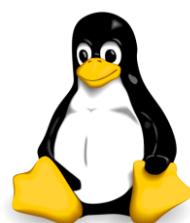




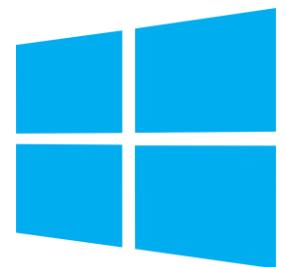
GASTON  
BERGER

LESPAGNOL  
Alexandre  
2SISR

09/11/2023



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





---

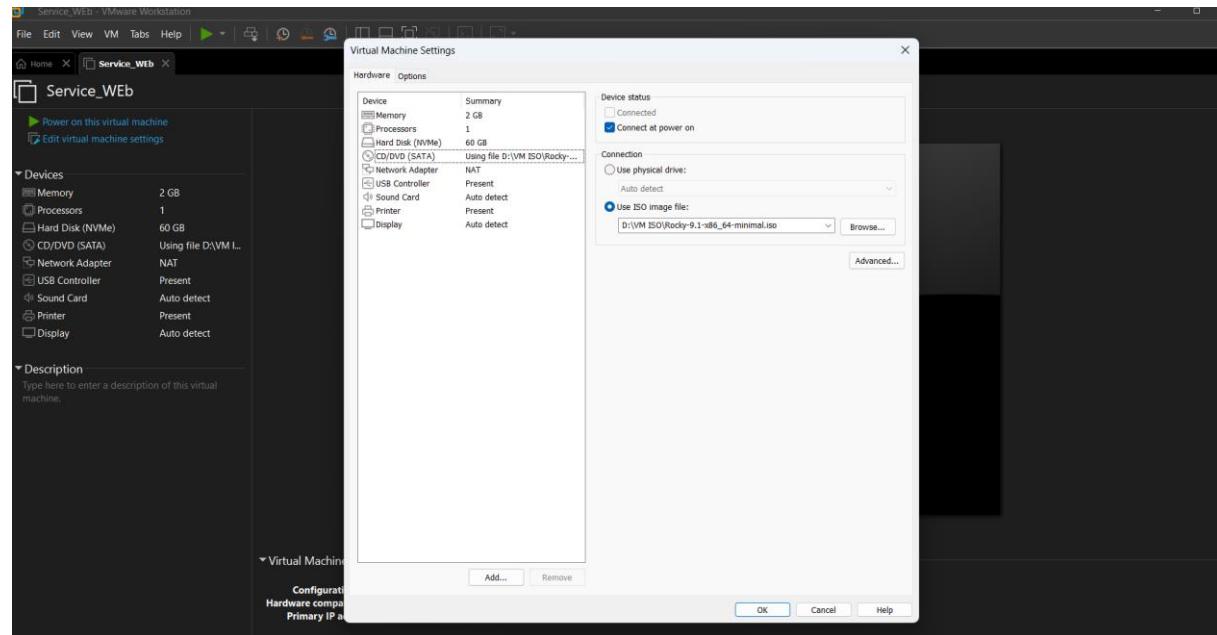
## Sommaire :

Installation machine service web .....	3
Installation d'un service web (NGINX) .....	3
Installation moteur BDD.....	13
Déploiement du site FR.....	15
Déploiement du site UK .....	24
Restriction accès IP .....	32
Analyse des logs.....	35
Analyse du site en http.....	40
Sécurisation d'un service WEB (nginx).....	43
Redirection http -https .....	56
Installation de deux serveurs red hat.....	59
Configuration BDD .....	61
Mise en place d'un serveur HA.....	64
Depuis le serveur HA Proxy .....	67
Sécurisation .....	69
Exercices .....	73
Validation de bon fonctionnement .....	74
Installation de Pacemaker .....	77
Stonith.....	82



## 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
Rocky Linux 9 - AppStream
Rocky Linux 9 - Extras
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
Total                                         3.3 MB/s | 694 kB   00:00
Rocky Linux 9 - BaseOS
Importing GPG key 0x35BD275D:
  Userid : "Rocky Enterprise Software Foundation - Release key 2022 <rele...@rockylinux.org>"
  Fingerprint: 21CB 256A E16F C54C 6E65 2949 782D 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

#AllowUsers root
PermitRootLogin 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
```

File menu    Edit menu    View menu    Insert menu    Format menu    Tools menu    Help menu

Help    Write Out    Where Is    Cut    Paste    Execute    Location    Undo    Set Mark    To Bracket    Previous  
Exit    Read File    Replace    Justify    Go To Line    Redo    Copy    Where Was    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-htpd 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-htpd-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-htpd-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-htpd-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/systemd/system/php-fpm.service.
```

Installation package mariadb-server :



```
[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
 libsepolicy       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 x86_64  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 x86_64  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
d/ Systemctl php-fpm.service.
[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/systemd/system/mariadb.service.
[root@localhost ~]#
```

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 dhcpcv6-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
```



```
root@localhost:/etc/nginx          ngnix.conf      Modified
GNU nano 5.6.1
'$status $body_bytes_sent "$http_referer" '
'"$http_user_agent" "$http_x_forwarded_for"';
```

```
access_log /var/log/nginx/access.log main;
```

```
sendfile on;
tcp_nopush on;
tcp_nodelay on;
keepalive_timeout 65;
types_hash_max_size 4096;
```

```
include /etc/nginx/mime.types;
default_type application/octet-stream;
```

```
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/ngx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;
```

```
server {
    listen 8080;
    listen [::]:80;
    server_name ;
    root /usr/share/nginx/html;
```

```
# Load configuration files for the default server block.
include /etc/nginx/default.d/*.conf;
```

```
error_page 404 /404.html;
location = /404.html { }
```

```
error_page 500 502 503 504 /50x.html;
location = /50x.html { }
```

```
}
```

```
# Settings for a TLS enabled server.
#
# server {
#     listen 443 ssl http2;
#     listen [::]:443 ssl http2;
#     server_name ;
#     root /usr/share/nginx/html;
```

```
^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 relance le service et testons une connexion avec le port :

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



The screenshot shows a Microsoft Edge browser window with a dark theme. At the top, the address bar displays the URL `192.168.111.161:8080`. A notification bar at the top right says "Définir Microsoft Edge comme votre navigateur par défaut" with buttons for "Confirmer" and "Pas maintenant". The main content area features a large white cloud icon with two small circles below it. Below the icon, the text "Désolé, impossible d'accéder à cette page." is displayed in bold. Underneath, it says "192.168.111.161 a mis trop de temps pour répondre". A section titled "Essayez :" lists several troubleshooting steps: "Exécuter des diagnostics de réseau avec Get Help.", "Effectuez une recherche sur le web pour [192 168 111 161](#)", "Vérification de la connexion", and "Vérification du proxy et du pare-feu". At the bottom left, there is a blue button labeled "Actualiser". At the bottom right, there is a "Détails" link.

