-9-7.el9.noarch                            1/1
  Running scriptlet: epel-release-9-7.el9.noarch                            1/1
Many EPEL packages require the CodeReady Builder (CRB) repository.
It is recommended that you run /usr/bin/crb enable to enable the CRB repository.

  Verifying        : epel-release-9-7.el9.noarch                            1/1

Installed:
  epel-release-9-7.el9.noarch

Complete!
[root@localhost nginx]#
```

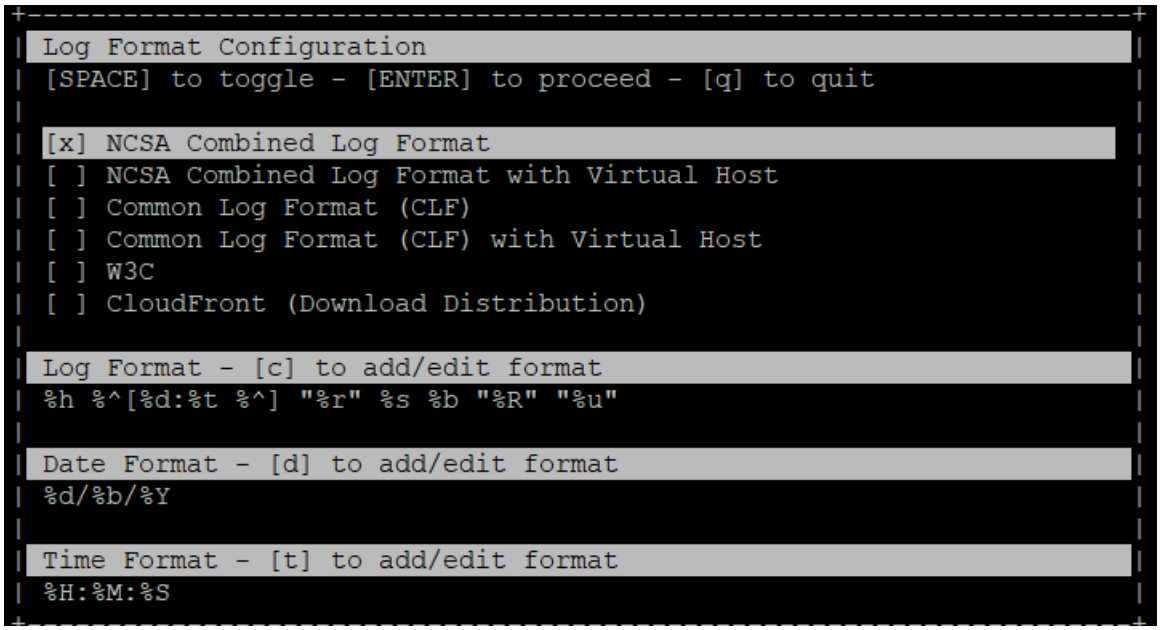
Puis d'installer le package goaccess :

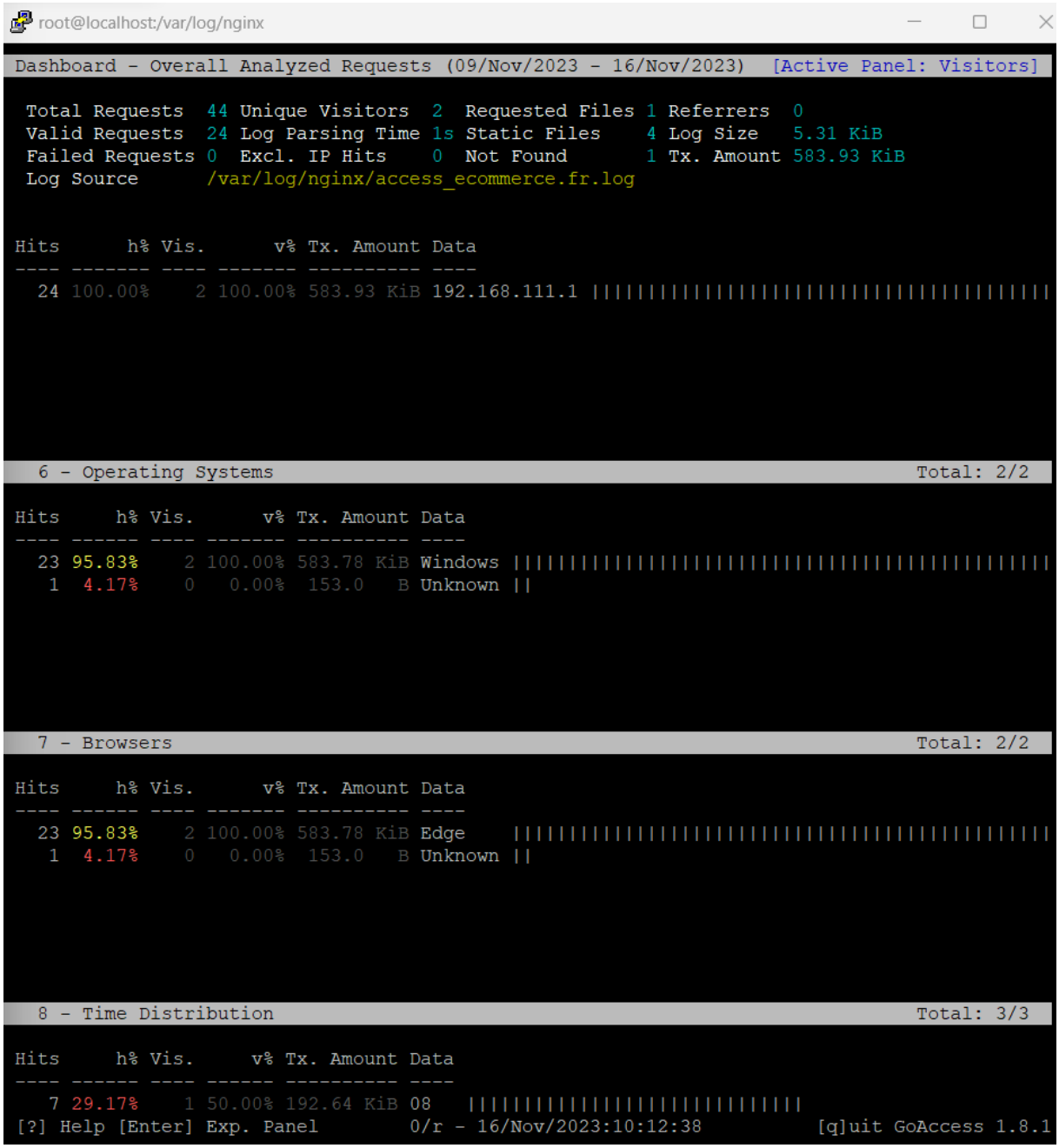
```
[root@localhost nginx]# dnf install goaccess -y
Extra Packages for Enterprise Linux 9 - x86_64                10 MB/s |  19 MB     00:01
Extra Packages for Enterprise Linux 9 openh264 (From Cisco)  1.4 kB/s | 2.5 kB     00:01
Dependencies resolved.
================================================================================
 Package                Architecture      Version                Repository           Size
================================================================================
Installing:
 goaccess               x86_64            1.8.1-1.el9            epel                408 k
Installing dependencies:
 libmaxminddb           x86_64            1.5.2-3.el9            appstream            33 k

Transaction Summary
================================================================================
Install  2 Packages

Total download size: 441 k
Installed size: 1.4 M
Downloading Packages:
(1/2): libmaxminddb-1.5.2-3.el9.x86_64.rpm                   402 kB/s |  33 kB     00:00
(2/2): goaccess-1.8.1-1.el9.x86_64.rpm                       4.4 MB/s | 408 kB     00:00
--------------------------------------------------------------------------------
Total                                                        347 kB/s | 441 kB     00:01
Extra Packages for Enterprise Linux 9 - x86_64               1.6 MB/s | 1.6 kB     00:00
Importing GPG key 0x3228467C:
 Userid     : "Fedora (epel9) <epel@fedoraproject.org>"
 Fingerprint: FF8A D134 4597 106E CE81 3B91 8A38 72BF 3228 467C
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-9
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : libmaxminddb-1.5.2-3.el9.x86_64                        1/2
  Installing       : goaccess-1.8.1-1.el9.x86_64                            2/2
  Running scriptlet: goaccess-1.8.1-1.el9.x86_64                            2/2
  Verifying        : goaccess-1.8.1-1.el9.x86_64                            1/2
  Verifying        : libmaxminddb-1.5.2-3.el9.x86_64                        2/2

Installed:
  goaccess-1.8.1-1.el9.x86_64              libmaxminddb-1.5.2-3.el9.x86_64

Complete!
[root@localhost nginx]#
```

Pour utiliser le goaccess, il faut exécuter la commande ci-contre :

```
[root@localhost nginx]# goaccess -f /var/log/nginx/access_ecommerce.fr.log
```

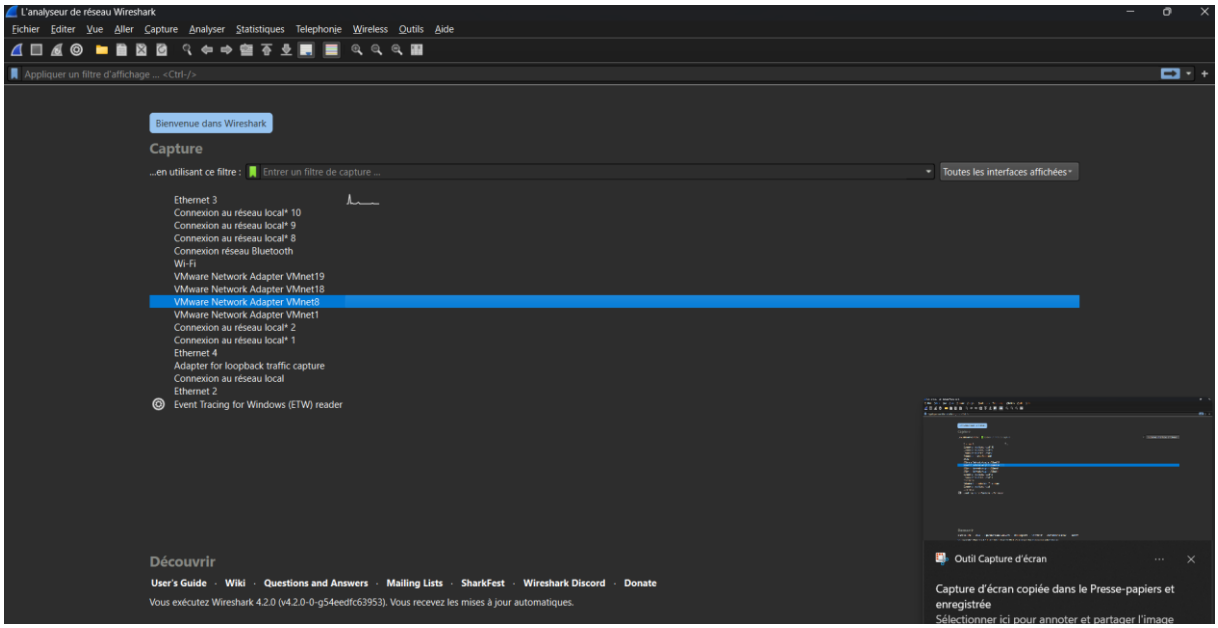
Vous aurez une interface, il faut sélectionner le premier format et faire entrer :


Log Format Configuration
[SPACE] to toggle - [ENTER] to proceed - [q] to quit
[x] NCSA Combined Log Format
[ ] NCSA Combined Log Format with Virtual Host
[ ] Common Log Format (CLF)
[ ] Common Log Format (CLF) with Virtual Host
[ ] W3C
[ ] CloudFront (Download Distribution)
Log Format - [c] to add/edit format
%h %^[%d:%t %^] "%r" %s %b "%R" "%u"
Date Format - [d] to add/edit format
%d/%b/%Y
Time Format - [t] to add/edit format
%H:%M:%S

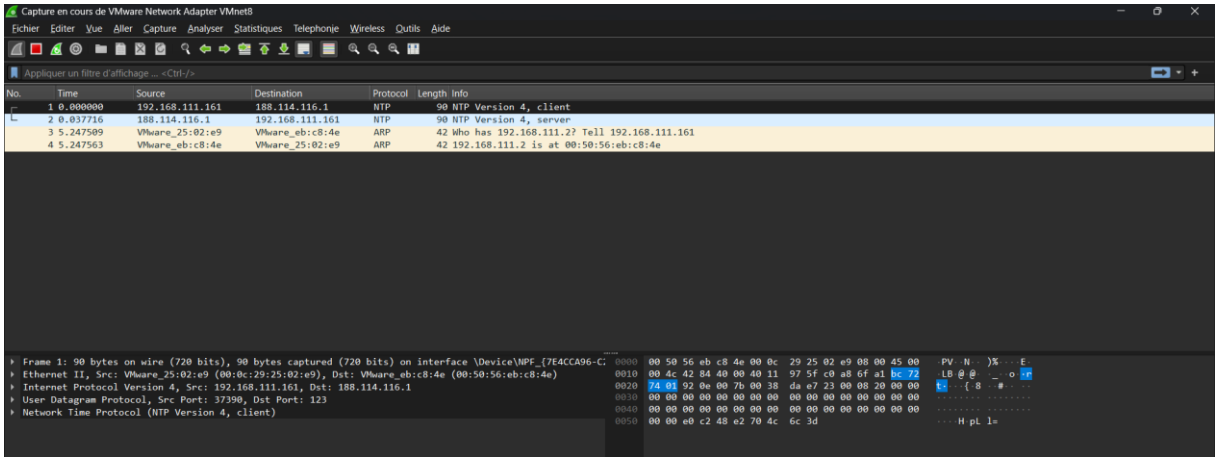


Analyse du site en http

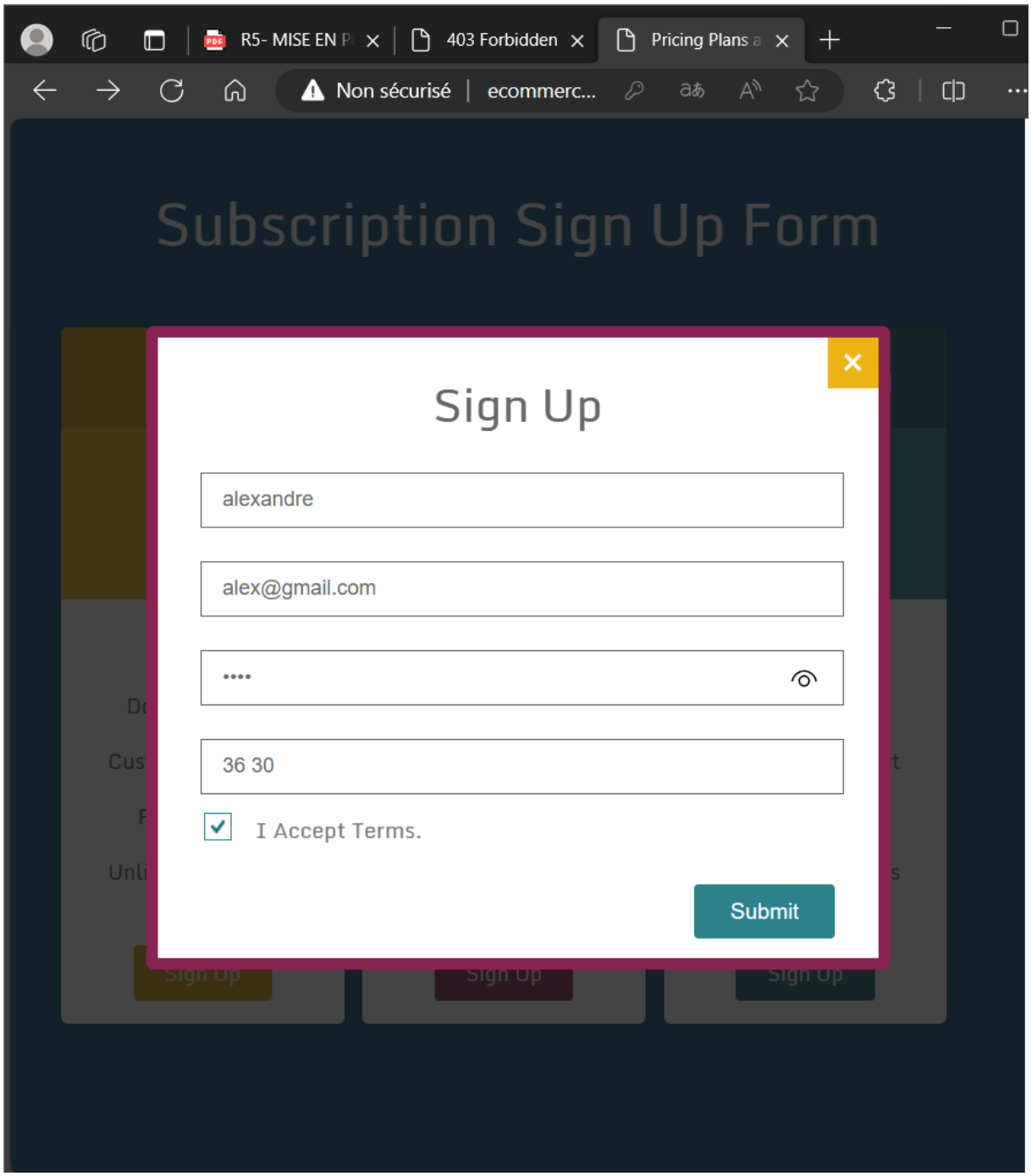
On lance wireshark on sélectionne notre carte réseau :



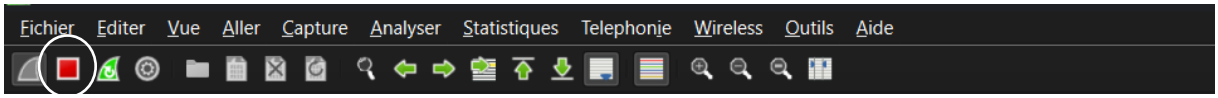
On entre sur cette page :



Je fais une simulation d'inscription :

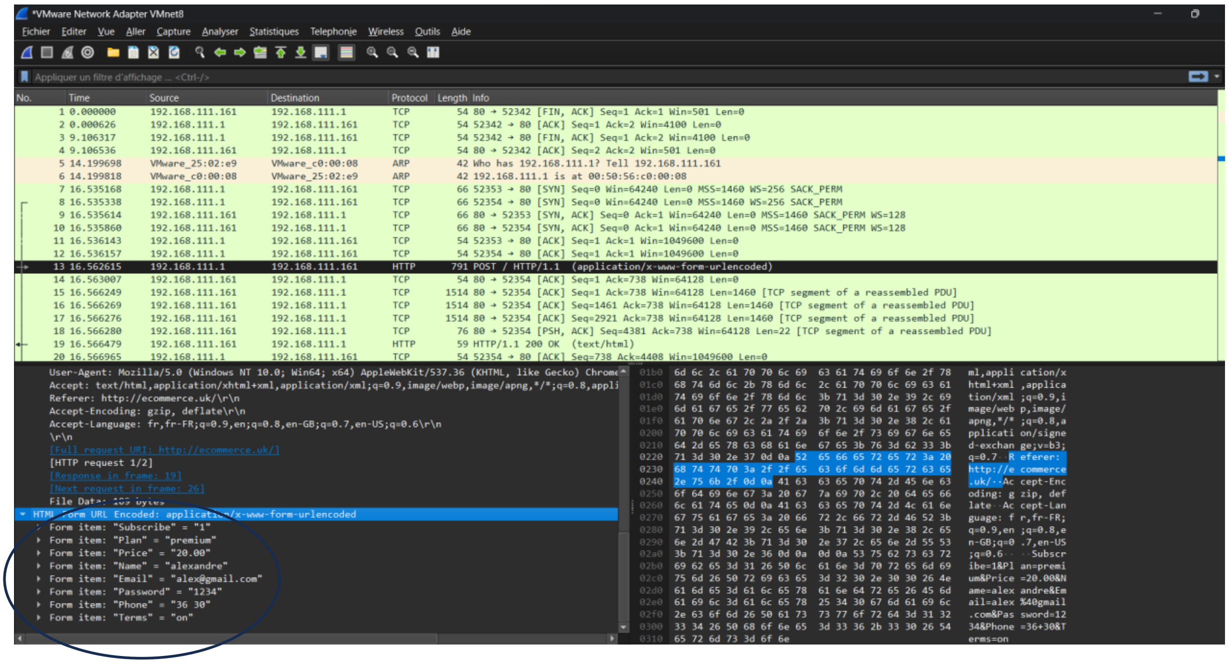


Ensuite avec wireshark on peut récupérer les données que l'on a rentré en appuyant sur stop :



LESPAGNOL
Alexandre
2SISR

09/11/2023



Il est donc indispensable de changer cela et de sécuriser cela notamment avec la mise en place du protocole https !

Sécurisation d’un service WEB (nginx)

On va créer les répertoires : mkdir /etc/ssl/private :