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
```

```
GNU nano 5.6.1                               nginx.conf                         Modified
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile      on;
    tcp_nopush    on;
    tcp_nodelay   on;
    keepalive_timeout  65;
    types_hash_max_size 4096;

    include        /etc/nginx/mime.types;
    default_type   application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

server {
    listen      80;          listen      [::]:80;
    server_name ;
    root       /usr/share/nginx/html;

    # Load configuration files for the default server block.
}

^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@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/systemd/system/mariadb.service.
[root@localhost ~]#
```

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)]> █
```

Créer 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 :

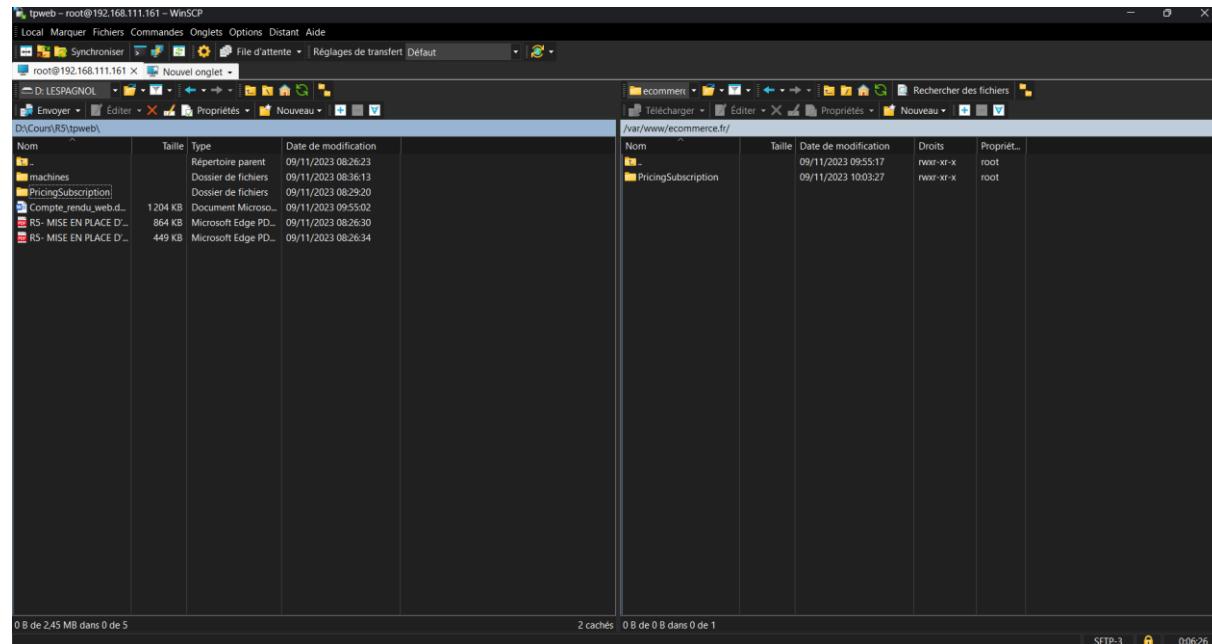
```
MariaDB [web]> exit
Bye
```

## 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    ^Y Replace     ^U Paste      ^J Justify    ^L Go To Line
```

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

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

The screenshot shows a terminal window titled "root@localhost:~". The command entered is "nano /etc/nginx/conf.d/site\_commerce.fr.conf". The file content is a Nginx configuration script:

```
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 fourni (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                               site_commerce.fr.conf                         Modified
GNU nano 5.6.1
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 :



The screenshot shows a web page titled "Subscription Sign up Form". It displays three pricing plans: Basic, Standard, and Premium. Each plan is represented by a colored box containing the price, a brief description of included features, and a "Sign Up" button.

Plan	Price	Included Features
Basic	\$5 per month	Full access, Documentation, Customers Support, Free Updates, Unlimited Domains
Standard	\$10 per month	Full access, Documentation, Customers Support, Free Updates, Unlimited Domains
Premium	\$20 per month	Full access, Documentation, Customers Support, Free Updates, Unlimited Domains

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
```



```
root@localhost:/var/www/ecommerce.fr/PricingSubscription
GNU nano 5.6.1                               sign-up.php                         Modified
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) {
    $('#signUpForm').find('input[name="Plan"]').val($(this).attr('value'));
    $('#signUpForm').find('input[name="Price"]').val($(this).attr('value'));
});
$('#signUpForm').validate({
    errorPlacement: function(error, element) {
        if (element.attr('name') == 'Terms') {
            error.insertAfter(element.parent());
        } else {
            error.insertAfter(element);
        }
    }
});
if ($('#paypalForm').length > 0) {
    $('#paypalForm').trigger('submit');
}
});
</script>
</head>
<body>
<!--header start here-->
<div class="priceing-table w3l">
    <div class="wrap">
        <h1>Inscrivez-vous maintenant</h1>
        <div class="priceing-table-main">
            <?php
                if(isset($_POST['Subscribe']) && !empty($_POST['Name'])) {
                    // Connect to database
                    $connection = new mysqli($SETTINGS["hostname"], $SETTINGS["username"], $SETTINGS["password"]);
                    if ($connection->connect_error) {
                        die('Unable to connect to MySQL server');
                    }
                    $data = array(
                        'Name' => $_POST['Name'],
                        'Email' => $_POST['Email'],
                        'Phone' => $_POST['Phone'],
                        'Address' => $_POST['Address'],
                        'City' => $_POST['City'],
                        'State' => $_POST['State'],
                        'Zip' => $_POST['Zip'],
                        'Country' => $_POST['Country'],
                        'Comments' => $_POST['Comments']
                    );
                    $query = "INSERT INTO subscribers (Name, Email, Phone, Address, City, State, Zip, Country, Comments) VALUES ('" . implode("','", $data) . "')";
                    $connection->query($query);
                    if ($connection->affected_rows > 0) {
                        echo "Inscription réussie !";
                    } else {
                        echo "Une erreur s'est produite lors de l'enregistrement.";
                    }
                }
            </?php>
        </div>
    </div>
</div>
<?php
    if(isset($_POST['Subscribe']) && !empty($_POST['Name'])) {
        // Connect to database
        $connection = new mysqli($SETTINGS["hostname"], $SETTINGS["username"], $SETTINGS["password"]);
        if ($connection->connect_error) {
            die('Unable to connect to MySQL server');
        }
        $data = array(
            'Name' => $_POST['Name'],
            'Email' => $_POST['Email'],
            'Phone' => $_POST['Phone'],
            'Address' => $_POST['Address'],
            'City' => $_POST['City'],
            'State' => $_POST['State'],
            'Zip' => $_POST['Zip'],
            'Country' => $_POST['Country'],
            'Comments' => $_POST['Comments']
        );
        $query = "INSERT INTO subscribers (Name, Email, Phone, Address, City, State, Zip, Country, Comments) VALUES ('" . implode("','", $data) . "')";
        $connection->query($query);
        if ($connection->affected_rows > 0) {
            echo "Inscription réussie !";
        } else {
            echo "Une erreur s'est produite lors de l'enregistrement.";
        }
    }
?>
```

On redémarre nginx :

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

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



The screenshot shows a web browser window with a blue header bar. The address bar displays 'Non sécurisé | 192.168.11...'. The main content features a large oval at the top containing the text 'Inscrit toi maintenant'. Below this are three colored boxes representing different subscription plans:

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

At the bottom of each plan section is a 'Sign Up' button.

## 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 sign-up.php Modified
GNU nano 5.6.1
<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) {
        $('#signUpForm').find('input[name="Plan"]').val($(this).attr('value'));
        $('#signUpForm').find('input[name="Price"]').val($(this).attr('value'));
    });
    $('#signUpForm').validate({
        errorPlacement: function(error, element) {
            if (element.attr('name') == 'Terms') {
                error.insertAfter(element.parent());
            } else {
                error.insertAfter(element);
            }
        }
    });
    if ($('#paypalForm').length > 0) {
        $('#paypalForm').trigger('submit');
    }
});
</script>
</head>
<body>
<!--header start here-->
<div class="priceing-table w31">
    <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 databases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to
your MariaDB server version for the right syntax to use near '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 :



```
root@localhost:/etc/nginx/conf.d                               site_commerce.uk.conf                         Modified
GNU nano 5.6.1
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;
    }
}
```

Il faut modifier le fichier config.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 config.php
config.php css/
[root@localhost PricingSubscription]# nano config.php
```



```
root@localhost:/var/www/ecommerce.uk/PricingSubscription config.php Modified
GNU nano 5.6.1
<?php
$SETTINGS["mysql_user"]='USERWEBUK';
$SETTINGS["mysql_pass"]='web';
$SETTINGS["hostname"]='localhost';
$SETTINGS["mysql_database"]='webuk';
$SETTINGS["data_table"]='registrations';
$SETTINGS["paypal_address"]='email@domain.com';
?>

^G Help      ^O Write Out   ^W Where Is   ^K Cut      ^T Execute   ^C Location
^X Exit      ^R Read File  ^P Print     ^J Paste    ^F Find      ^L Location
```

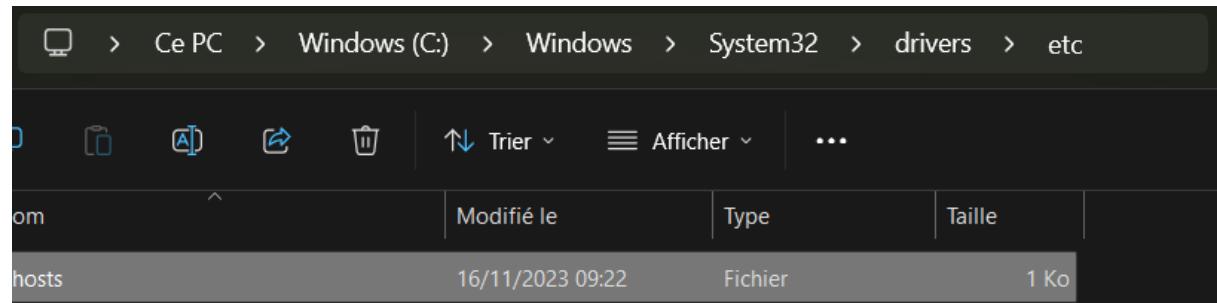
Nous allons maintenant exécuter le script SQL fourni (database.sql). Pour cela, il faut se connecter sur le moteur BDD et sélectionner la base. Nous pouvons exécuter le script en faisant source emplacement\_du\_script :

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

Pour changer les résolution dns local :



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

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97    rhino.acme.com        # source server
#      38.25.63.10    x.acme.com            # x client host
#
# localhost name resolution is handled within DNS itself.
#      127.0.0.1    localhost
#      ::1         localhost
192.168.111.161 ecommerce.fr
192.168.111.161 ecommerce.uk
```

Et voilà nous avons bien notre site en uk:



The screenshot shows a web browser window with a dark blue header bar. The title bar includes the text "R5- MISE EN PLACE D'UN SE" and "Pricing Plans and Subscription". Below the header, the address bar shows a warning icon "Non sécurisé" and the URL "ecommerc...". The main content area has a dark blue background and features a large white title "Subscription Sign Up Form". Below the title, there are three colored boxes representing different subscription plans: "Basic" (yellow), "Standard" (purple), and "Premium" (green). Each plan box displays its price (\$5, \$10, or \$20 per month) and a list of included features: Full access, Documentation, Customers Support, Free Updates, and Unlimited Domains. At the bottom of each plan box is a "Sign Up" button.

Plan	Price	Features
Basic	\$5 per month	Full access, Documentation, Customers Support, Free Updates, Unlimited Domains
Standard	\$10 per month	Full access, Documentation, Customers Support, Free Updates, Unlimited Domains
Premium	\$20 per month	Full access, Documentation, Customers Support, Free Updates, Unlimited Domains



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
[root@localhost nginx]#
```