```
[root@localhost ~]# mkdir /etc/ssl/private
[root@localhost ~]# cd /etc/ssl/private
```

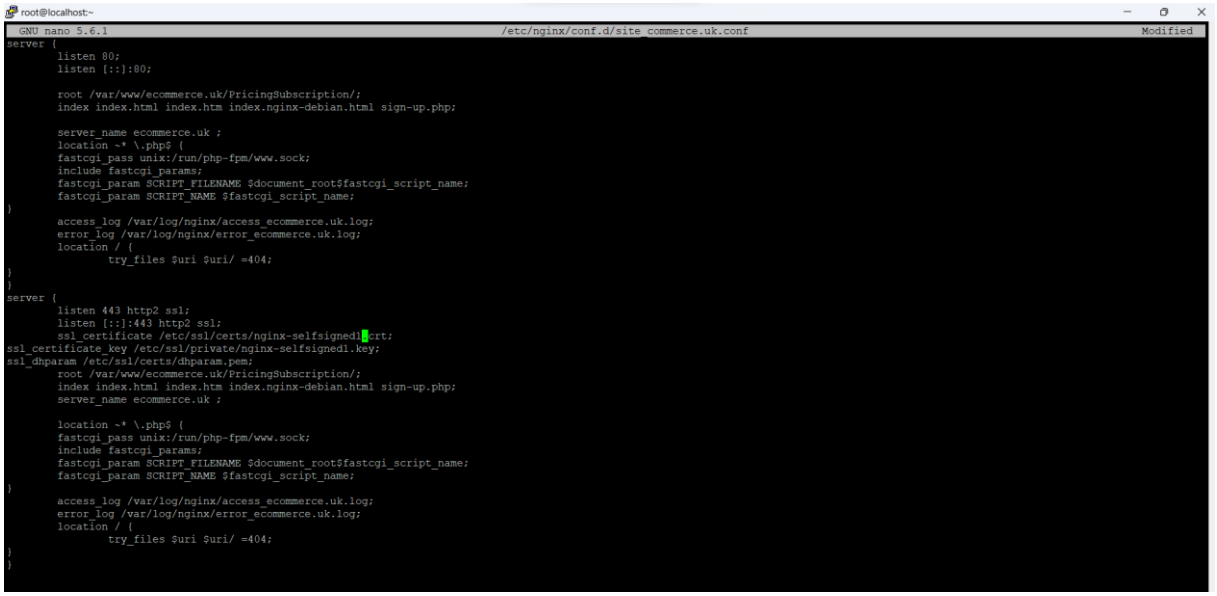
On va limiter les accès sur ce répertoire :

```
[root@localhost ~]# chmod 700 /etc/ssl/private
```

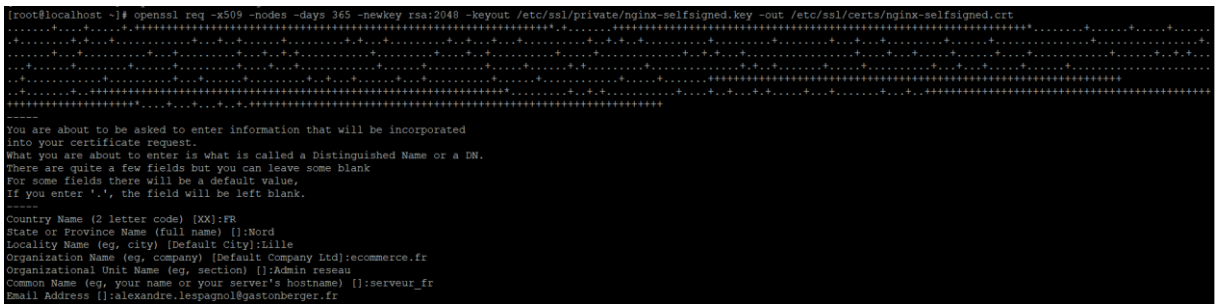
Création d’un certificat (uk) :

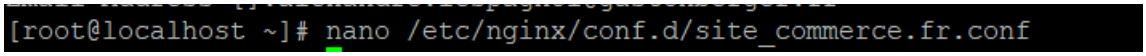
```
[root@localhost ~]# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned1.key -out /etc/ssl/certs/nginx-selfsigned1.crt
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter 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.
-----
Country Name (2 letter code) [XX]:UK
State or Province Name (full name) []:Manchester
Locality Name (eg, city) [Default City]:Manchester
Organization Name (eg, company) [Default Company Ltd]:ecommerce.uk
Organizational Unit Name (eg, section) []:Network manager
Common Name (eg, your name or your server's hostname) []:serveur_uk
Email Address []:alexandre.lespagnol@gastonberger.fr
```

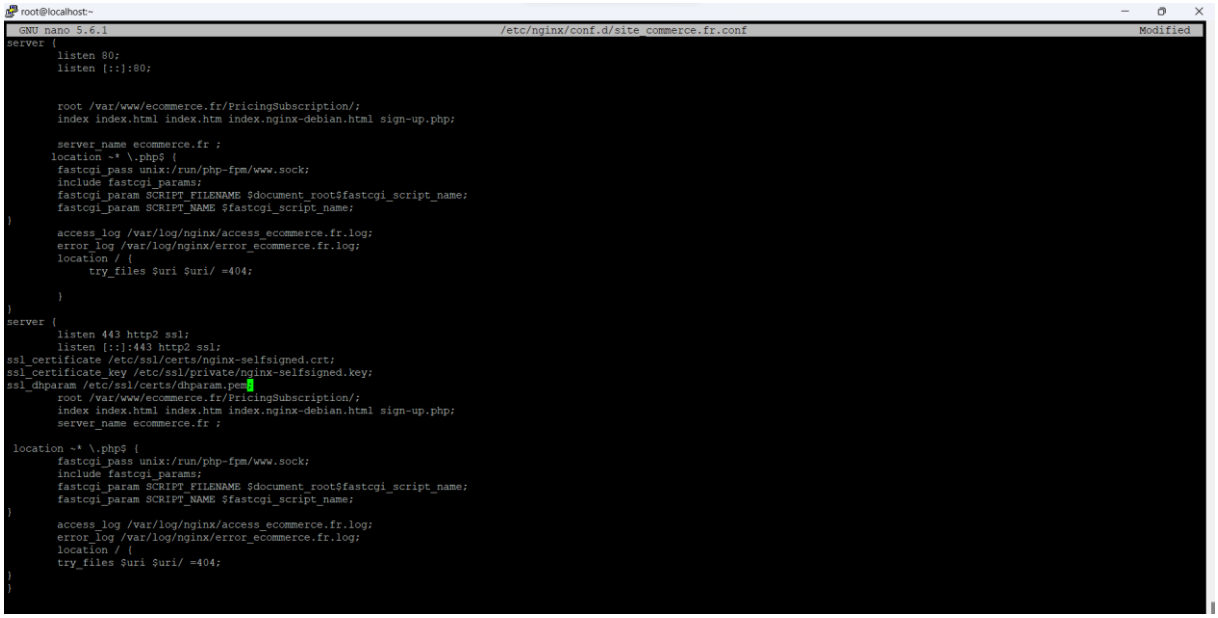
```
[root@localhost ~]# nano /etc/nginx/conf.d/site_commerce.uk.conf
```



Création d'un certificat (fr) :

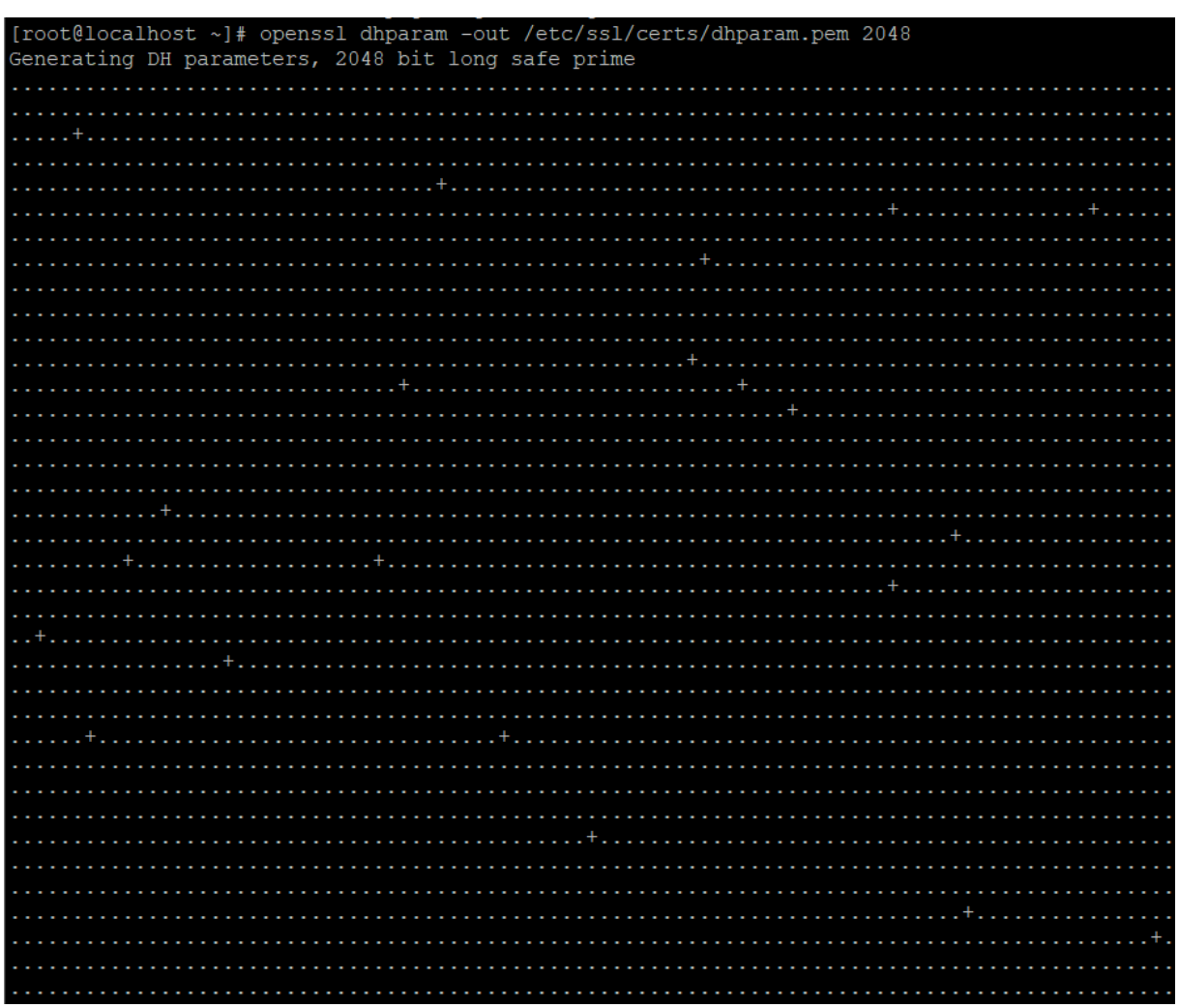






Nous allons maintenant générer un un groupe DiffieHellman fort qui permet la négociation de PFS (Perfact Forward Secrecy) :

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



Nous allons maintenant éditer nos fichiers de configuration (/etc/nginx/conf .d) de chaque site pour intégrer le protocole https. :

On va spécifier les ports de fonctionnement de https (443) :

Puis spécifier notre certificat et notre clé :

```
[root@localhost ~]# cd /etc/nginx/
[root@localhost nginx]# ls
blockip.conf          fastcgi_params          mime.types.default   uwsgi_params
conf.d                fastcgi_params.default  nginx.conf           uwsgi_params.default
default.d             koi-utf                 nginx.conf.default   win-utf
fastcgi.conf          koi-win                 scgi_params
fastcgi.conf.default  mime.types              scgi_params.default
[root@localhost nginx]# cd conf.d/
[root@localhost conf.d]# ls
php-fpm.conf  site_commerce.fr.conf  site_commerce.uk.conf
[root@localhost conf.d]# nano site_commerce.fr.conf
```

root@localhost:/etc/nginx/conf.d

```
  GNU nano 5.6.1                    site_commerce.fr.conf                    Modified
server {
        listen 80;
        listen [::]:80;


        root /var/www/ecommerce.fr/PricingSubscription/;
        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;

        }
}
server {
        listen 443 http2 ssl;
        listen [::]:443 http2 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/ecommerce.fr/PricingSubscription/;
        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;
}
}
```

On fait ceci pour nos deux sites internet :

```
[root@localhost ~]# cd /etc/nginx/
[root@localhost nginx]# ls
blockip.conf          fastcgi_params          mime.types.default   uwsgi_params
conf.d                fastcgi_params.default  nginx.conf           uwsgi_params.default
default.d             koi-utf                 nginx.conf.default   win-utf
fastcgi.conf          koi-win                 scgi_params
fastcgi.conf.default  mime.types              scgi_params.default
[root@localhost nginx]# cd conf.d/
[root@localhost conf.d]# ls
php-fpm.conf  site_commerce.fr.conf  site_commerce.uk.conf
[root@localhost conf.d]# nano site_commerce.fr.conf
[root@localhost conf.d]# nano /etc/nginx/conf.d/site_commerce.uk.conf
```

root@localhost:~

```
 GNU nano 5.6.1              /etc/nginx/conf.d/site_commerce.uk.conf              Modified
server {
        listen 80;
        listen [::]:80;

        root /var/www/ecommerce.uk/PricingSubscription/;
        index index.html index.htm index.nginx-debian.html sign-up.php;

        server_name ecommerce.uk ;
        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.uk.log;
        error_log /var/log/nginx/error_ecommerce.uk.log;
        location / {
                try_files $uri $uri/ =404;
}
}
server {
        listen 443 http2 ssl;
        listen [::]:443 http2 ssl;
ssl_certificate /etc/ssl/certs/nginx-selfsigned1.crt;
  ssl_certificate_key /etc/ssl/private/nginx-selfsigned1.key;
  ssl_dhparam /etc/ssl/certs/dhparam.pem;
        root /var/www/ecommerce.uk/PricingSubscription/;
        index index.html index.htm index.nginx-debian.html sign-up.php;
        server_name ecommerce.uk ;

        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.uk.log;
        error_log /var/log/nginx/error_ecommerce.uk.log;
        location / {
                try_files $uri $uri/ =404;
}
}
```

root@localhost:/etc/nginx/conf.d

```
  GNU nano 5.6.1              /etc/nginx/conf.d/site_commerce.uk.conf              Modified
server {
        listen 80;
        listen [::]:80;

        root /var/www/ecommerce.uk/PricingSubscription/;
        index index.html index.htm index.nginx-debian.html sign-up.php;

        server_name ecommerce.uk ;
        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.uk.log;
        error_log /var/log/nginx/error_ecommerce.uk.log;
        location / {
                try_files $uri $uri/ =404;
}
}
server {
        listen 443 http2 ssl;
        listen [::]:443 http2 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/ecommerce.uk/PricingSubscription/;
        index index.html index.htm index.nginx-debian.html sign-up.php;
        server_name ecommerce.uk ;

        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.uk.log;
        error_log /var/log/nginx/error_ecommerce.uk.log;
        location / {
                try_files $uri $uri/ =404;
}
}
```

Avant de relancer le service, on teste nos fichiers : Nginx -t

```
[root@localhost ~]# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

```
[root@localhost ~]# systemctl restart nginx
```