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/
```



```
GNU nano 5.6.1          /etc/nginx/blockip.conf      Modified
deny 192.168.111.1;
```

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

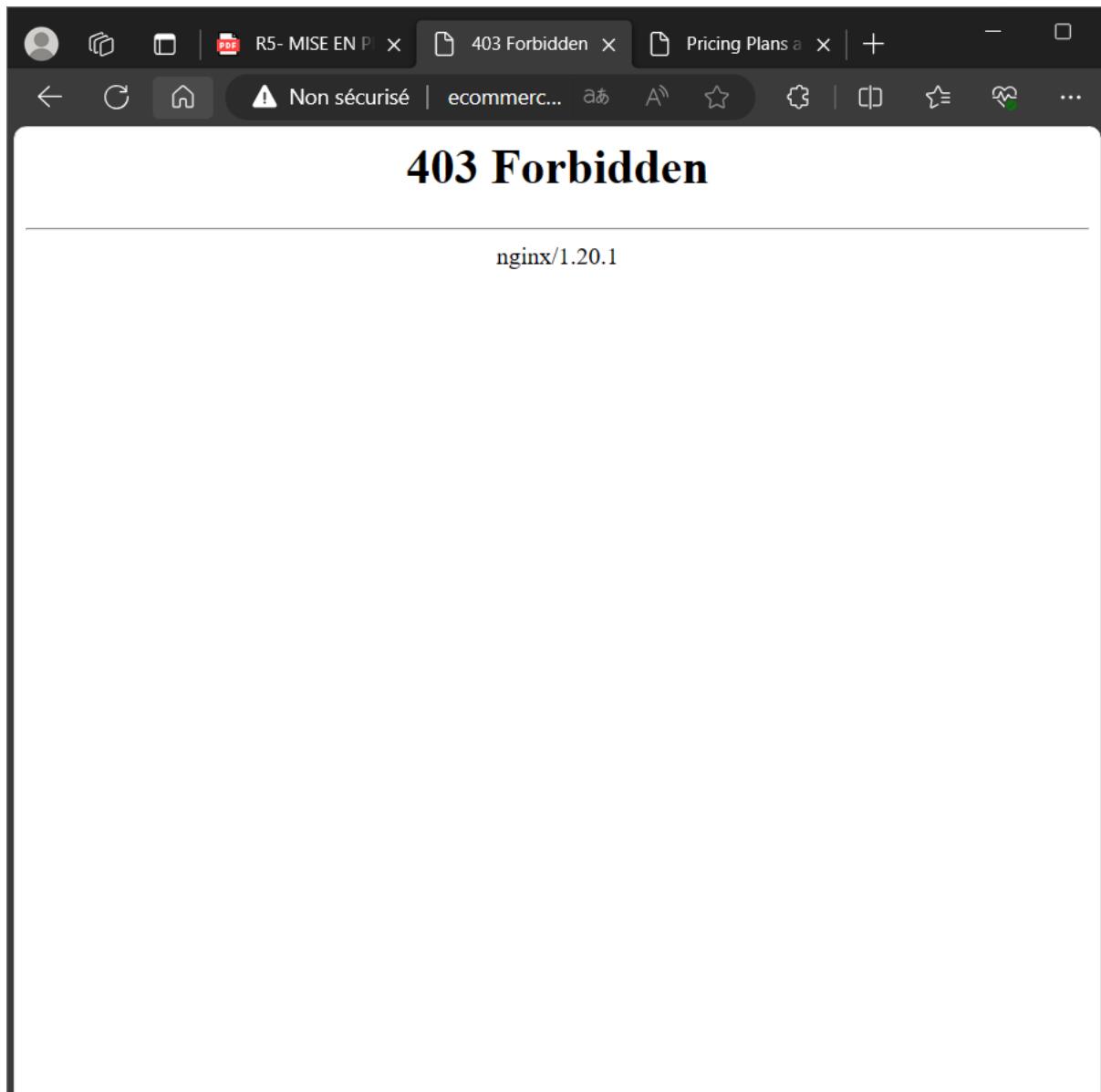
    root /var/www/ecommerce.fr/PricingSubscription/;
    index index.html index.htm index.nginx-debian.html sign-up.php;

    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 (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/2023:09:16:18 +0100] "GET /icons/poweredsby.png HTTP/1.1" 200 15443 "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/2023:09:16:18 +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://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/2023:09:26:42 +0100] "GET / HTTP/1.1" 200 7620 "-" "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 /icons/poweredsby.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://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" "
[root@localhost nginx]# 
```

Pour nous simplifier la lecture, nous utiliserons Goaccesss , 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]#
```

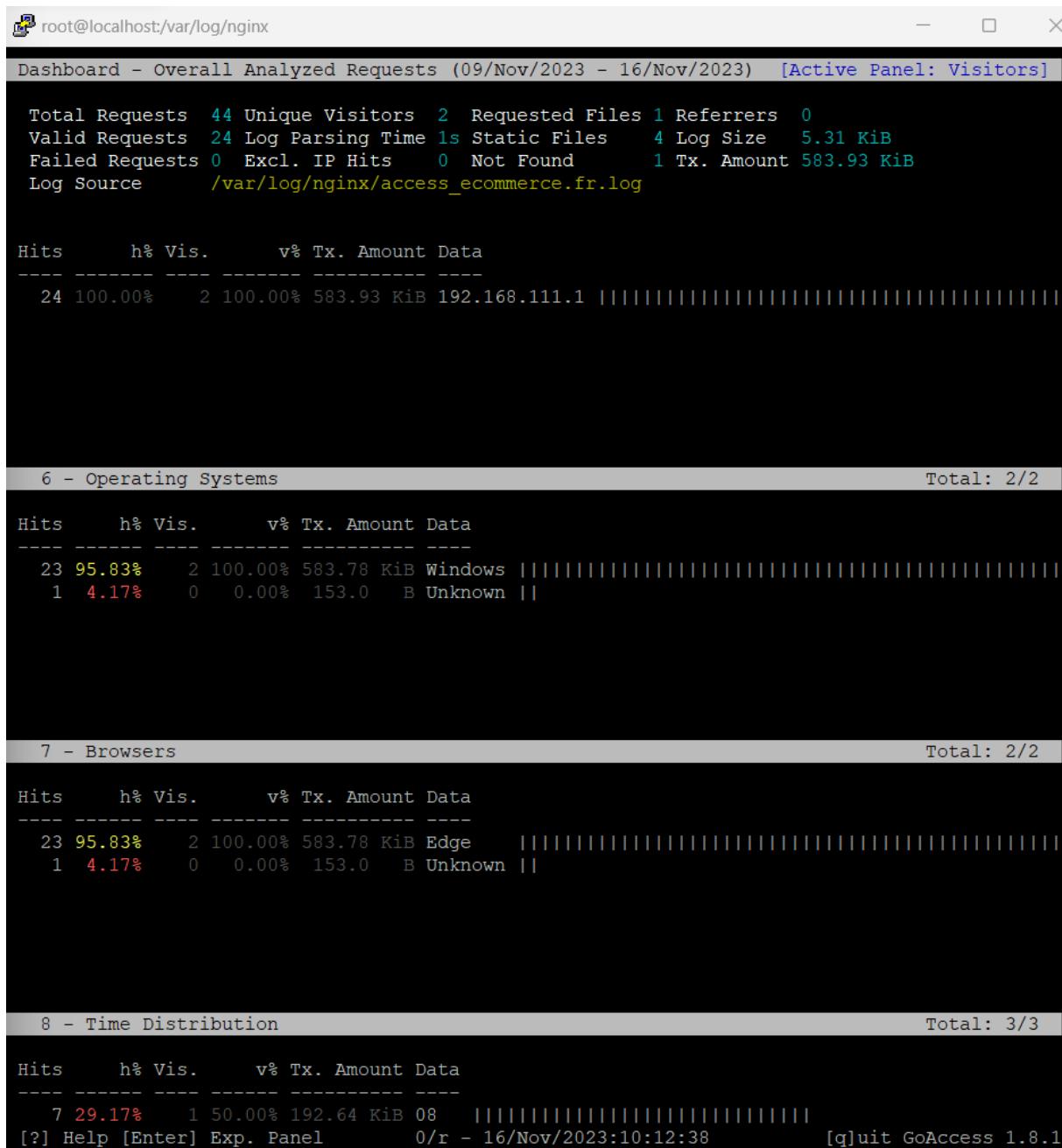
Pour utiliser le goaccess, il faut exécuter la commande ci-contre :