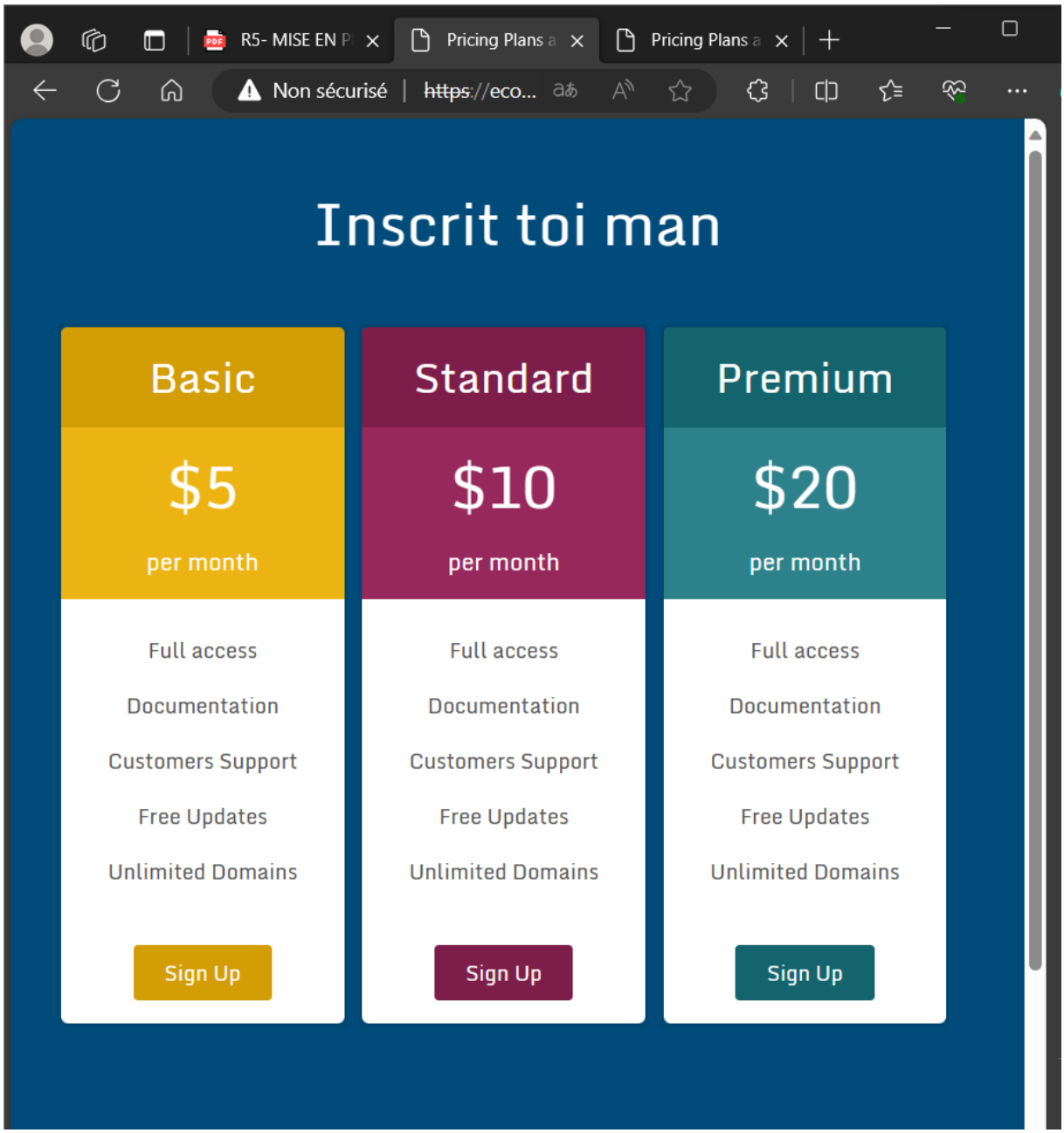
On ouvre le firewall sur le port 443 :

```
[root@localhost ~]# firewall-cmd --add-port=443/tcp  --permanent
success
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client http ssh
  ports: 8080/tcp 443/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

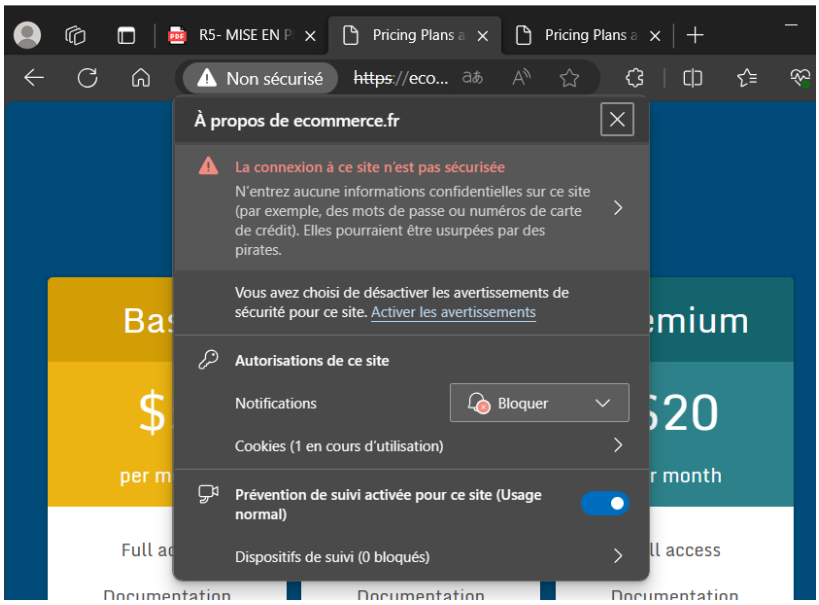
Quand on tente d'accéder au site, nous avons une alerte : Celle-ci est normale car depuis 2019, les navigateurs signalent les certificats « auto-signés » Faire « avancer / Accepter le risque et poursuivre ». :

Certificat (fr) :

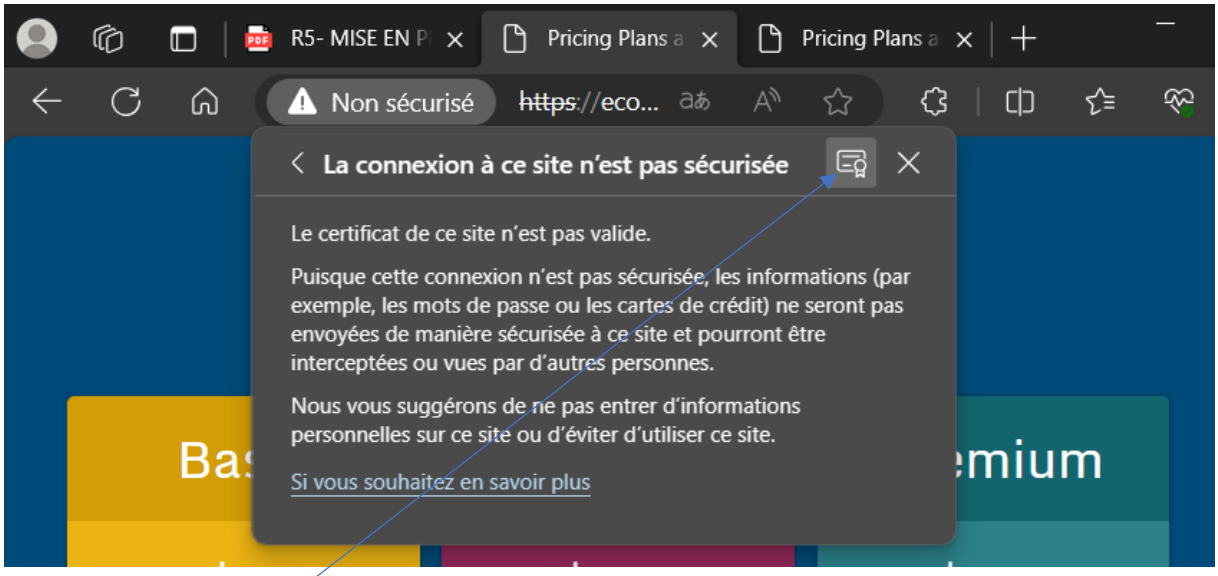
On tape dans notre navigateur https://ecommerce.fr



On clique sur non sécuriser :

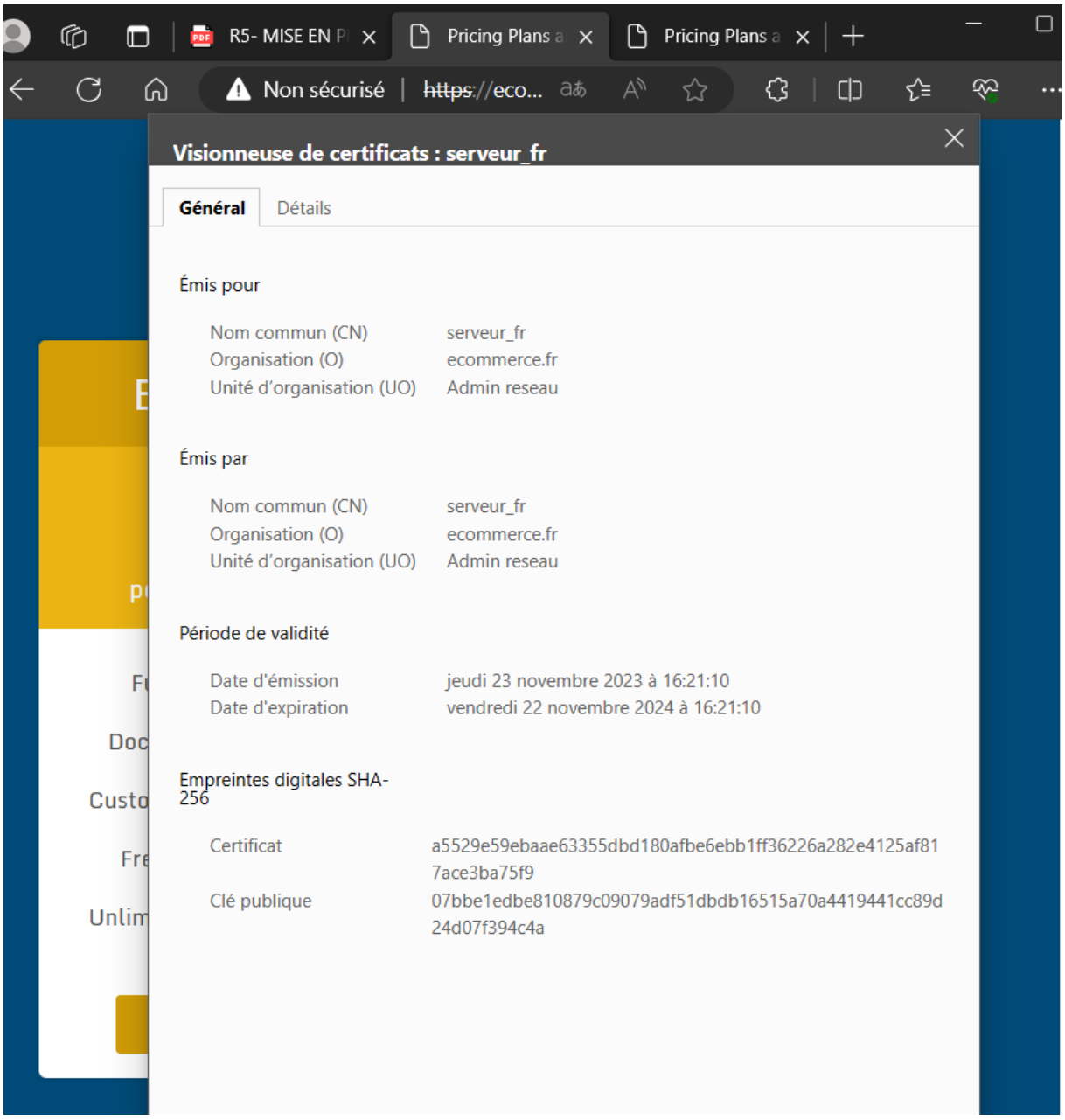


Ensuite sur la connexion à ce site n'est pas sécurisée :



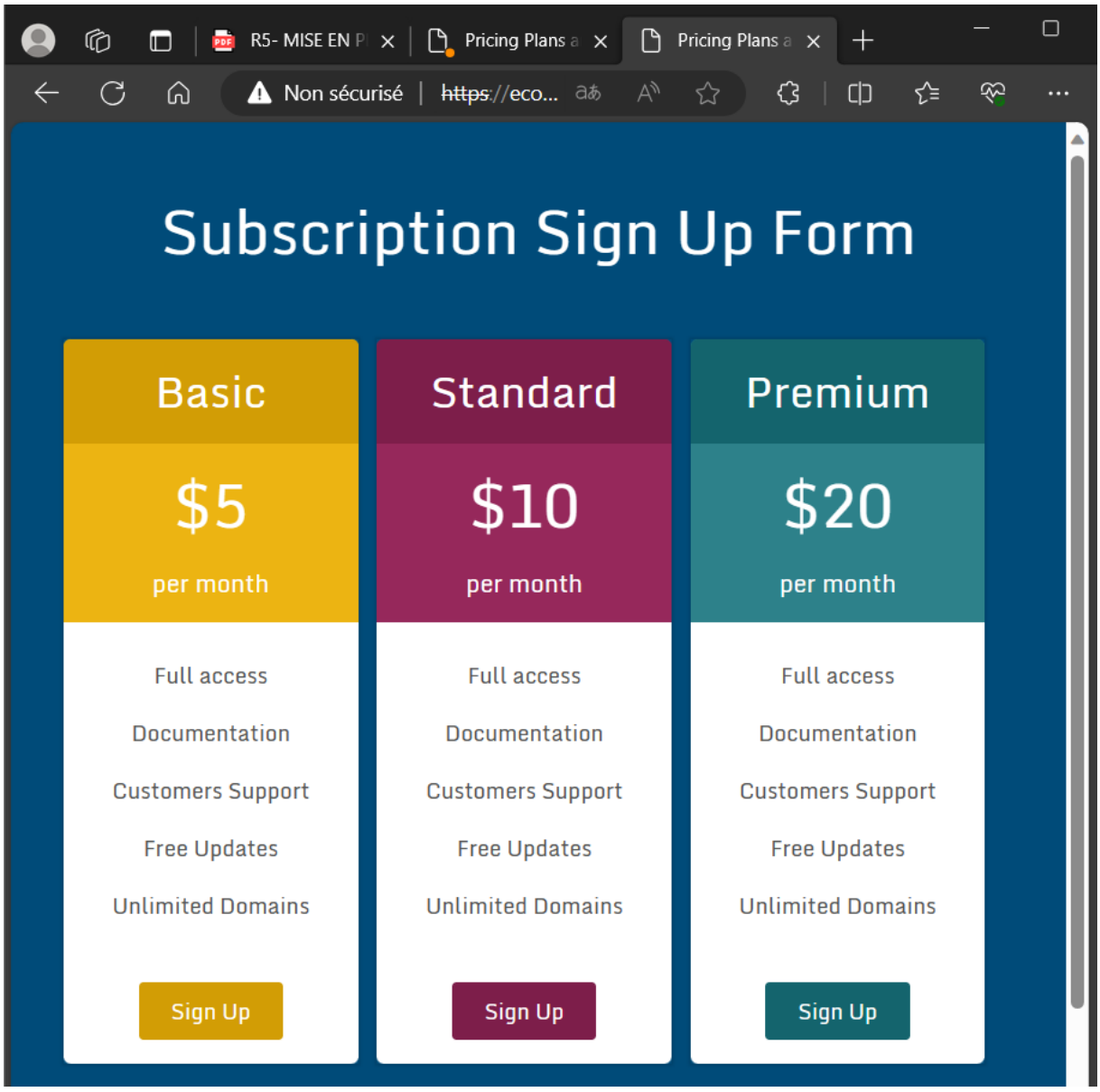
Ensuite on clique sur ça

Et on remarque bien notre certificat fr :

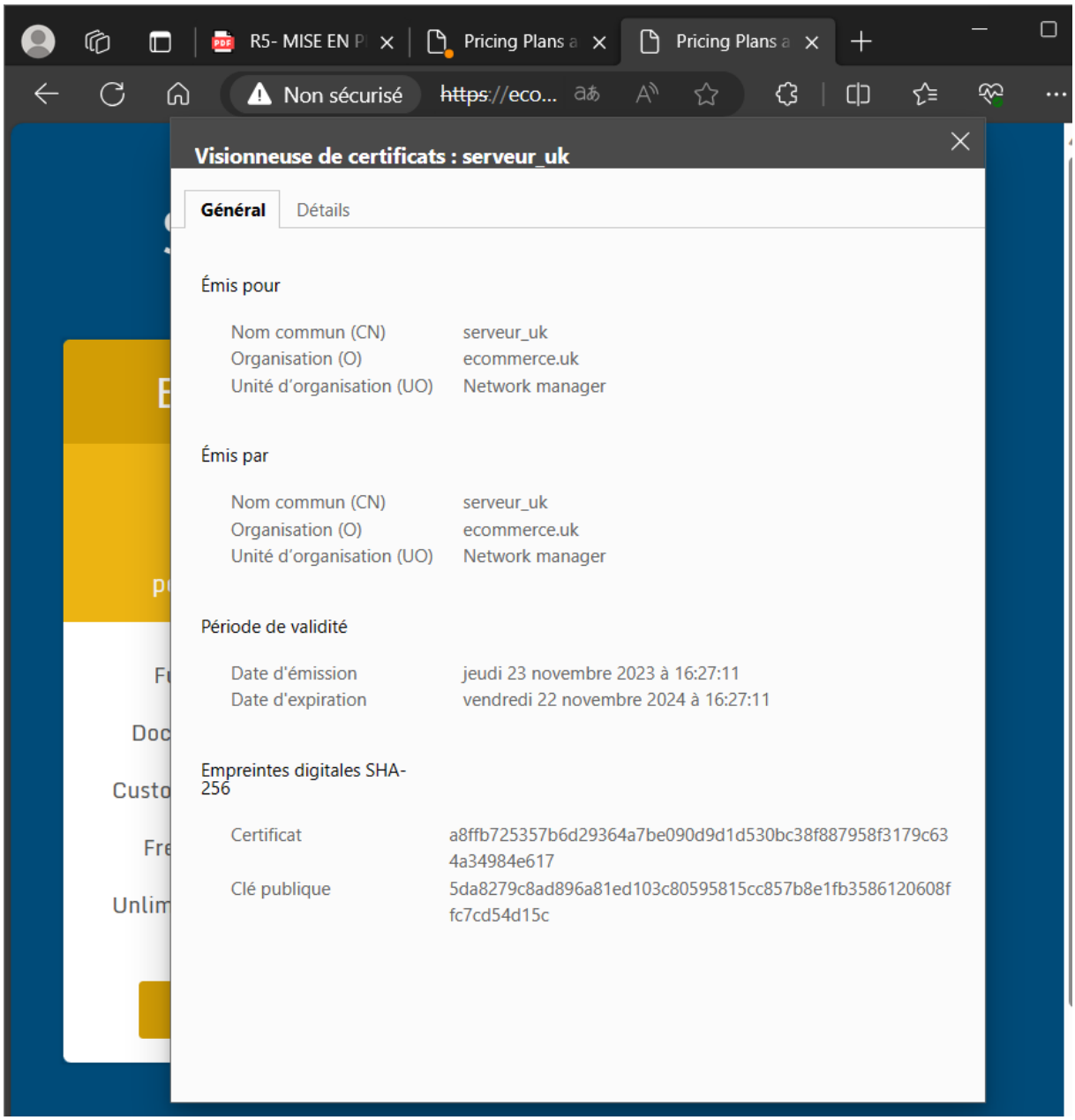


On vérifie également par le même principe mais pour le site uk :

En tapant dans notre navigateur : https://ecommerce.uk

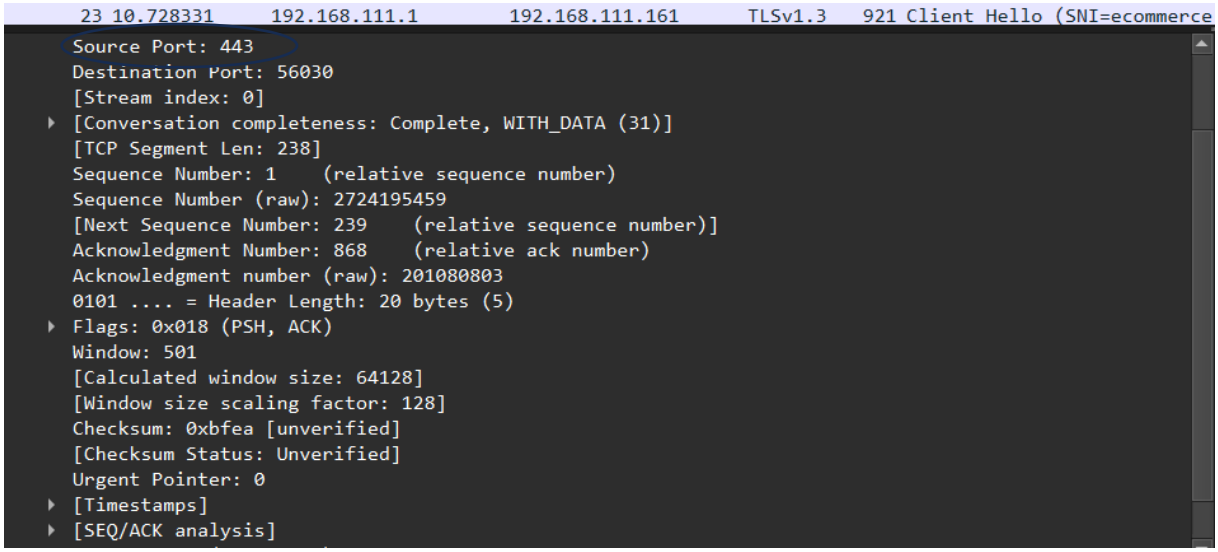


Et on remarque que notre certificat (uk) à bien été pris en compte :



Comment être sûr du bon fonctionnement en https ?

On lance, une petite analyse de trame via WireShark En faisant le même exercice, on remarque, les données sont chiffrées :



On remarque bien qu'on est sur le port 443 donc https.

Redirection http-https

Cependant, si nous testons d'aller sur le site en http, cela fonctionne ! Nous allons mettre en place pour nos utilisateurs, une redirection http vers https. Il faut pour cela ajouter une directive dans la partie serveur http : L'ensemble des requêtes http sont redirigés vers HTTPS :

On edit notre fichier de conf pour nos deux site ecommerce :

FR :

Et on rajoute cette ligne dans les deux fichiers de conf :

root@localhost:/etc/nginx/conf.d

```
 GNU nano 5.6.1                    site_commerce.fr.conf                    Modified
        listen 80;
        listen [::]:80;


        root /var/www/ecommerce.fr/PricingSubscription/;
        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;

        }
}
server {
        listen 443 http2 ssl;
        listen [::]:443 http2 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/ecommerce.fr/PricingSubscription/;
        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 $uri/ =404;
       return 301 https://$host$request_uri/;
}
}
```

UK :

```
[root@localhost ~]# nano /etc/nginx/conf.d/site_commerce.uk.conf
```

root@localhost:~

```
  GNU nano 5.6.1              /etc/nginx/conf.d/site_commerce.uk.conf              Modified
server {
        listen 80;
        listen [::]:80;

        root /var/www/ecommerce.uk/PricingSubscription/;
        index index.html index.htm index.nginx-debian.html sign-up.php;

        server_name ecommerce.uk ;
        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.uk.log;
        error_log /var/log/nginx/error_ecommerce.uk.log;
        location / {
                try_files $uri $uri/ =404;
}
}
server {
        listen 443 http2 ssl;
        listen [::]:443 http2 ssl;
        ssl_certificate /etc/ssl/certs/nginx-selfsigned1.crt;
ssl_certificate_key /etc/ssl/private/nginx-selfsigned1.key;
ssl_dhparam /etc/ssl/certs/dhparam.pem;
        root /var/www/ecommerce.uk/PricingSubscription/;
        index index.html index.htm index.nginx-debian.html sign-up.php;
        server_name ecommerce.uk ;

        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.uk.log;
        error_log /var/log/nginx/error_ecommerce.uk.log;
        location / {
                try_files $uri $uri/ =404;
        return 301 https://$host$request_uri/;
}
}
```

L’ensemble des requêtes http sont redirigés vers HTTPS

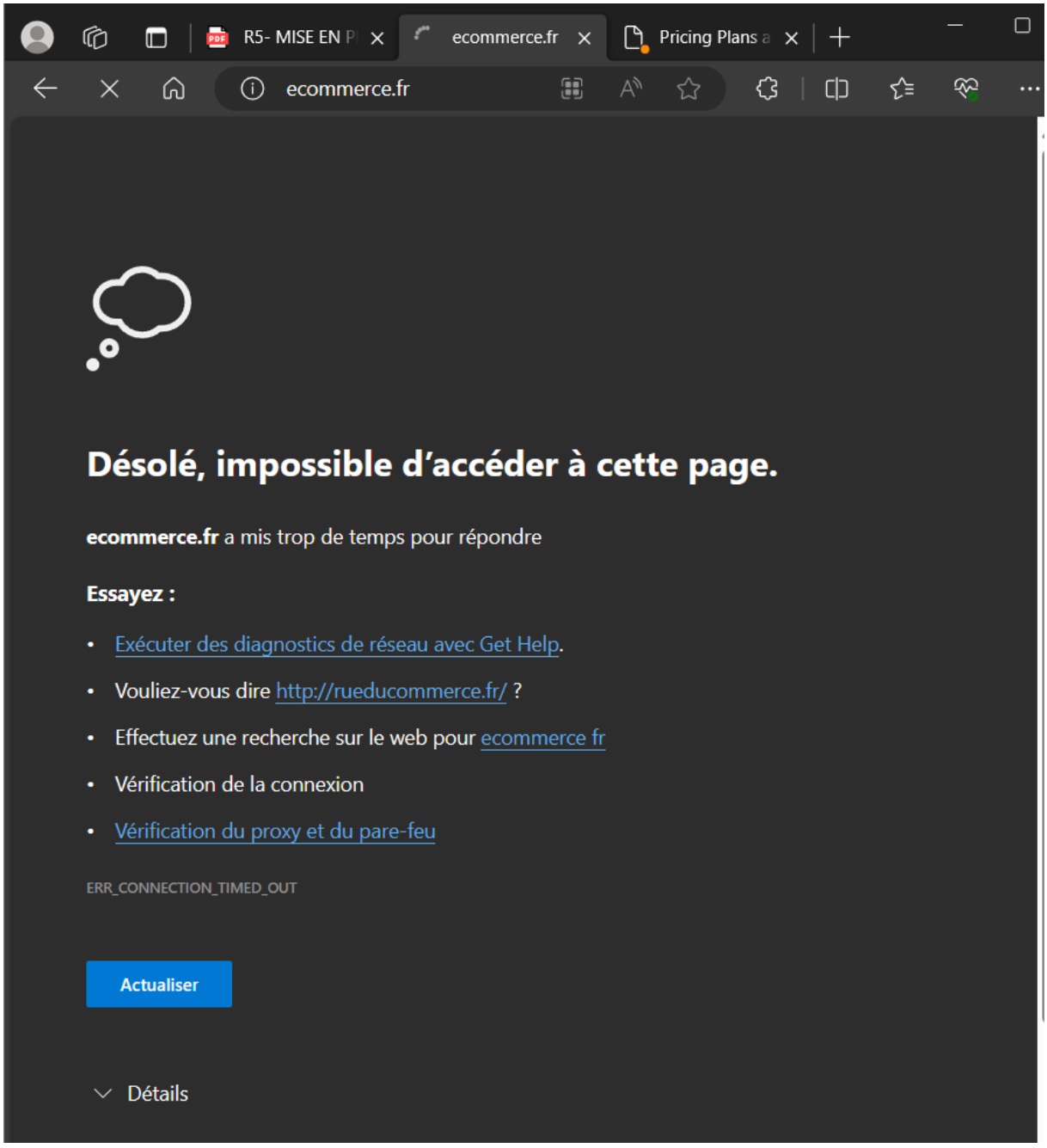
Il ne faudra pas oublier d’ouvrir le port https pour que cela fonctionne :

```
[root@localhost ~]# firewall-cmd --remove-service=http
success
```

On voit bien que le port 443 est bien ouvert :

```
[root@localhost ~]# firewall-cmd --list-port
443/tcp 8080/tcp
```

On remarque bien que maintenant en http on arrive plus à accéder au site :



Installation de deux serveurs red hat

Création d'une vm service_web02 à l'identique que service_web et et d'une VM HA01.

LESPAGNOL
Alexandre
2SISR

Installation d’haproxy sur la machine HA01 :

```
[root@localhost ~]# dnf install haproxy -y
Rocky Linux 9 - BaseOS                                                    11 kB/s | 4.1 kB     00:00
Rocky Linux 9 - AppStream                                                 16 kB/s | 4.5 kB     00:00
Rocky Linux 9 - Extras                                                   6.9 kB/s | 2.9 kB     00:00
Dependencies resolved.
=======================================================================================================
 Package            Architecture         Version                 Repository                      Size
=======================================================================================================
Installing:
 haproxy            x86_64               2.4.22-1.el9            appstream                      2.2 M

Transaction Summary
=======================================================================================================
Install  1 Package

Total download size: 2.2 M
Installed size: 6.6 M
Downloading Packages:
haproxy-2.4.22-1.el9.x86_64.rpm                                          7.5 MB/s | 2.2 MB     00:00
-------------------------------------------------------------------------------------------------------
Total                                                                    3.9 MB/s | 2.2 MB     00:00
Rocky Linux 9 - AppStream                                                1.5 MB/s | 1.7 kB     00:00
Importing GPG key 0x350D275D:
 Userid     : "Rocky Enterprise Software Foundation - Release key 2022 <releng@rockylinux.org>"
 Fingerprint: 21CB 256A E16F C54C 6E65 2949 702D 426D 350D 275D
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-Rocky-9
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                              1/1
  Running scriptlet: haproxy-2.4.22-1.el9.x86_64                                                  1/1
  Installing       : haproxy-2.4.22-1.el9.x86_64                                                  1/1
  Running scriptlet: haproxy-2.4.22-1.el9.x86_64                                                  1/1
[  745.681836] systemd-rc-local-generator[1485]: /etc/rc.d/rc.local is not marked executable, skipping.
  Verifying        : haproxy-2.4.22-1.el9.x86_64                                                  1/1

Installed:
  haproxy-2.4.22-1.el9.x86_64

Complete!
[root@localhost ~]#
```

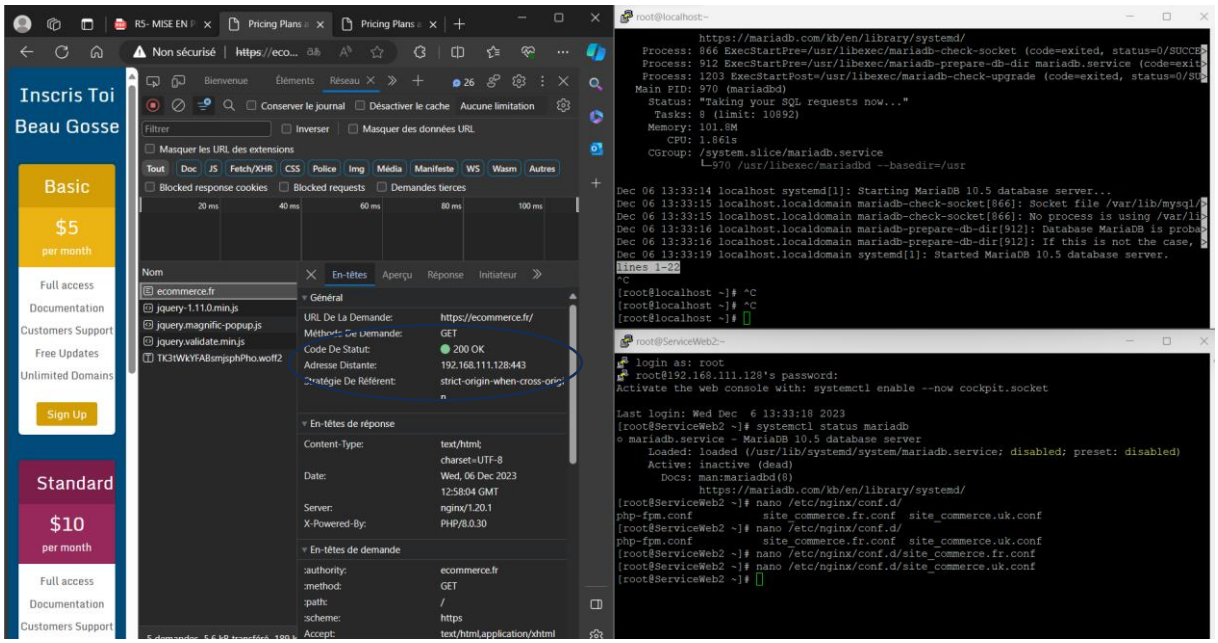
Après avoir fait cela on regarde si avec notre service_web02 on arrive bien à se connecter au site ecommerce.fr et ecommerce.uk :

On change dans notre fichier hosts dans notre c de notre pc dans windows, system32,drivers,etc,hosts :

Et on met en commentaire l’adresse ip de service_web qui va pour ecommerce.fr et ecommerce.uk et on ajoute l’adresse ip de service_web02 pour ecommerce.fr et ecommerce.uk en mode administration bien sur :

```
#       192.168.111.161 ecommerce.fr
#       192.168.111.161 ecommerce.uk
#       172.16.64.227 wiki.berlingastonmedilab.com
        192.168.111.128 ecommerce.fr
        192.168.111.128 ecommerce.uk
```

Ensuite en tapant ecommerce.fr et ecommer.uk on remarque bien que c notre service_web02 qui nous donne la page :



Configuration BDD

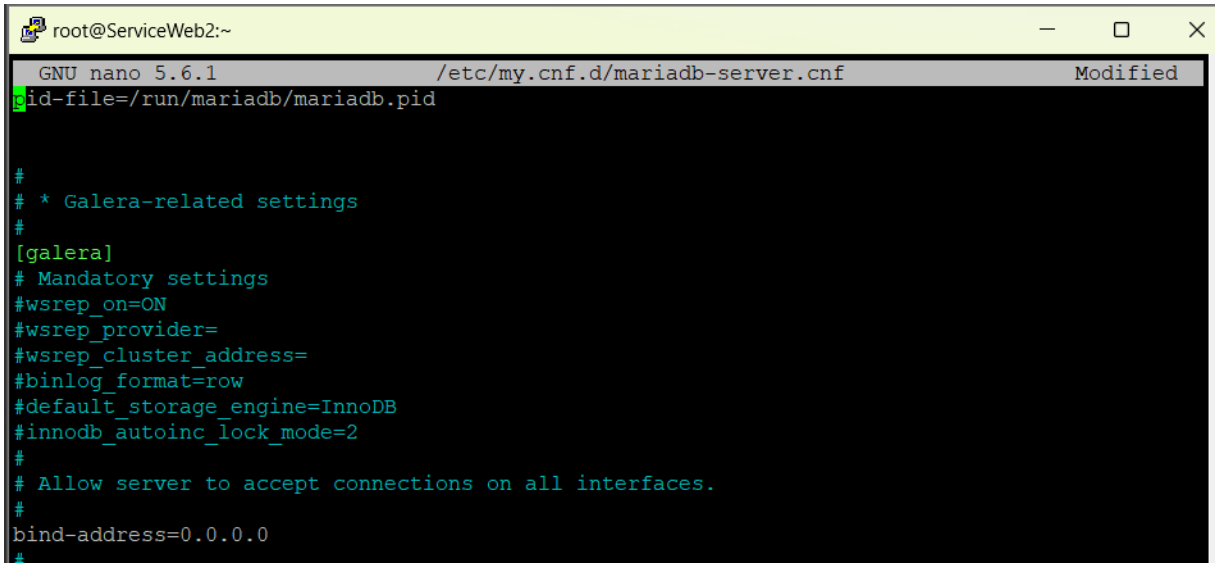
Suite à notre modification d'infrastructure, nous devons autoriser les connexions externes au moteur BD. Pour cela, il va falloir faire sur WEB01, les actions ci-contre :

Ouvrir le firewall sur le port 3306/tcp :

```
root@localhost:~
[root@localhost ~]# firewall-cmd --add-port=3306/tcp --permanent
success
[root@localhost ~]# firewall-cmd reload
usage: see firewall-cmd man page
firewall-cmd: error: unrecognized arguments: reload
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client http ssh
  ports: 8080/tcp 443/tcp 3306/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost ~]#
```

configuration du moteur BDD pour autoriser les connexions extérieures dans le fichiers /etc/my.cnf.d/mariadbserver.cnf et décommenter la ligne bind-address=0.0.0.0 :

```
[root@ServiceWeb2 ~]# nano /etc/my.cnf.d/mariadb-server.cnf
```




```
[root@localhost ~]# systemctl restart mariadb
```

Puis ensuite se connecter sur le moteur BDD dans service_web et attribuer à notre utilisateur USER-web (dans mon cas) l'autorisation que cet utilisateur puisse se connecter depuis la machine web02 sur la base de donnée de WEB01. GRANT ALL on fr.* to exploit_fr@'IP' IDENTIFIED BY 'exploit'; :

```
MariaDB [(none)]> GRANT ALL on web.* to 'USER-web'@'192.168.111.128' IDENTIFIED BY 'web';
Query OK, 0 rows affected (0.002 sec)
```

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

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

MariaDB [(none)]> GRANT ALL on webuk.* to 'USERWEBUK'@'192.168.111.128' IDENTIFIED BY 'webuk'
;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)
```

```
[root@localhost ~]# systemctl restart mariadb
```

On modifie aussi l'adresse ip dans ce fichier en mettant celle de notre service_web :

```
[root@ServiceWeb2 ~]# cd /var/www
[root@ServiceWeb2 www]# ls
cgi-bin  ecommerce.fr  ecommerce.uk  html
[root@ServiceWeb2 www]# cd ecommerce.fr
[root@ServiceWeb2 ecommerce.fr]# ls
PricingSubscription
[root@ServiceWeb2 ecommerce.fr]# cd PricingSubscription
[root@ServiceWeb2 PricingSubscription]# LS
-bash: LS: command not found
[root@ServiceWeb2 PricingSubscription]# ls
config.php  css  database.sql  images  js  sign-up.php
[root@ServiceWeb2 PricingSubscription]# nano c
config.php  css/
[root@ServiceWeb2 PricingSubscription]# nano config.php
```

root@ServiceWeb2:/var/www/ecommerce.fr/PricingSubscription

```
 GNU nano 5.6.1                    config.php                    Modifie
<?php
$SETTINGS["mysql_user"]='USER-web';
$SETTINGS["mysql_pass"]='web';
$SETTINGS["hostname"]='192.168.111.161';
$SETTINGS["mysql_database"]='web';
$SETTINGS["data_table"]='registrations';
$SETTINGS["paypal_address"]='email@domain.com';
?>
```