```
[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 :

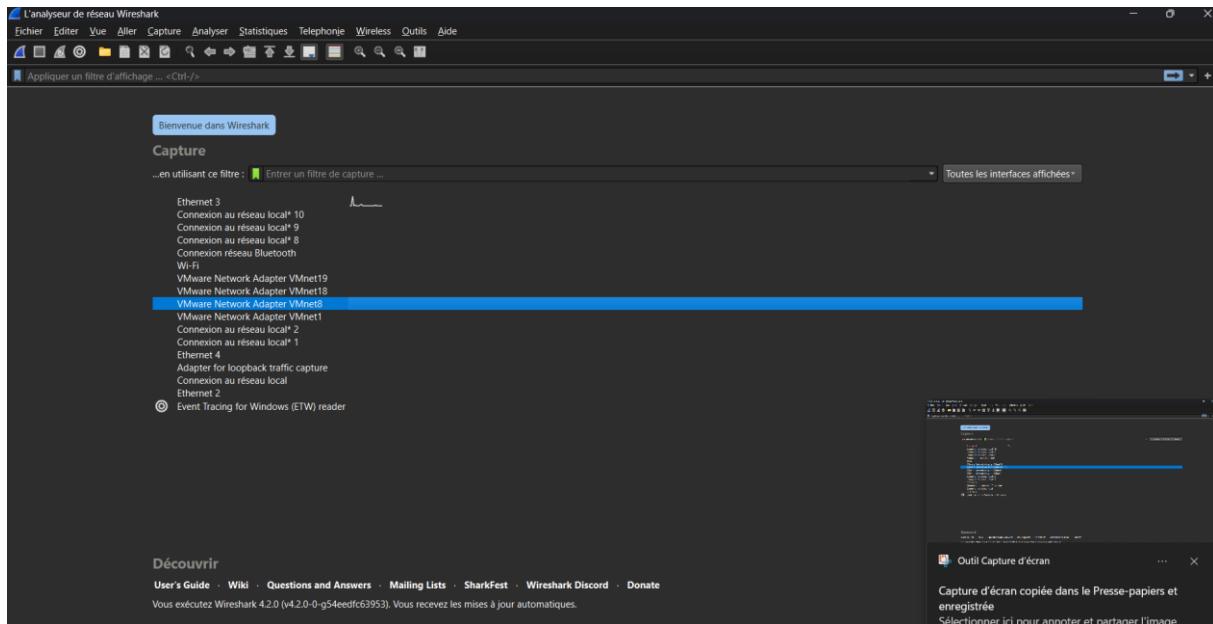


```
+-- Log Format Configuration
| [SPACE] to toggle - [ENTER] to proceed - [q] to quit
|
| [x] NCSA Combined Log Format
| [ ] NCSA Combined Log Format with Virtual Host
| [ ] Common Log Format (CLF)
| [ ] Common Log Format (CLF) with Virtual Host
| [ ] W3C
| [ ] CloudFront (Download Distribution)
|
| 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
+
```

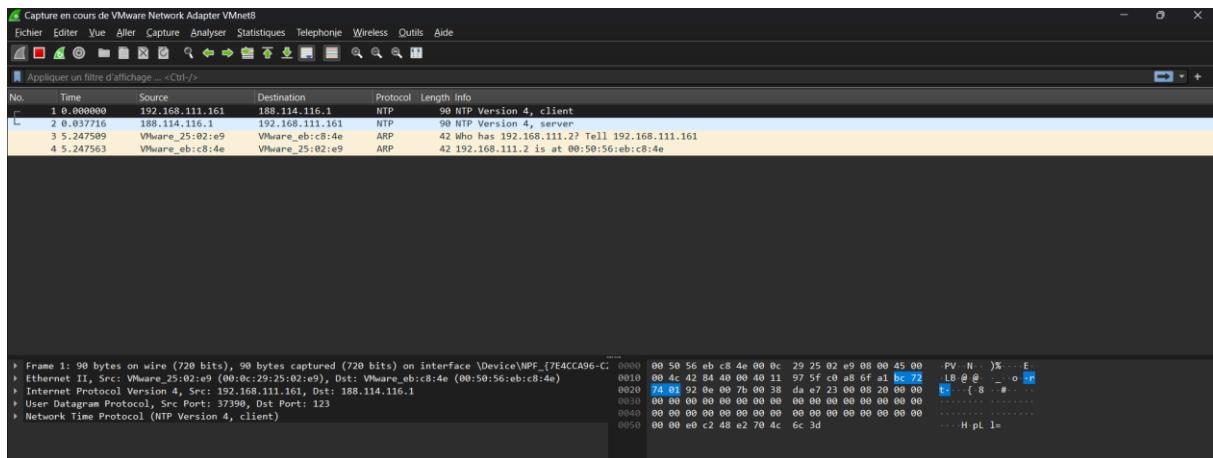


## Analyse du site en http

On lance wireshark on sélectionne notre carte réseau :



On entre sur cette page :



Je fais une simulation d'inscription :



Gaston  
berger

LESPAGNOL  
Alexandre  
2SISR

09/11/2023

Subscription Sign Up Form

Sign Up

alexandre

alex@gmail.com

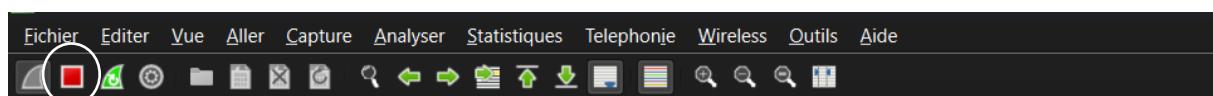
....