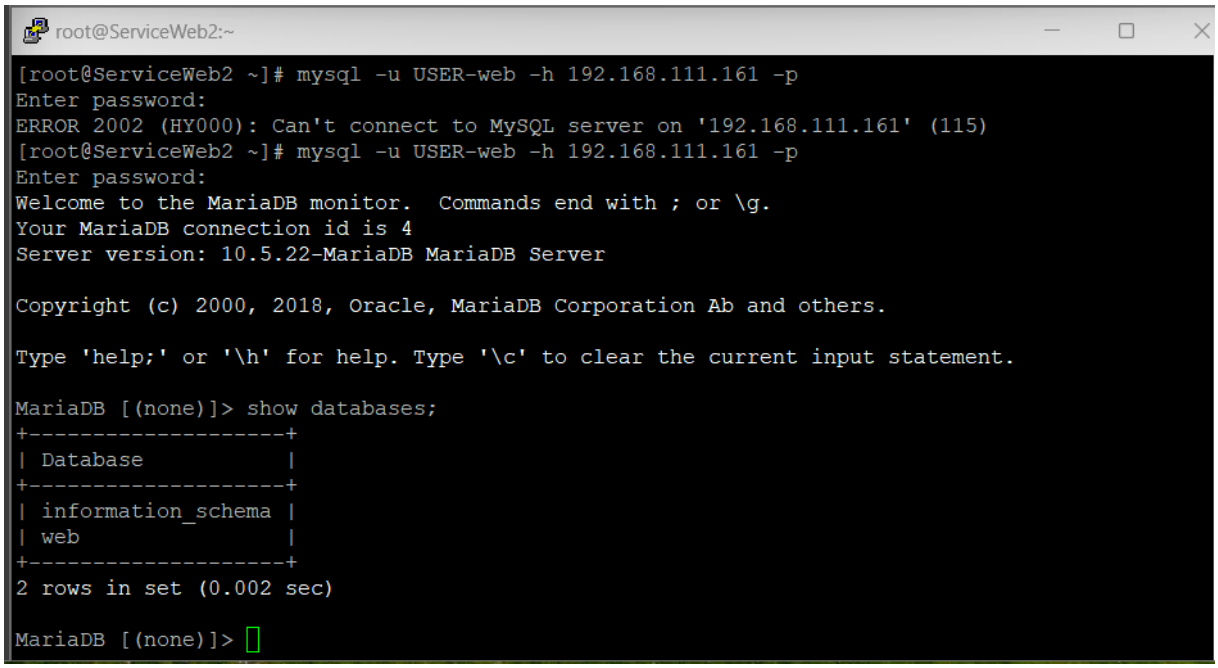
Modifier ceci dans les deux sites :

```
[root@ServiceWeb2 ~]# nano /var/www/ecommerce.uk/PricingSubscription/config.php
```

root@ServiceWeb2:/var/www/ecommerce.uk/PricingSubscription

```
 GNU nano 5.6.1                    config.php                    Modified
<?php
$SETTINGS["mysql_user"]='USERWEBUK';
$SETTINGS["mysql_pass"]='webuk';
$SETTINGS["hostname"]='192.168.111.161';
$SETTINGS["mysql_database"]='webuk';
$SETTINGS["data_table"]='registrations';
$SETTINGS["paypal_address"]='email@domain.com';
?>
```

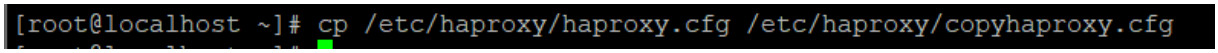
Normalement si tout est fait correctement, vous allez pouvoir depuis WEB02, vous connectez sur le moteur BDD de WEB01 :



Mise en place d'un serveur HA

On lance l'installation du package haproxy yum install haproxy

Le fichier de configuration se trouve dans /etc/haproxy/haproxy.cfg On fait une copie du fichier avant toute modification. :



Nous allons modifier les paramètres pour ajouter nos serveurs :

```
#---------------------------------------------------------------------
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#---------------------------------------------------------------------
defaults
    mode                    tcp
    log                     global
    option                  httplog
    option                  dontlognull
    option http-server-close
    option forwardfor       except 127.0.0.0/8
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                 3000
    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

#---------------------------------------------------------------------
```

```
#---------------------------------------------------------------------
# round robin balancing between the various backends
#---------------------------------------------------------------------
backend app
    balance     roundrobin
    server  app1 127.0.0.1:5001 check
    server  app2 127.0.0.1:5002 check
    server  app3 127.0.0.1:5003 check
    server  app4 127.0.0.1:5004 check
        server service_web 192.168.111.161:443 ssl verify none check
        server service_web02 192.168.111.128:443 ssl verify none check
```

On active rsyslog : Dans le répertoire /etc/rsyslog.conf Il faut décommenter les lignes cf à gauche.

module(load="imudp") # needs to be done just once

input(type="imudp" port="514")

```
[root@localhost ~]# nano /etc/rsyslog.conf
```

root@localhost:~

```
  GNU nano 5.6.1                        /etc/rsyslog.conf                        Modified
# rsyslog configuration file

# For more information see /usr/share/doc/rsyslog-*/rsyslog_conf.html
# or latest version online at http://www.rsyslog.com/doc/rsyslog_conf.html
# If you experience problems, see http://www.rsyslog.com/doc/troubleshoot.html

#### GLOBAL DIRECTIVES ####

# Where to place auxiliary files
global(workDirectory="/var/lib/rsyslog")

# Use default timestamp format
module(load="builtin:omfile" Template="RSYSLOG_TraditionalFileFormat")

# Include all config files in /etc/rsyslog.d/
include(file="/etc/rsyslog.d/*.conf" mode="optional")

#### MODULES ####

module(load="imuxsock"    # provides support for local system logging (e.g. via logger comma>
       SysSock.Use="off") # Turn off message reception via local log socket;
                          # local messages are retrieved through imjournal now.
module(load="imjournal"             # provides access to the systemd journal
       StateFile="imjournal.state") # File to store the position in the journal
#module(load="imklog") # reads kernel messages (the same are read from journald)
#module(load="immark") # provides --MARK-- message capability

# Provides UDP syslog reception
# for parameters see http://www.rsyslog.com/doc/imudp.html
module(load="imudp") # needs to be done just once
input(type="imudp" port="514")

# Provides TCP syslog reception
# for parameters see http://www.rsyslog.com/doc/imtcp.html
#module(load="imtcp") # needs to be done just once
#input(type="imtcp" port="514")

#### RULES ####

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
```

SElinux veille sur tout, il est nécessaire de lui donner une policy :

```
[root@localhost ~]# setsebool -P haproxy_connect_any 1
```

On démarre haproxy et rsyslog :

```
[root@localhost ~]# systemctl start haproxy
[root@localhost ~]# systemctl start rsyslog
```

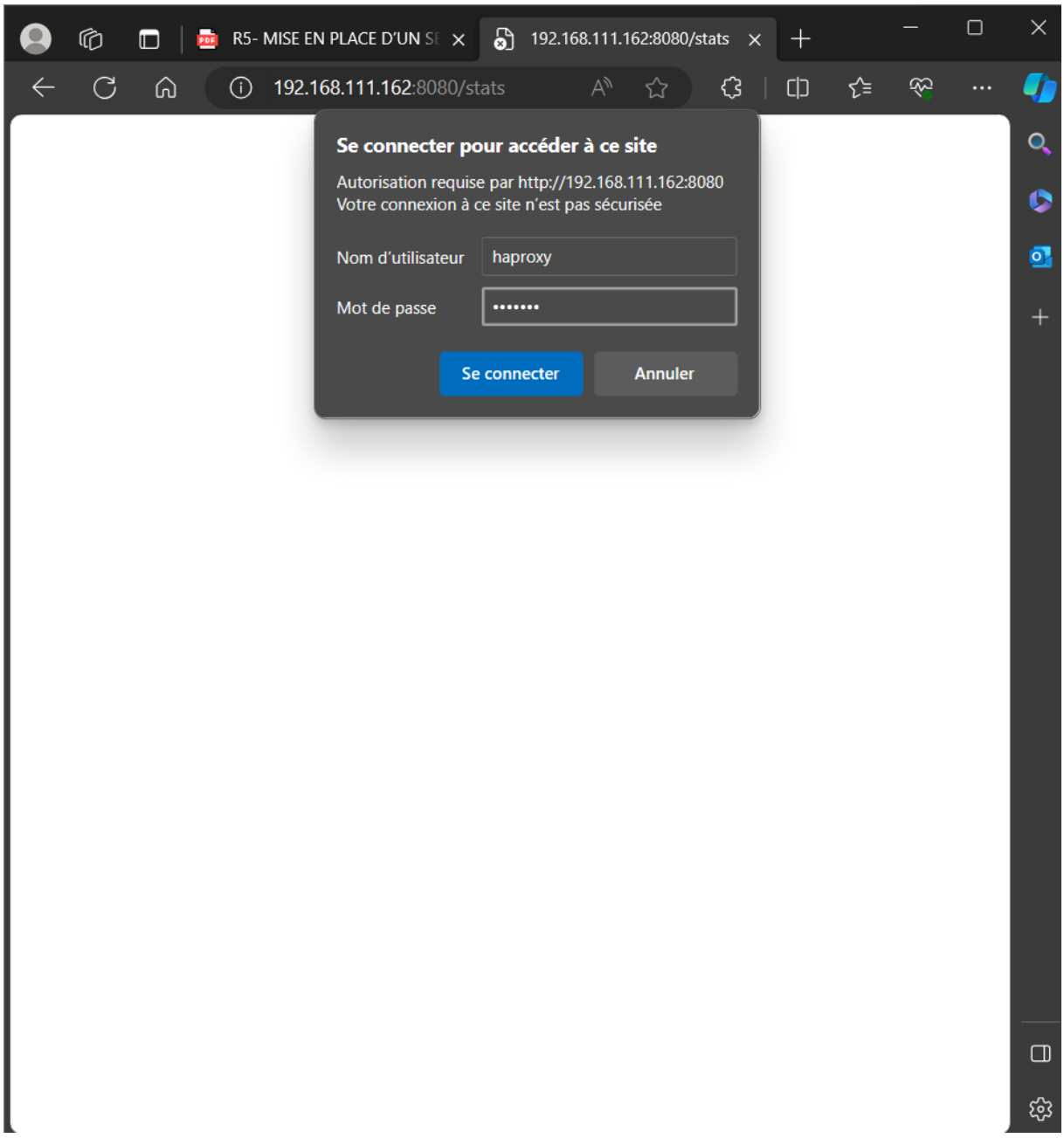
Depuis le serveur HA Proxy

On ouvre le port 8080 :

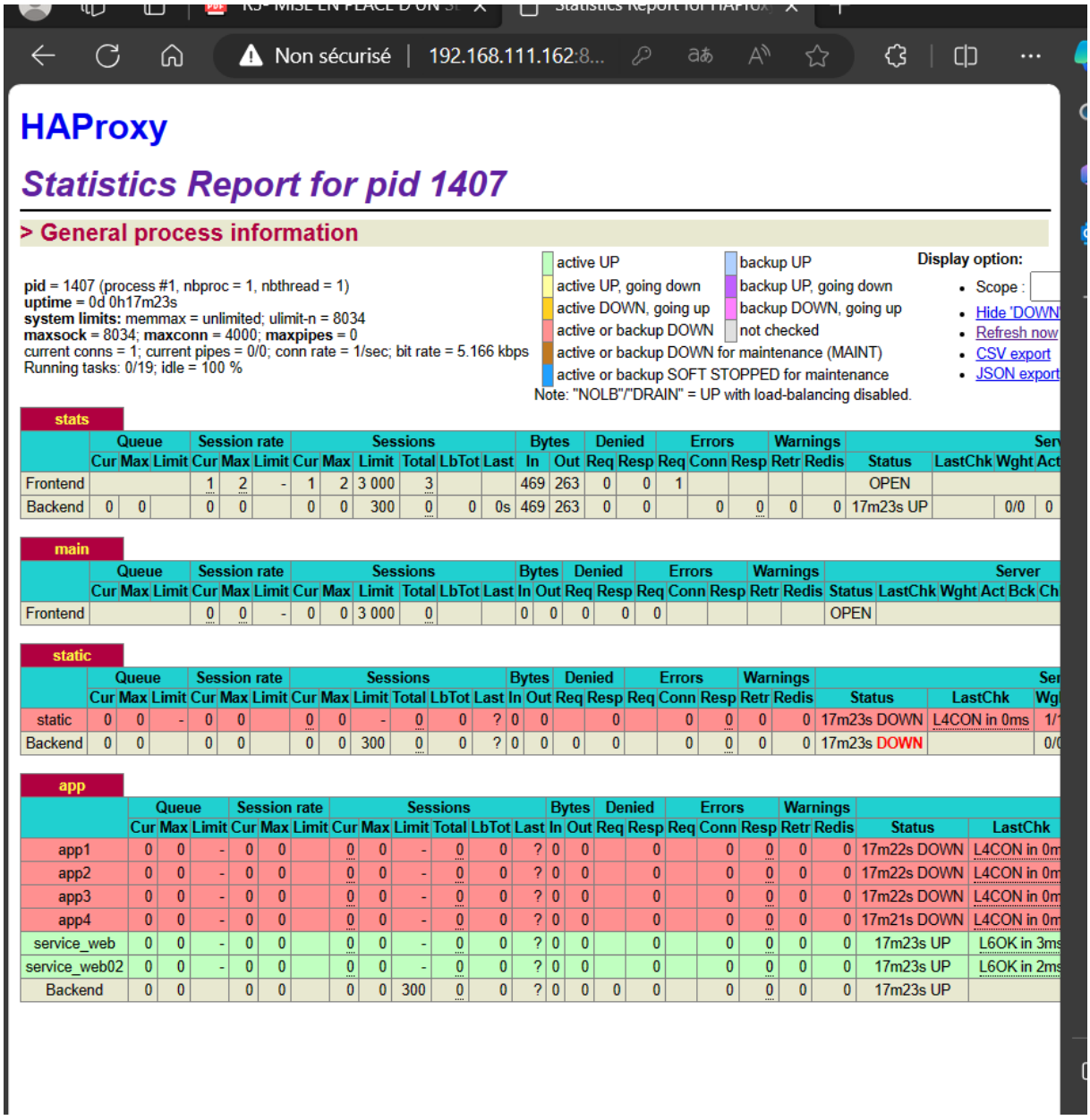
```
[root@localhost ~]# firewall-cmd --add-port=8080/tcp -permanent
usage: see firewall-cmd man page
firewall-cmd: error: unrecognized arguments: -permanent
[root@localhost ~]# firewall-cmd --add-port=8080/tcp --permanent
success
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 8080/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

On vérifie dans un navigateur WEB, l'accès à la page d'administration : http://192.168.111.162:8080/stats :

On nous demande un utilisateur et un mdp que l'on préalablement modifié dans le fichier de conf haproxy (U : haproxy ; mdp : haproxy)



On appui sur connecter



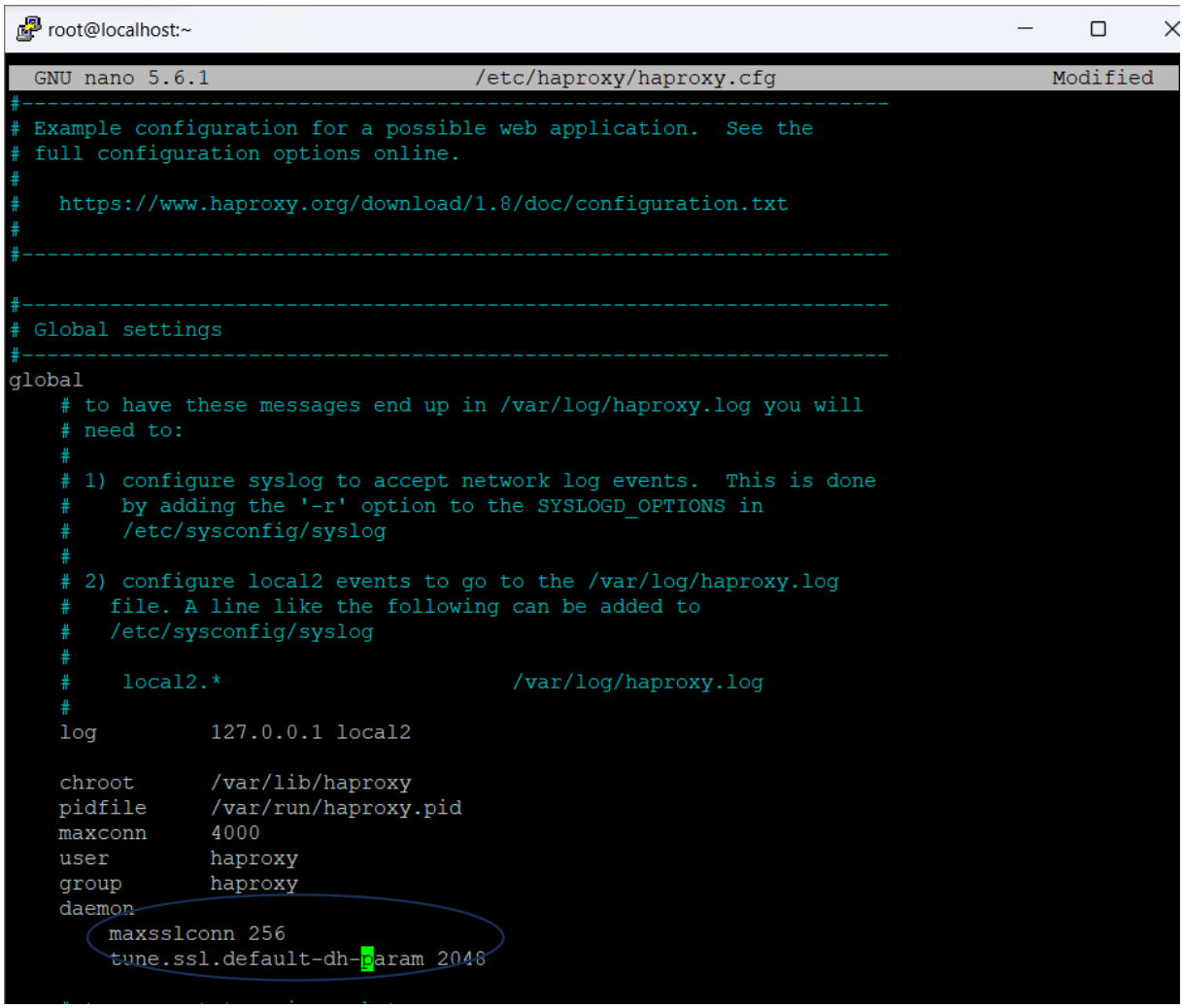
Sécurisation

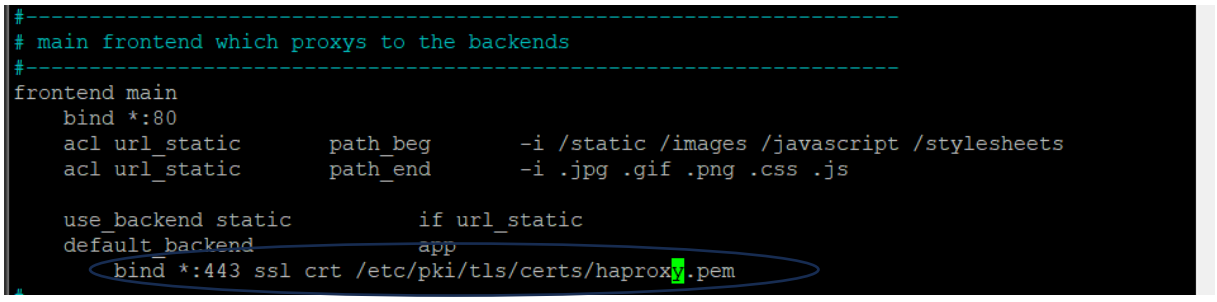
Nous allons mettre en place un certificat autosigné pour le service web Haproxy : cd /etc/pki/tls/certs :

```
[root@localhost certs]# openssl req -x509 -nodes -newkey rsa:2048 -keyout /etc/pki/tls/certs/
haproxy.pem -out /etc/pki/tls/certs/haproxy.pem -days 365
....+.....+....++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*..+.........+
.......+...+............+..+...+......+.+...+...+...+............+..+............+...+.+.....
+.+...+.....+..............++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+*.........+..........+...+.........+.....+....+.........+..+...........+..............+....+.
.......+....+...+...+..............+............+...+........+.+......+.....+...+.........+.
..+.......+......+......+........+...+............+.+......+........+.+.........+..+.........
.+.....+.+...+..+.........+...+..............+.+.....+.+......+.....+.....+..........+........+.+.
.....+..+.+..........+...+..+.......+...+......+.....+..........+..+....+.....+...+...+........
+.....+...+.......+..+....+.....+............+..............+....+.....+............+..+...
.......+...+.....+...+...+.........+......+.+.....+....+................+.......+..+.+.....+
...+............+...+.......+.........+...+..............+........+.......+.....+.+.........
........+...+....+..............+......+.........+..++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++
..........+.........+..+...+.+...+.....++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++*...+..+....+.....+.++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++*......+...+...................+......+...........+..............+...+.+.....+.......+..+
.......+......+..+......+......+...+............+.+.........+.........+..+...+......+.+.....+
.......+.....+.......+......+..+.......+......+..+.+.....+............+...........+............
..................+...+..+.+......+...........+....+........+.+..............+.............+..+
...+....+.........+.......+......+...+.....+....+..+...+............+....+.....+.+.........+...
+.......+..+...+.+.....+......+.+...+...........+.+.....+.......+..+..................+.....
......+....+.........+.....+.......+........+...........+.+..+..........+..+..............+
....+..............+......+........+...+............+.+...........+...+........+.....+...+...
...+....+.........+.......+...+.........+.........+...........+..+....+............+...+..++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----
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]:HA
State or Province Name (full name) []:FR
Locality Name (eg, city) [Default City]:Lille
Organization Name (eg, company) [Default Company Ltd]:haproxy
Organizational Unit Name (eg, section) []:admin
Common Name (eg, your name or your server's hostname) []:haproxy
Email Address []:alexandre.lespagnol@gastonberger.fr
[root@localhost certs]#
```

Une fois généré, on doit ajouter les informations suivantes dans notre fichier de configuration haproxy :

```
[root@localhost ~]# nano /etc/haproxy/haproxy.cfg
```





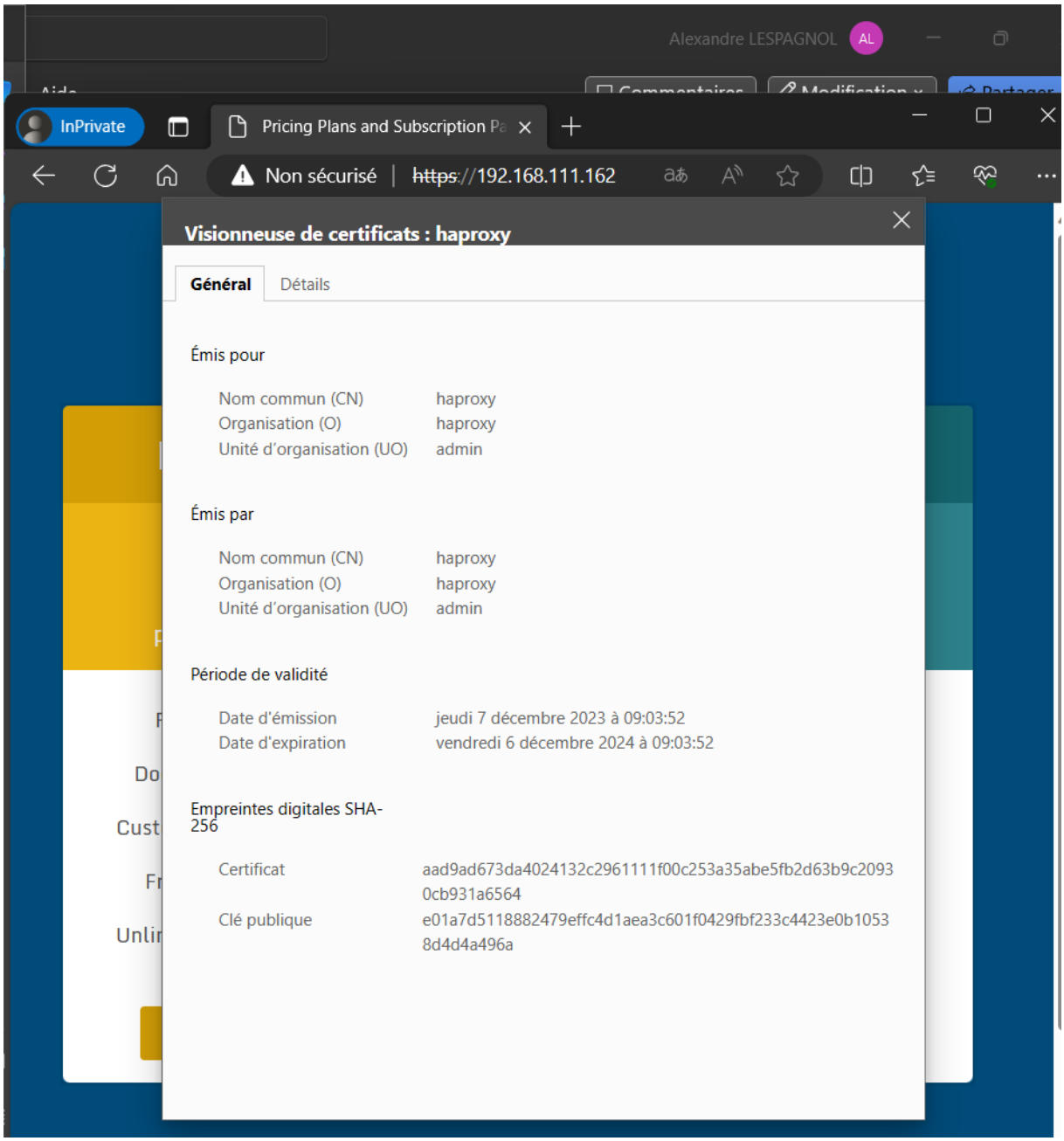
Et on redémarre le service :

```
[root@localhost ~]# systemctl restart haproxy
```

On ouvre maintenant le port 443 pour le https :

```
[root@localhost ~]# firewall-cmd --add-port=443/tcp --permanent
success
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 8080/tcp 443/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost ~]#
```

En accédant à notre page internet, nous avons un message d'erreur en raison de notre certificat autosigné :



Exercices

On ferme notre vm service web :

On regarde le comportement dans l'interface en tapant dans notre navigateur (192.168.111.162:8080/stats) :

On remarque bien que le service web est bien en rouge le le service web 2 en vert :

app

	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	Dwntme	Thrtle
app1	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	13m50s DOWN	L4CON in 0ms	1/1	Y	-	1	1	13m50s	-
app2	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	13m50s DOWN	L4CON in 0ms	1/1	Y	-	1	1	13m50s	-
app3	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	13m50s DOWN	L4CON in 0ms	1/1	Y	-	1	1	13m50s	-
app4	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	13m50s DOWN	L4CON in 0ms	1/1	Y	-	1	1	13m50s	-
service_web	0	0	-	0	2		0	3	-	5	5	5m38s	229 265	129 282		0		0	0	0	0	3m2s DOWN	* L4TOUT in 2000ms	1/1	Y	-	3	1	3m2s	-
service_web02	0	0	-	0	3		3	3	-	7	7	44s	4 217	218 765		0		0	0	0	0	13m51s UP	L6OK in 3ms	1/1	Y	-	0	0	0s	-
Backend	0	0		0	4		3	5	300	12	12	44s	233 482	348 047	0	0		0	0	0	0	13m51s UP		1/1	1	0		0	0s	

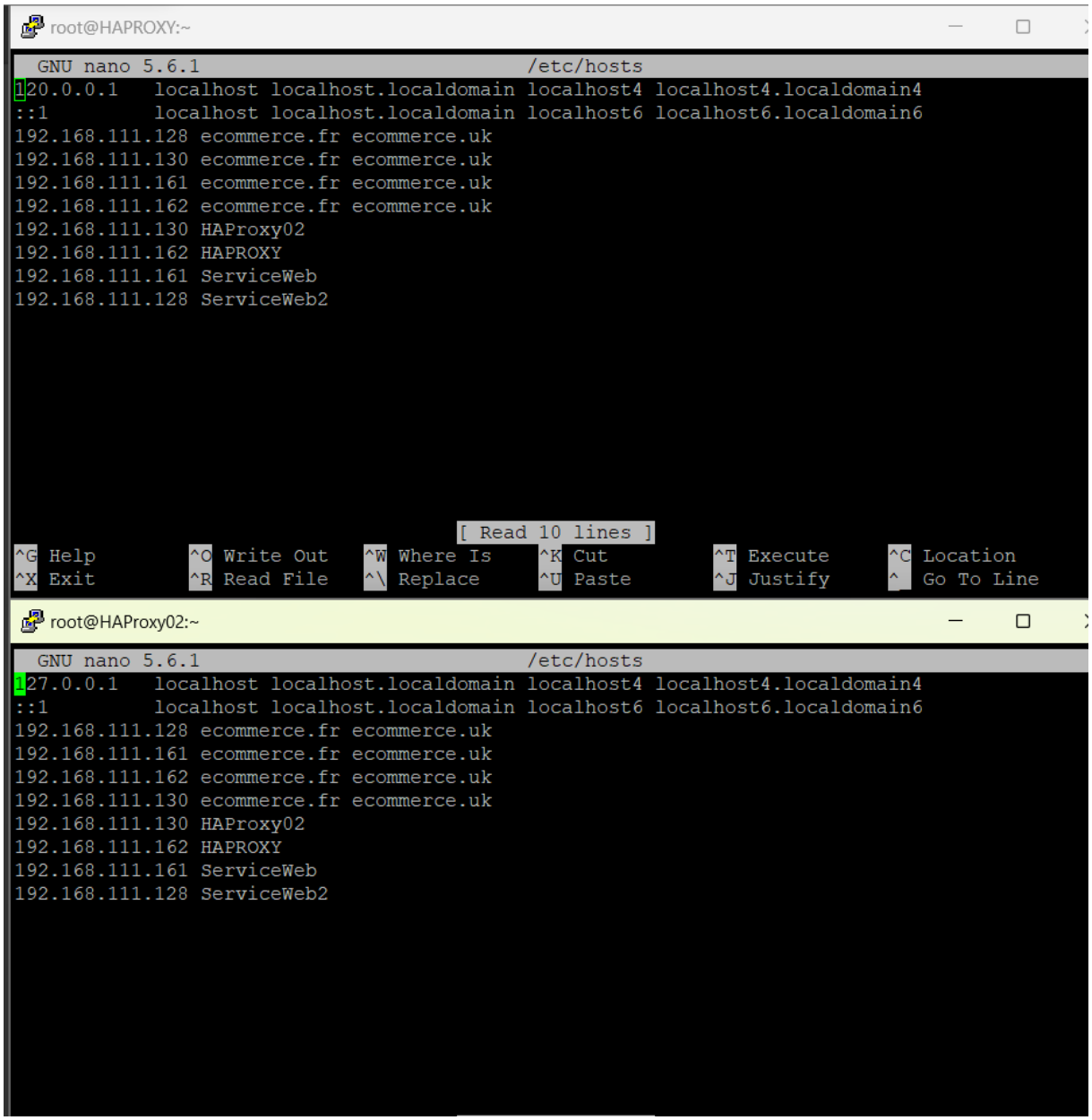
Installation HA proxy bis

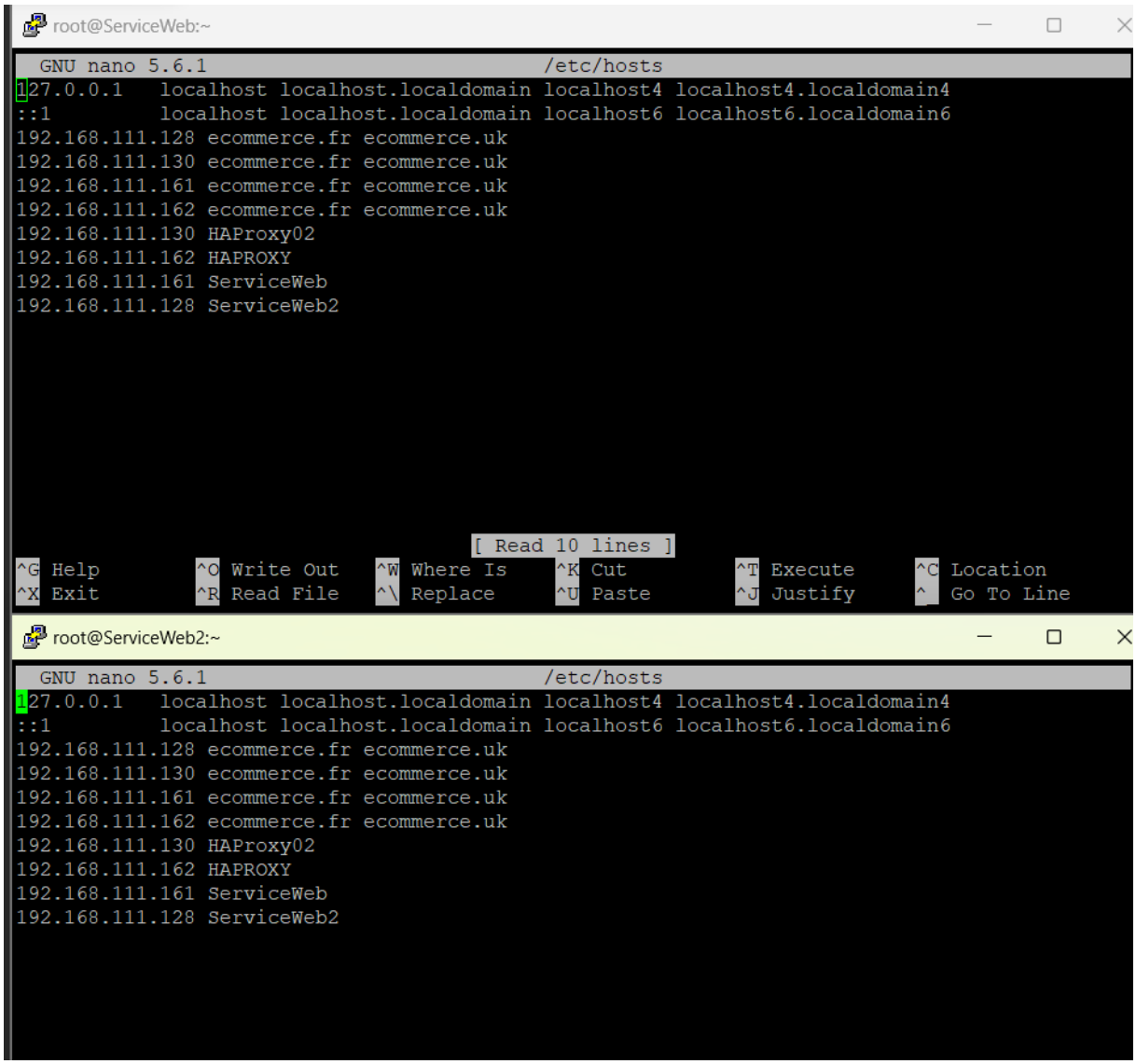
On fait un clone de notre haproxy

Validation de bon fonctionnement

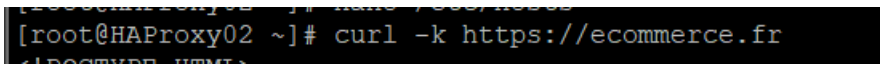
Dans chaque fichier hosts de toute nos machines faire ceci :