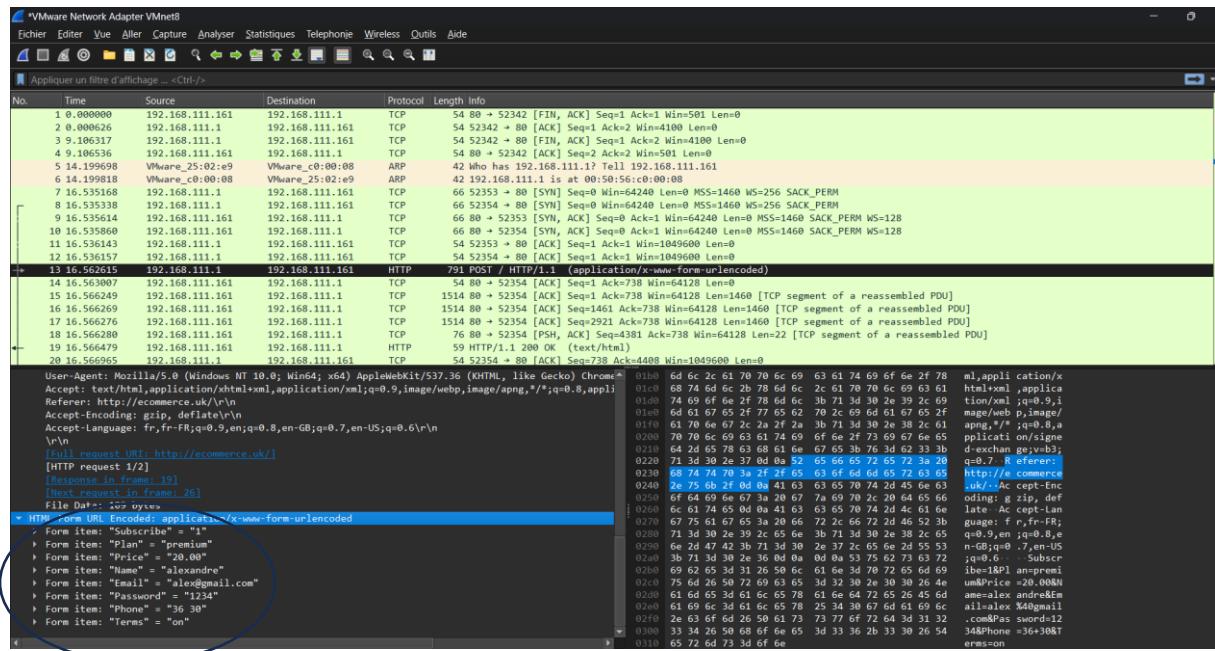
36 30

I Accept Terms.

Submit

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





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
```

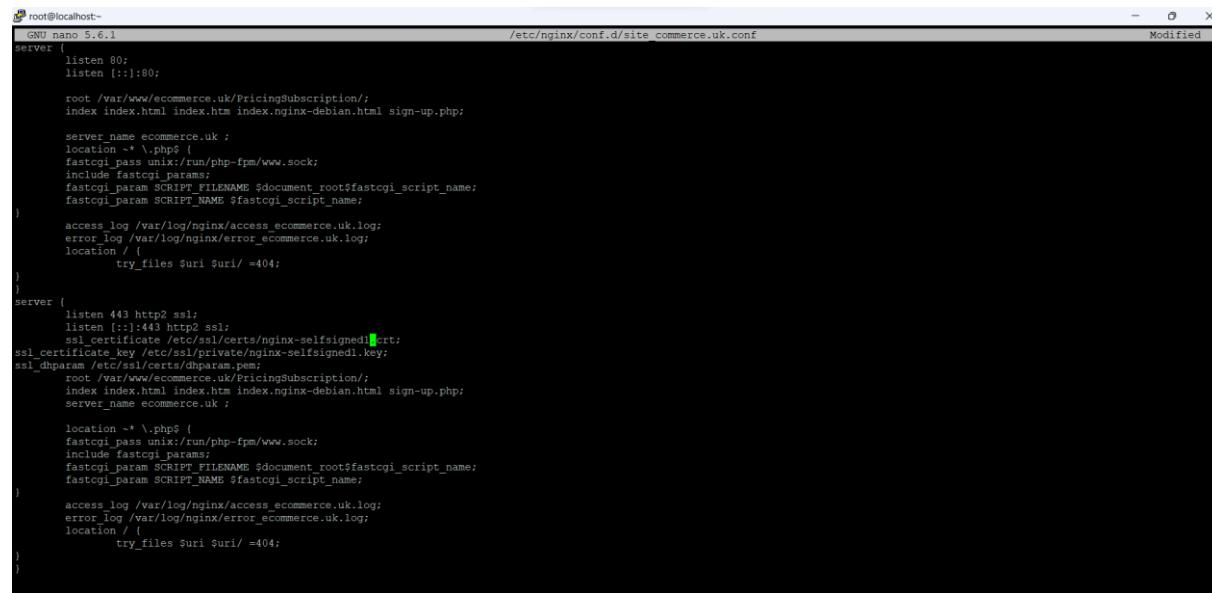
On va limiter les accès sur ce répertoire :

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

Création d'un certificat (uk) :

```
[root@localhost ~]# openssl req -x509 -nodes 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) []:
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
```



```
root@localhost ~# nano 5.6.1
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;
    }