```
[root@HAProxy02 ~]# nano /etc/hosts
```


root@HAPROXY:~
GNU nano 5.6.1 /etc/hosts
120.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.111.128 ecommerce.fr ecommerce.uk
192.168.111.130 ecommerce.fr ecommerce.uk
192.168.111.161 ecommerce.fr ecommerce.uk
192.168.111.162 ecommerce.fr ecommerce.uk
192.168.111.130 HAProxy02
192.168.111.162 HAPROXY
192.168.111.161 ServiceWeb
192.168.111.128 ServiceWeb2
[Read 10 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
root@HAProxy02:~
GNU nano 5.6.1 /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.111.128 ecommerce.fr ecommerce.uk
192.168.111.161 ecommerce.fr ecommerce.uk
192.168.111.162 ecommerce.fr ecommerce.uk
192.168.111.130 ecommerce.fr ecommerce.uk
192.168.111.130 HAProxy02
192.168.111.162 HAPROXY
192.168.111.161 ServiceWeb
192.168.111.128 ServiceWeb2



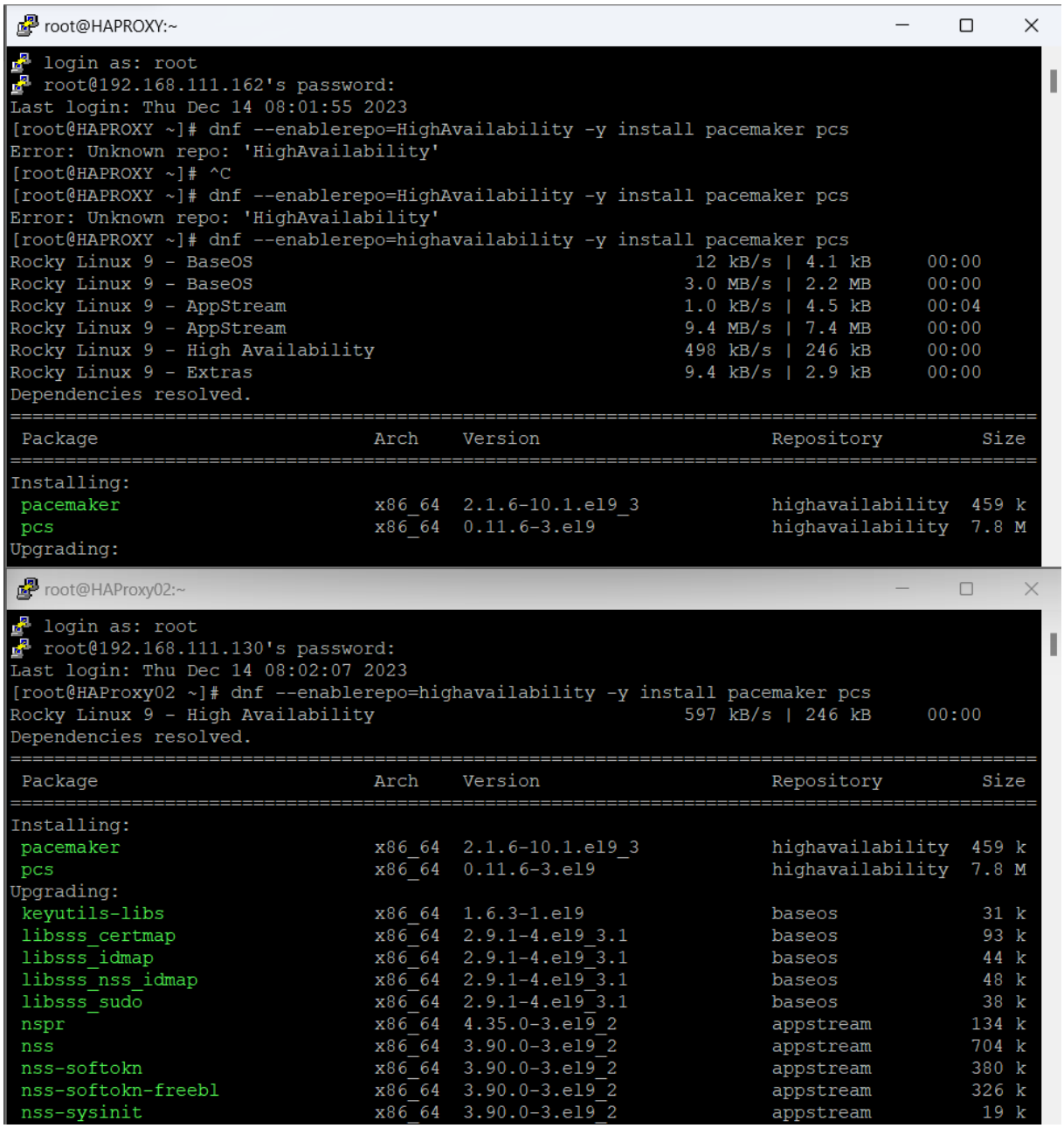
Ensuite avec cette commande on arrive bien à voir notre page :



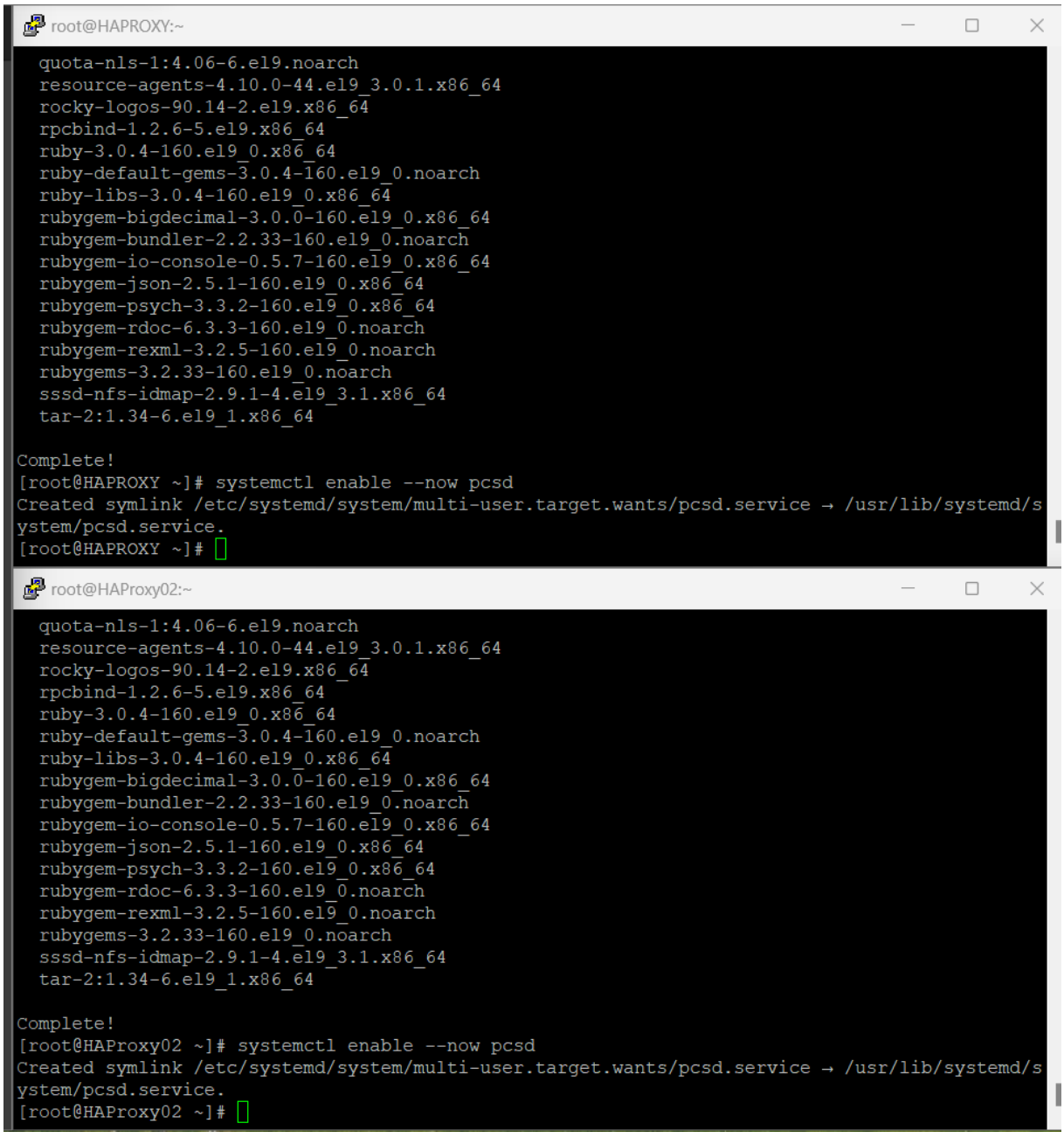
```
[root@HAProxy02 ~]# curl -k https://ecommerce.ff
<!DOCTYPE HTML>
<html>
        <head>
                <title>Pricing Plans and Subscription Payment En Francais | by PHPJabbers.com
</title>
                <link href="css/style.css" rel="stylesheet" type="text/css" media="all"/>
                <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1, maximum-s
cale=1">
                <link href="https://fonts.googleapis.com/css?family=Monda" rel="stylesheet">

                <script src="js/jquery-1.11.0.min.js"></script>
                <script src="js/jquery.magnific-popup.js" type="text/javascript"></script>
                <script src="js/jquery.validate.min.js" type="text/javascript"></script>
                <script>
                        $(document).ready(function() {
                                $('.popup-with-zoom-anim').magnificPopup({
                                        type: 'inline',
                                        fixedContentPos: false,
                                        fixedBgPos: true,
                                        overflowY: 'auto',
                                        closeBtnInside: true,
                                        preloader: false,
                                        midClick: true,
                                        removalDelay: 300,
                                        mainClass: 'my-mfp-zoom-in'
                                });
                                $('.popup-with-zoom-anim').on('click', function(e) {
                                        $('#signUpForm').find('input[name="Plan"]').val($(thi
s).data("plan"));

                                        $('#signUpForm').find('input[name="Price"]').val($(th
is).data("price"));
                                });
                                $('#signUpForm').validate({
                                        errorPlacement: function(error, element) {
                                                if (element.attr('name') == 'Terms') {
                                                        error.insertAfter(element.parent());
```

Installation de Pacemaker

Sur chaque serveur HA proxy, il est nécessaire de :



On active le démarrage sur les deux serveurs haproxy :



Ont créé notre cluster sur nos deux serveurs haproxy:

```
[root@HAPROXY ~]# passwd hacluster
Changing password for user hacluster.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@HAPROXY ~]#
```

```
[root@HAProxy02 ~]# passwd hacluster
Changing password for user hacluster.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

On réalise les ouvertures dans le firewall sur les deux serveurs haproxy, Il existe un service pour PaceMakeer.

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

On redémarre le firewall :

```
[root@HAPROXY ~]# firewall-cmd --reload
success
[root@HAPROXY ~]# firewall-cmd --list-services
cockpit dhcpv6-client high-availability ssh
```

```
[root@HAProxy02 ~]#  firewall-cmd --add-service=high-availability --permanent
success
[root@HAProxy02 ~]#  firewall-cmd --reload
success
[root@HAProxy02 ~]#  firewall-cmd --list-services
cockpit dhcpv6-client high-availability ssh
[root@HAProxy02 ~]#
```

```
root@HAPROXY:~
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default q
len 1000
    link/ether 00:0c:29:e7:88:58 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.111.162/24 brd 192.168.111.255 scope global dynamic noprefixroute ens160
       valid_lft 1451sec preferred_lft 1451sec
    inet6 fe80::20c:29ff:fee7:8858/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@HAPROXY ~]# nano /etc/hosts
[root@HAPROXY ~]# nano /etc/hosts
[root@HAPROXY ~]# pcs host auth HAPROXY HAProxy02
Username: hacluster
Password:
HAPROXY: Authorized
Error: HAProxy02: Username and/or password is incorrect
[root@HAPROXY ~]# nano /etc/hosts
[root@HAPROXY ~]# nano /etc/hosts
[root@HAPROXY ~]# pcs host auth HAPROXY HAProxy02
Username: hacluster
Password:
HAPROXY: Authorized
HAProxy02: Authorized
[root@HAPROXY ~]#
```

Nous allons déployer notre cluster **que sur HAPROXY !** :

```
root@HAPROXY:~
Username: hacluster
Password:
HAPROXY: Authorized
HAProxy02: Authorized
[root@HAPROXY ~]# pcs cluster setup ha_cluster HAPROXY HAProxy02
No addresses specified for host 'HAPROXY', using 'haproxy'
No addresses specified for host 'HAProxy02', using 'haproxy02'
Destroying cluster on hosts: 'HAPROXY', 'HAProxy02'...
HAPROXY: Successfully destroyed cluster
HAProxy02: Successfully destroyed cluster
Requesting remove 'pcsd settings' from 'HAPROXY', 'HAProxy02'
HAPROXY: successful removal of the file 'pcsd settings'
HAProxy02: successful removal of the file 'pcsd settings'
Sending 'corosync authkey', 'pacemaker authkey' to 'HAPROXY', 'HAProxy02'
HAPROXY: successful distribution of the file 'corosync authkey'
HAPROXY: successful distribution of the file 'pacemaker authkey'
HAProxy02: successful distribution of the file 'corosync authkey'
HAProxy02: successful distribution of the file 'pacemaker authkey'
Sending 'corosync.conf' to 'HAPROXY', 'HAProxy02'
HAPROXY: successful distribution of the file 'corosync.conf'
HAProxy02: successful distribution of the file 'corosync.conf'
Cluster has been successfully set up.
[root@HAPROXY ~]#
```

On démarre notre cluster sur HAPROXY :

```
[root@HAPROXY ~]# pcs cluster start --all
HAProxy02: Starting Cluster...
HAPROXY: Starting Cluster...
```

```
[root@HAPROXY ~]# pcs cluster enable --all
HAPROXY: Cluster Enabled
HAProxy02: Cluster Enabled
```

On va pouvoir vérifier l'état de notre cluster :

```
[root@HAPROXY ~]# pcs cluster status
Cluster Status:
 Cluster Summary:
   * Stack: corosync (Pacemaker is running)
   * Current DC: HAProxy02 (version 2.1.6-10.1.el9_3-6fdc9deea29) - partition with quorum
   * Last updated: Thu Dec 14 09:09:36 2023 on HAPROXY
   * Last change:  Thu Dec 14 09:08:26 2023 by hacluster via crmd on HAProxy02
   * 2 nodes configured
   * 0 resource instances configured
 Node List:
   * Online: [ HAPROXY HAProxy02 ]

PCSD Status:
  HAProxy02: Online
  HAPROXY: Online
```

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

Membership information
----------------------
    Nodeid      Votes Name
         1          1 HAPROXY (local)
         2          1 HAProxy02
```

```
[root@HAPROXY ~]# pcs status cluster
Cluster Status:
 Cluster Summary:
   * Stack: corosync (Pacemaker is running)
   * Current DC: HAProxy02 (version 2.1.6-10.1.el9_3-6fdc9deea29) - partition with quorum
   * Last updated: Thu Dec 14 09:10:59 2023 on HAPROXY
   * Last change:  Thu Dec 14 09:08:26 2023 by hacluster via crmd on HAProxy02
   * 2 nodes configured
   * 0 resource instances configured
 Node List:
   * Online: [ HAPROXY HAProxy02 ]

PCSD Status:
  HAPROXY: Online
  HAProxy02: Online
```

```
[root@HAPROXY ~]#

pcs status nodes
Pacemaker Nodes:
 Online: HAPROXY HAProxy02
 Standby:
 Standby with resource(s) running:
 Maintenance:
 Offline:
Pacemaker Remote Nodes:
 Online:
 Standby:
 Standby with resource(s) running:
 Maintenance:
 Offline:
```

On peut vérifier si nous avons des erreurs :

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

Nous avons des erreurs concernant STONIGH car il n'est pas configuré.

Stonith

Stonith ou "Shoot The Other Node In The Head" ou encore "Shoot The Offending Node In The Head" C'est une méthode d'isolation d'un nœud qui pour une raison ou une autre ne répond plus. L'idée est d'éviter à tout prix le fameux split-brain qui peut amener tous vos nœuds à fournir le même service (ou à ne plus le fournir du tout).

Faire ceci afin de ne plus avoir l'erreur :

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

Nous pouvons vérifier les versions avec la commande :

```
[root@HAProxy02 ~]# 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
```

Nous allons maintenant créer une adresse IP virtuelle permettant de rediriger le trafic sur l'un ou l'autre nœud :

```
[root@HAProxy02 ~]# pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=192.168.111.163 c
idr_netmask=24 op monitor interval=30s
```

Nous voyons bien notre ip virtuelle, elle est portée par le serveur maître (HAPROXY). :

```
[root@HAProxy02 ~]# pcs status resources
  * virtual_ip  (ocf:heartbeat:IPaddr2):         Started HAPROXY
```

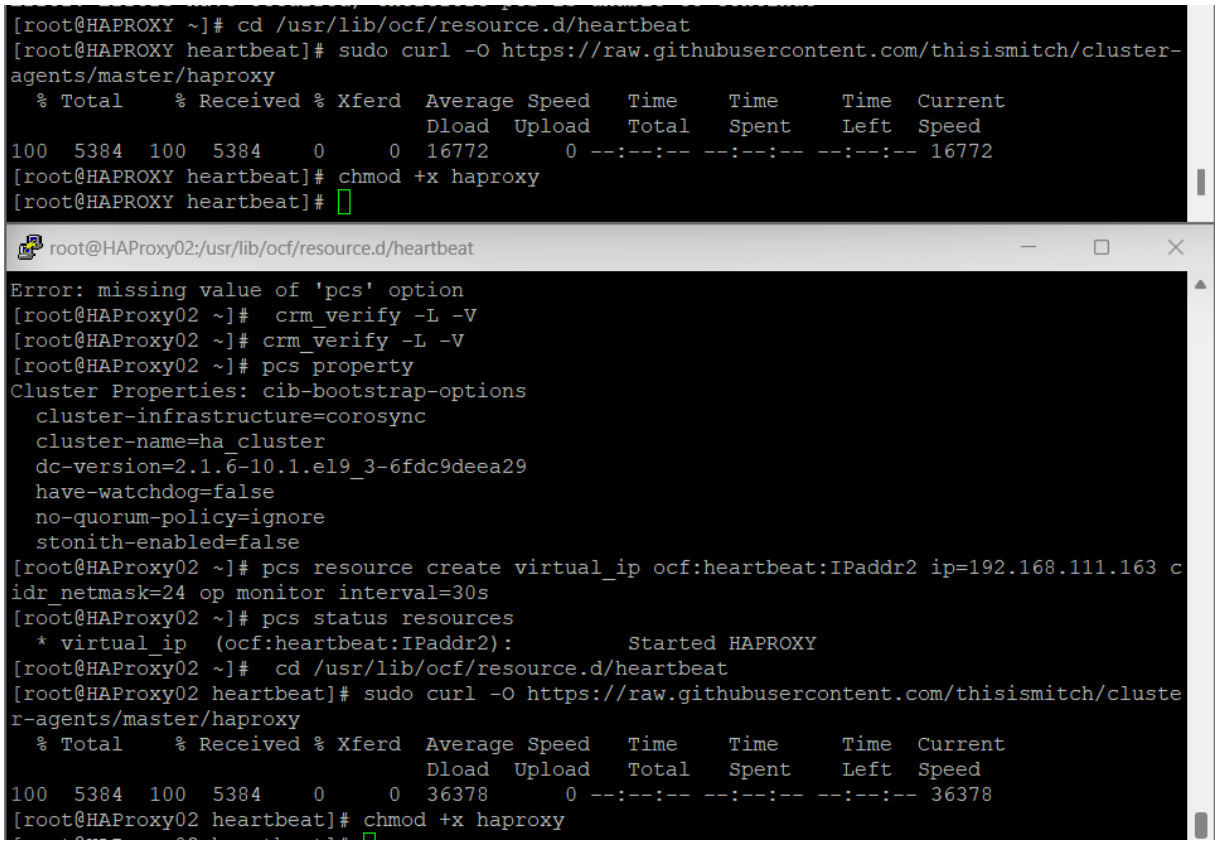
Si je fais un ip -c a, je vois que mon serveur HAPROXY dispose de l'iP que j'ai précédemment renseigné

```
[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 fq_codel state UP group default q
len 1000
    link/ether 00:0c:29:e7:88:58 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.111.162/24 brd 192.168.111.255 scope global dynamic noprefixroute ens160
       valid_lft 1750sec preferred_lft 1750sec
    inet 192.168.111.163/24 brd 192.168.111.255 scope global secondary ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fee7:8858/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

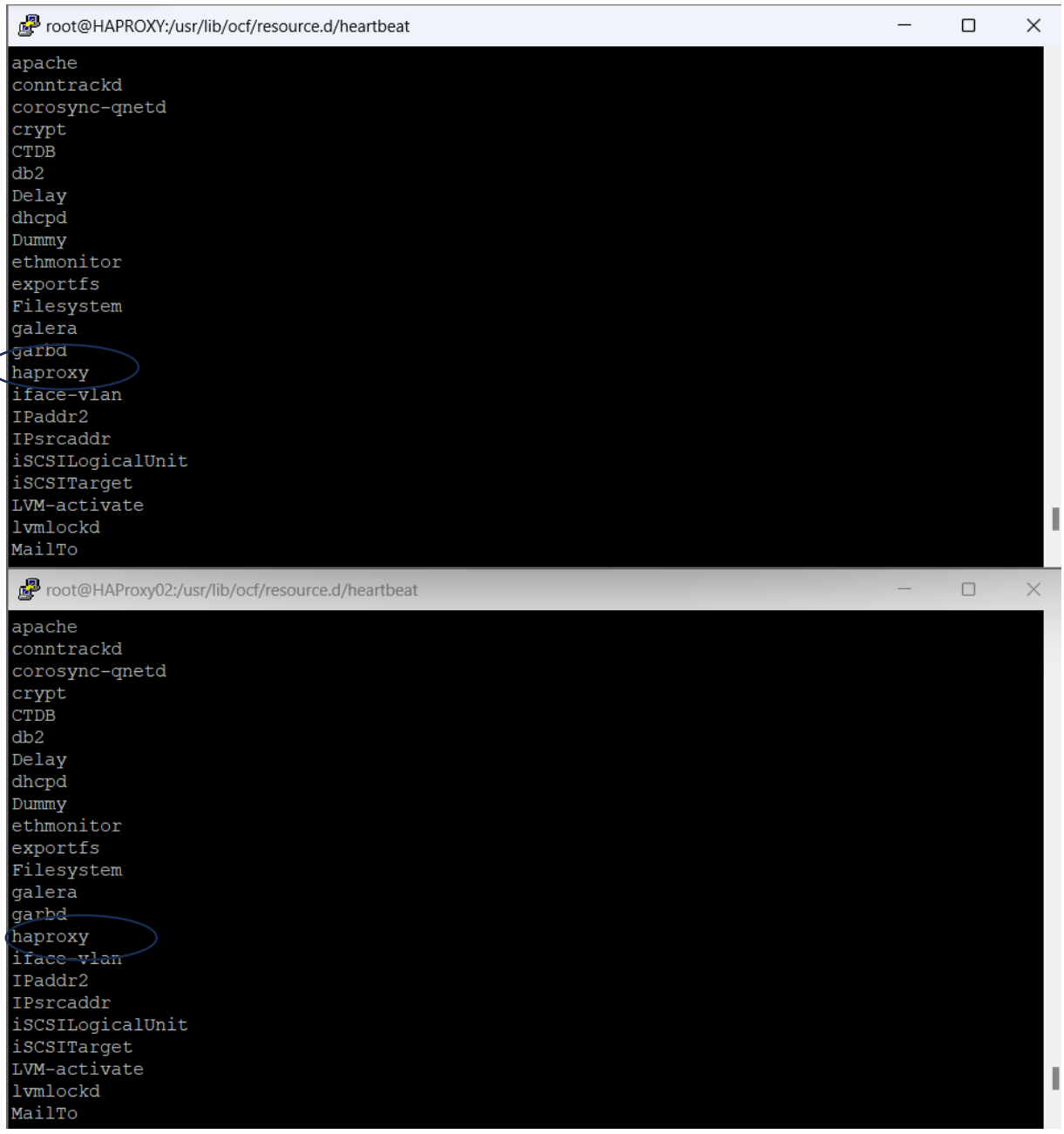
Nous allons créer notre cluster de ressource maintenance :

```
[root@HAPROXY ~]# pcs resource create haproxy ocf:heartbeat:haproxy binpath=/usr/sbin/haprox
conffile=/etc/haproxy/haproxy.cfg op monitor interval=10s
Error: Agent 'ocf:heartbeat:haproxy' is not installed or does not provide valid metadata: crm
```

Si vous avez une erreur de ressource, il est nécessaire de récupérer l'agent HAproxy sur les 2 serveurs :



Nous pouvons savoir les agents qui sont disponibles sur le serveur avec la commande :



Ensuite on retape cette commande et il n'y a donc plus d'erreur :

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

Nous affectons notre ressource à notre cluster :

```
[root@HAPROXY heartbeat]# pcs resource group add HAproxyGroup virtual_ip haproxy
```

Nous ajoutons une contrainte pour que l'IP virtuelle et le HAproxy soit opérationnel sur le même nœud en même temps :

```
[root@HAPROXY heartbeat]# pcs constraint colocation add haproxy with virtual_ip
```

Pcs status

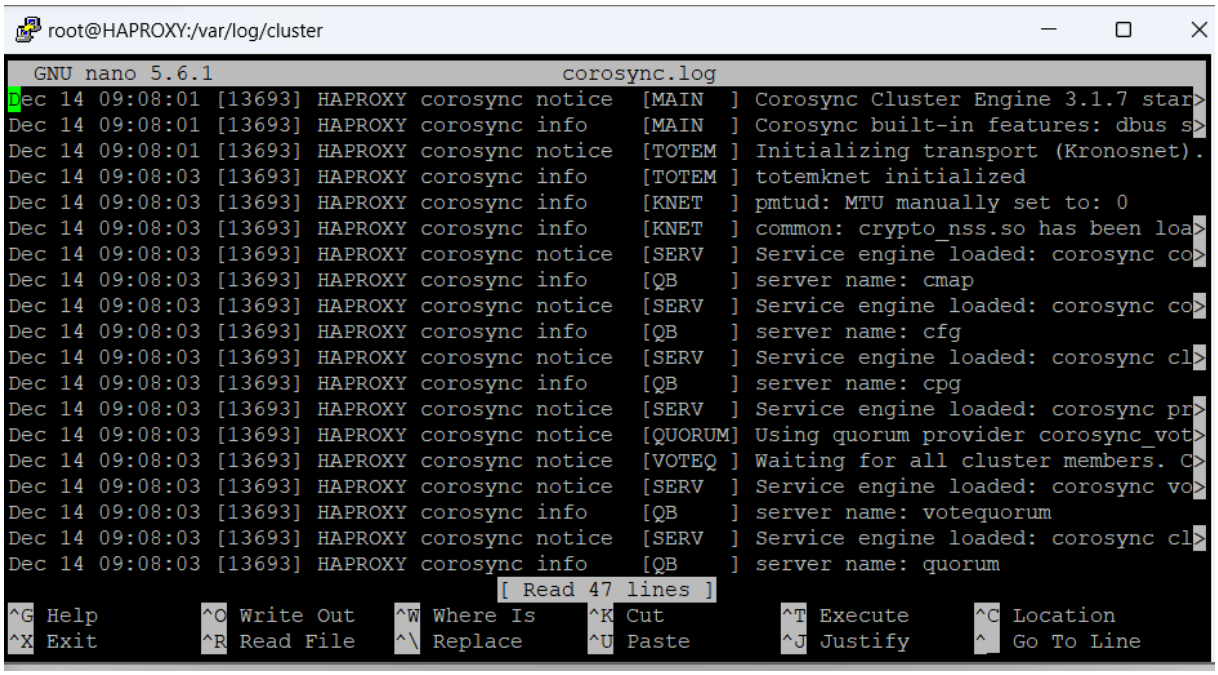
Cela affiche l'ensemble des informations de notre cluster :

```
[root@HAPROXY heartbeat]# pcs status
Cluster name: ha_cluster
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: HAProxy02 (version 2.1.6-10.1.el9_3-6fdc9deea29) - partition with quorum
  * Last updated: Thu Dec 14 10:07:18 2023 on HAPROXY
  * Last change:  Thu Dec 14 10:06:51 2023 by root via cibadmin on HAPROXY
  * 2 nodes configured
  * 2 resource instances configured

Node List:
  * Online: [ HAPROXY HAProxy02 ]

Full List of Resources:
  * Resource Group: HAproxyGroup:
    * virtual_ip        (ocf:heartbeat:IPaddr2):         Started HAPROXY
    * haproxy   (ocf:heartbeat:haproxy):         Stopped

Failed Resource Actions:
  * haproxy start on HAPROXY returned 'error' at Thu Dec 14 09:55:03 2023 after 48ms
  * haproxy start on HAProxy02 returned 'error' at Thu Dec 14 09:55:03 2023 after 60ms
```

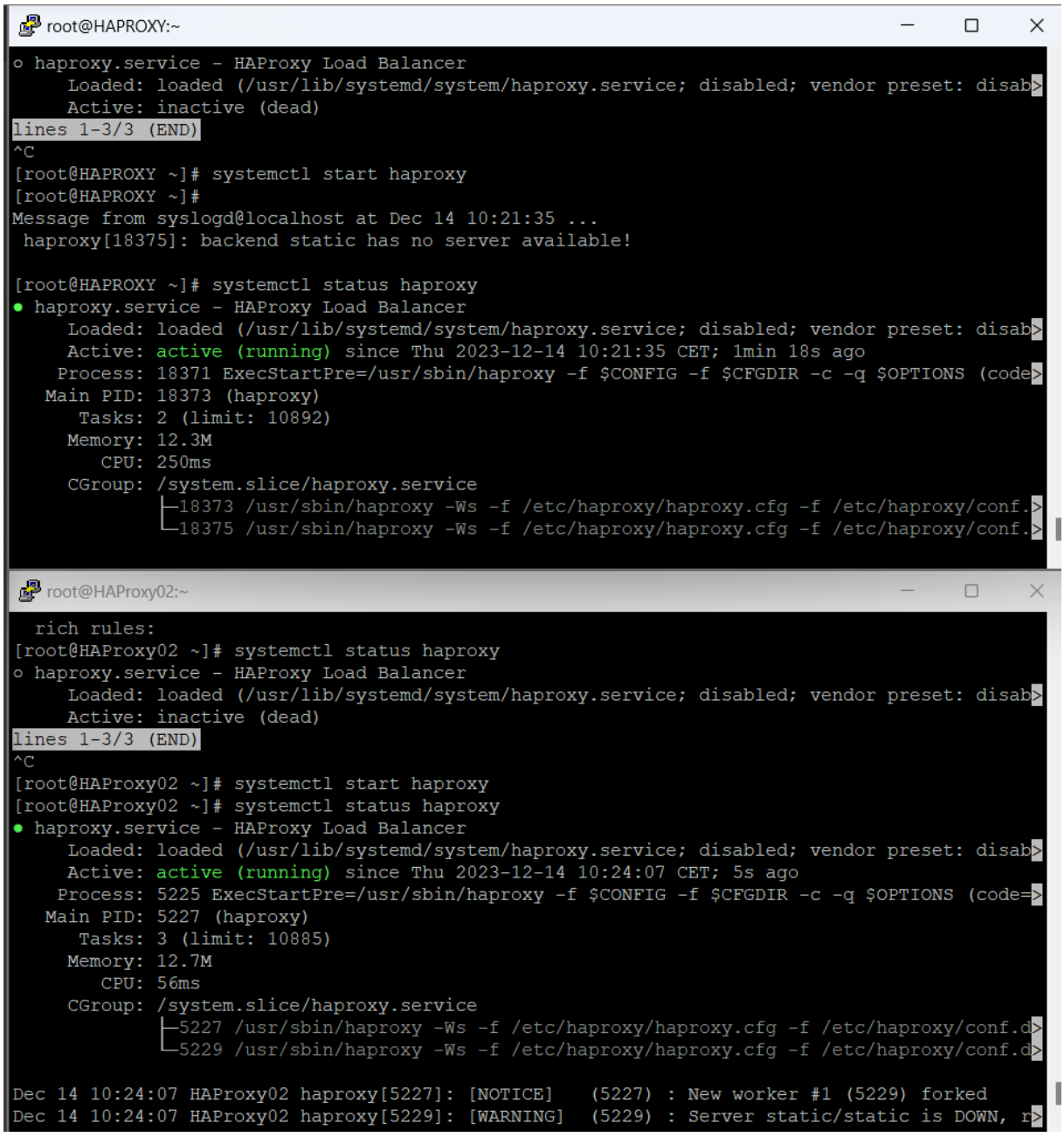
```
Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
[root@HAPROXY heartbeat]#
```

Nous pouvons retrouver les logs dans : /var/log/cluster :

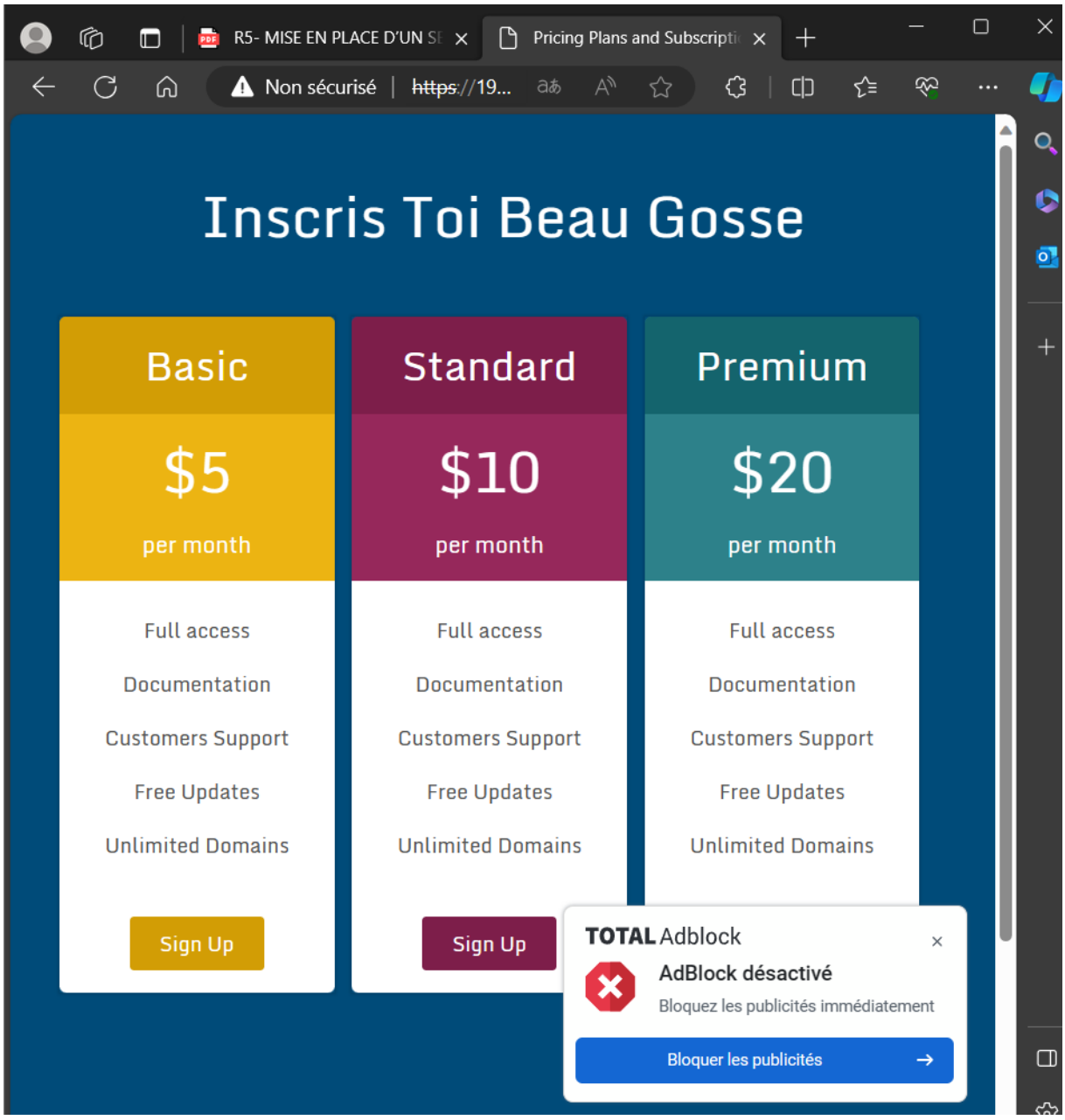
```
[root@HAPROXY cluster]# cd /var/log/cluster
[root@HAPROXY cluster]# ls
corosync.log
[root@HAPROXY cluster]# nano corosync.log
```



Sur nos deux serveurs haproxy on start haproxy :



Si nous allons sur l'adresse IP virtuelle (192.168.111.163), nous accédons à notre service WEB. Les requêtes sont automatiquement dispatchées entre nos 2 serveurs WEB. :


R5- MISE EN PLACE D'UN SE
Pricing Plans and Subscripti
Non sécurisé
https://19...
Inscris Toi Beau Gosse
Basic
$5
per month
Full access
Documentation
Customers Support
Free Updates
Unlimited Domains
Sign Up
Standard
$10
per month
Full access
Documentation
Customers Support
Free Updates
Unlimited Domains
Sign Up
Premium
$20
per month
Full access
Documentation
Customers Support
Free Updates
Unlimited Domains
TOTAL Adblock
AdBlock désactivé
Bloquez les publicités immédiatement
Bloquer les publicités

On arrête HAPROXY et on remarque bien que HAProxy02 à bien récupérer la VIP :

```
[root@HAProxy02 ~]# pcs status resources
  * Resource Group: HAproxyGroup:
    * virtual_ip        (ocf:heartbeat:IPaddr2):         Started HAProxy02
    * haproxy   (ocf:heartbeat:haproxy):         Stopped
```

```
root@HAProxy02:~
lines 1-22
^C
[root@HAProxy02 ~]#
[root@HAProxy02 ~]#
[root@HAProxy02 ~]#
[root@HAProxy02 ~]# 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 10
00
    link/ether 00:0c:29:de:69:36 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.111.130/24 brd 192.168.111.255 scope global dynamic noprefixroute ens160
       valid_lft 1583sec preferred_lft 1583sec
    inet 192.168.111.163/24 brd 192.168.111.255 scope global secondary ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fede:6936/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@HAProxy02 ~]#
```