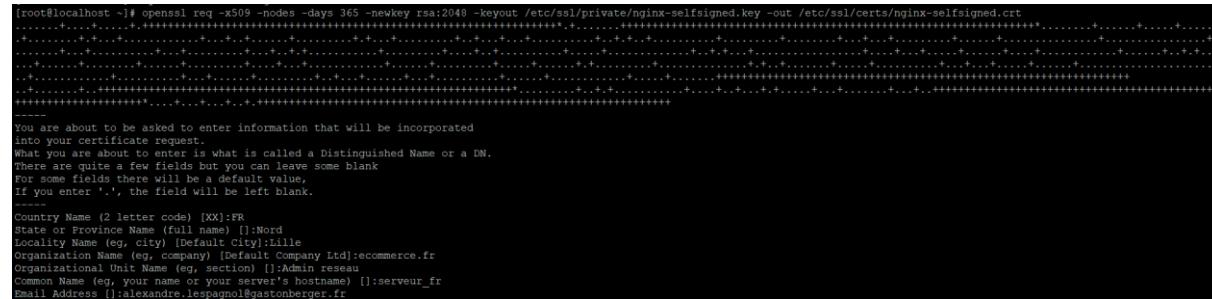
    access_log /var/log/nginx/access_ecommerce.uk.log;
    error_log /var/log/nginx/error_ecommerce.uk.log;
    location / {
        try_files $uri $uri/ =404;
    }
}

server {
    listen 443 http2 ssl;
    listen [::]:443 http2 ssl;
    ssl_certificate /etc/ssl/private/nginx-selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256;
    ssl_trusted_certificate /etc/ssl/certs/ca-certificates.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;
    }
}
```

## Création d'un certificat (fr) :



```
[root@localhost ~]# openssl req -x509 -nodes 365 -keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt
...
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
if you enter '.', the field will be left blank.
Country Name (2 letter code) [XX]:FR
State or Province Name (full name) []:Nord
Locality Name (eg, city) [Default City]:Lille
Organization Name (eg, company) [Default Company Ltd]:ecommerce.fr
Organizational Unit Name (eg, section) []:Admin reseau
Common Name (eg, your name or your server's hostname) []:serveur_fr
Email Address []:alexandre.lespagnol@gastonberger.fr
```

```
[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
server {
    listen 80;
    listen [::]:80;

    root /var/www/commerce.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/commerce.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;
    }
}
```

Nous allons maintenant générer un groupe DiffieHellman fort qui permet la négociation de PFS (Perfect 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 conf.d]#
```



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

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

    server_name e-commerce.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_e-commerce.fr.log;
    error_log /var/log/nginx/error_e-commerce.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/e-commerce.fr/PricingSubscription/;
    index index.html index.htm index.nginx-debian.html sign-up.php;
    server_name e-commerce.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_e-commerce.fr.log;
    error_log /var/log/nginx/error_e-commerce.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 dhcpcv6-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) :

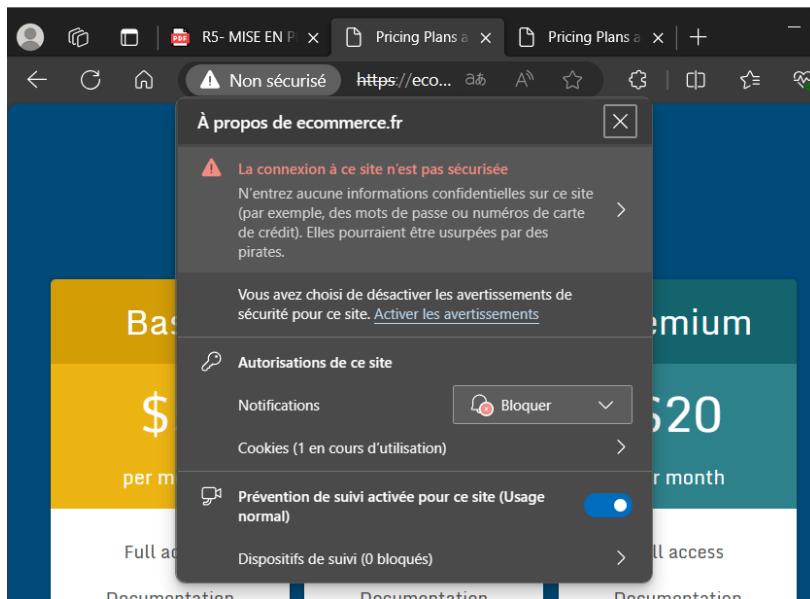
On tape dans notre navigateur <https://ecommerce.fr>



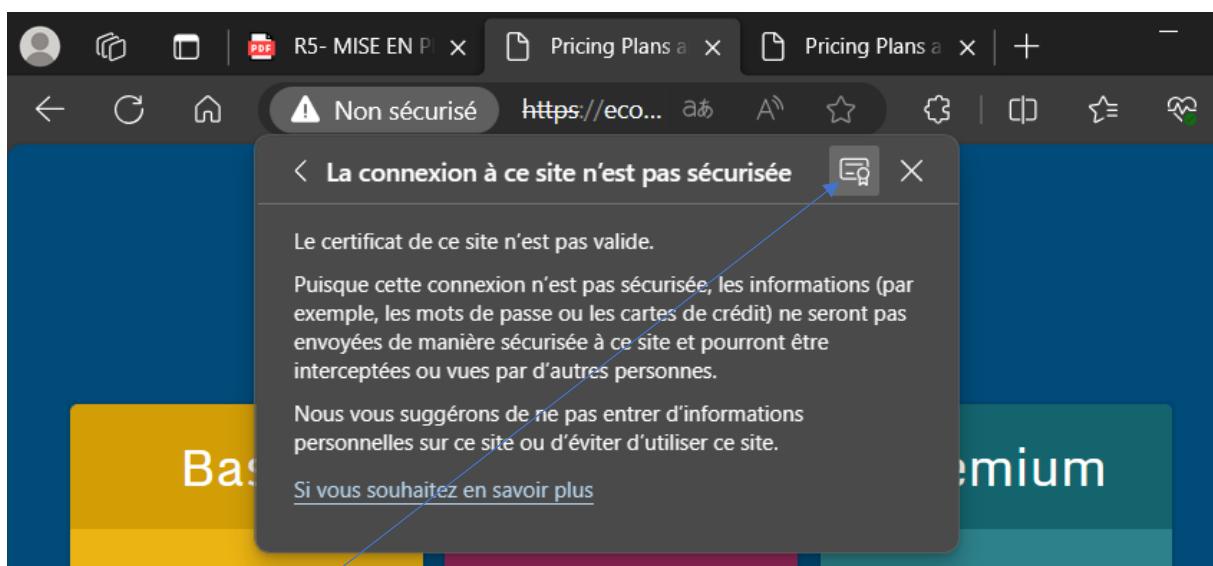
The screenshot shows a web browser window with three tabs open. The active tab displays a pricing page for a service named 'Inscrit toi man'. The page features three plan options: Basic (\$5 per month), Standard (\$10 per month), and Premium (\$20 per month). Each plan includes a list of features: Full access, Documentation, Customers Support, Free Updates, and Unlimited Domains. At the bottom of each plan section is a 'Sign Up' button.

Plan	Price	Features
Basic	\$5 per month	Full access, Documentation, Customers Support, Free Updates, Unlimited Domains
Standard	\$10 per month	Full access, Documentation, Customers Support, Free Updates, Unlimited Domains
Premium	\$20 per month	Full access, Documentation, Customers Support, Free Updates, Unlimited Domains

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 :



The screenshot shows a browser window with three tabs open. The active tab is titled "Visionneuse de certificats : serveur\_fr". The content of the page is as follows:

Général	
Émis pour	
Nom commun (CN)	serveur_fr
Organisation (O)	ecommerce.fr
Unité d'organisation (UO)	Admin reseau
Émis par	
Nom commun (CN)	serveur_fr
Organisation (O)	ecommerce.fr
Unité d'organisation (UO)	Admin reseau
Période de validité	
Date d'émission	jeudi 23 novembre 2023 à 16:21:10
Date d'expiration	vendredi 22 novembre 2024 à 16:21:10
Empreintes digitales SHA-256	
Certifikat	a5529e59ebaae63355dbd180afbe6ebb1ff36226a282e4125af817ace3ba75f9
Clé publique	07bbe1edbe810879c09079adf51dbdb16515a70a4419441cc89d24d07f394c4a

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

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

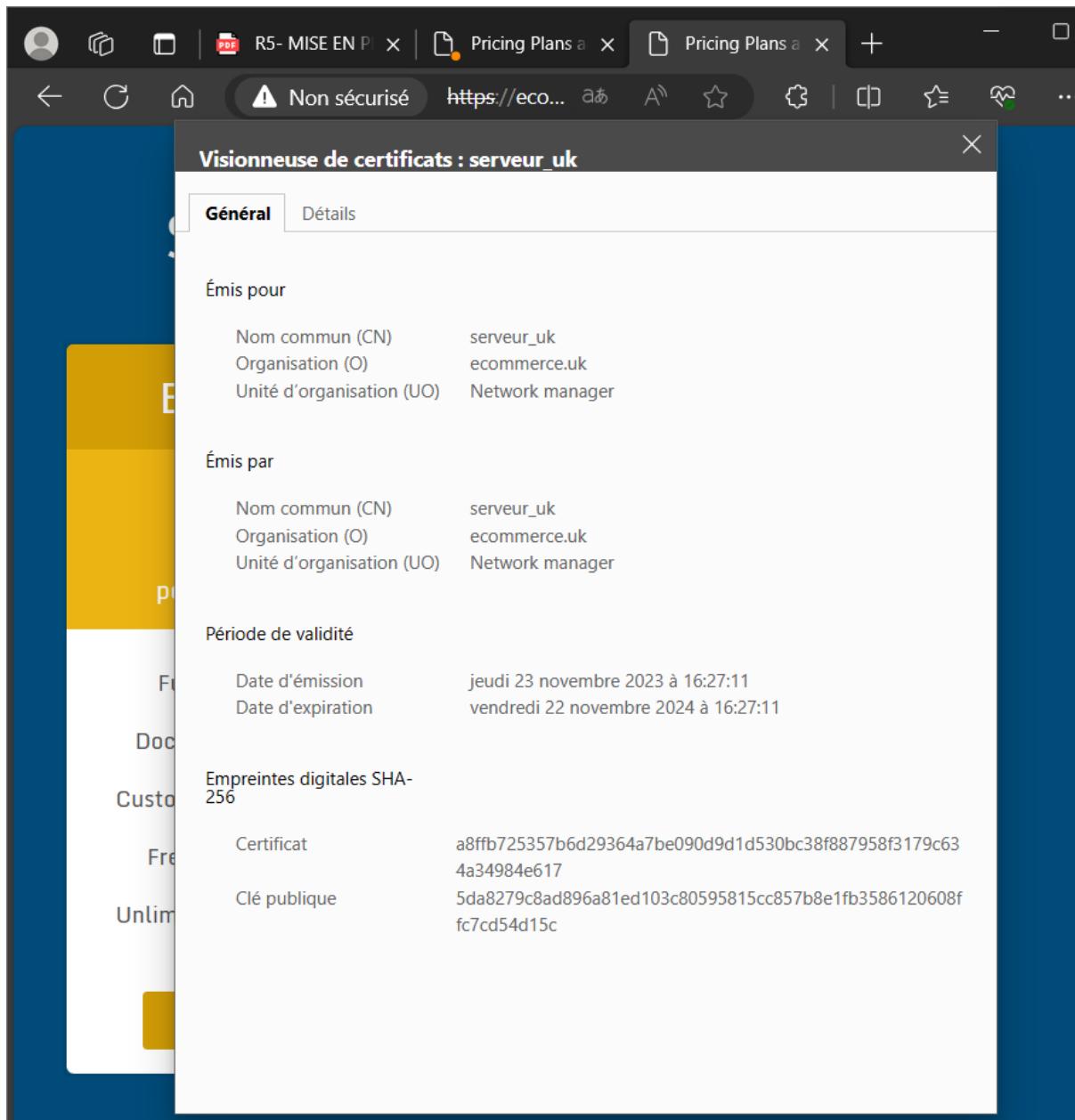


The screenshot shows a "Subscription Sign Up Form" with three pricing plans:

Plan	Price	Per Month	Features
Basic	\$5	per month	Full access Documentation Customers Support Free Updates Unlimited Domains
Standard	\$10	per month	Full access Documentation Customers Support Free Updates Unlimited Domains
Premium	\$20	per month	Full access Documentation Customers Support Free Updates Unlimited Domains

Each plan has a corresponding "Sign Up" button at the bottom.

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 :



```
23 10.728331 192.168.111.1 192.168.111.161 TLSv1.3 921 Client Hello (SNI=ecommerce
Source Port: 443
Destination Port: 56030
[Stream index: 0]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 238]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2724195459
[Next Sequence Number: 239 (relative sequence number)]
Acknowledgment Number: 868 (relative ack number)
Acknowledgment number (raw): 201080803
0101 .... = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
Window: 501
[Calculated window size: 64128]
[Window size scaling factor: 128]
Checksum: 0xbfea [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
```

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 :

```
[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
```

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



```
root@localhost:/etc/nginx/conf.d          site_commerce.fr.conf      Modified
GNU nano 5.6.1
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:~# nano /etc/nginx/conf.d/site_commerce.uk.conf
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 :

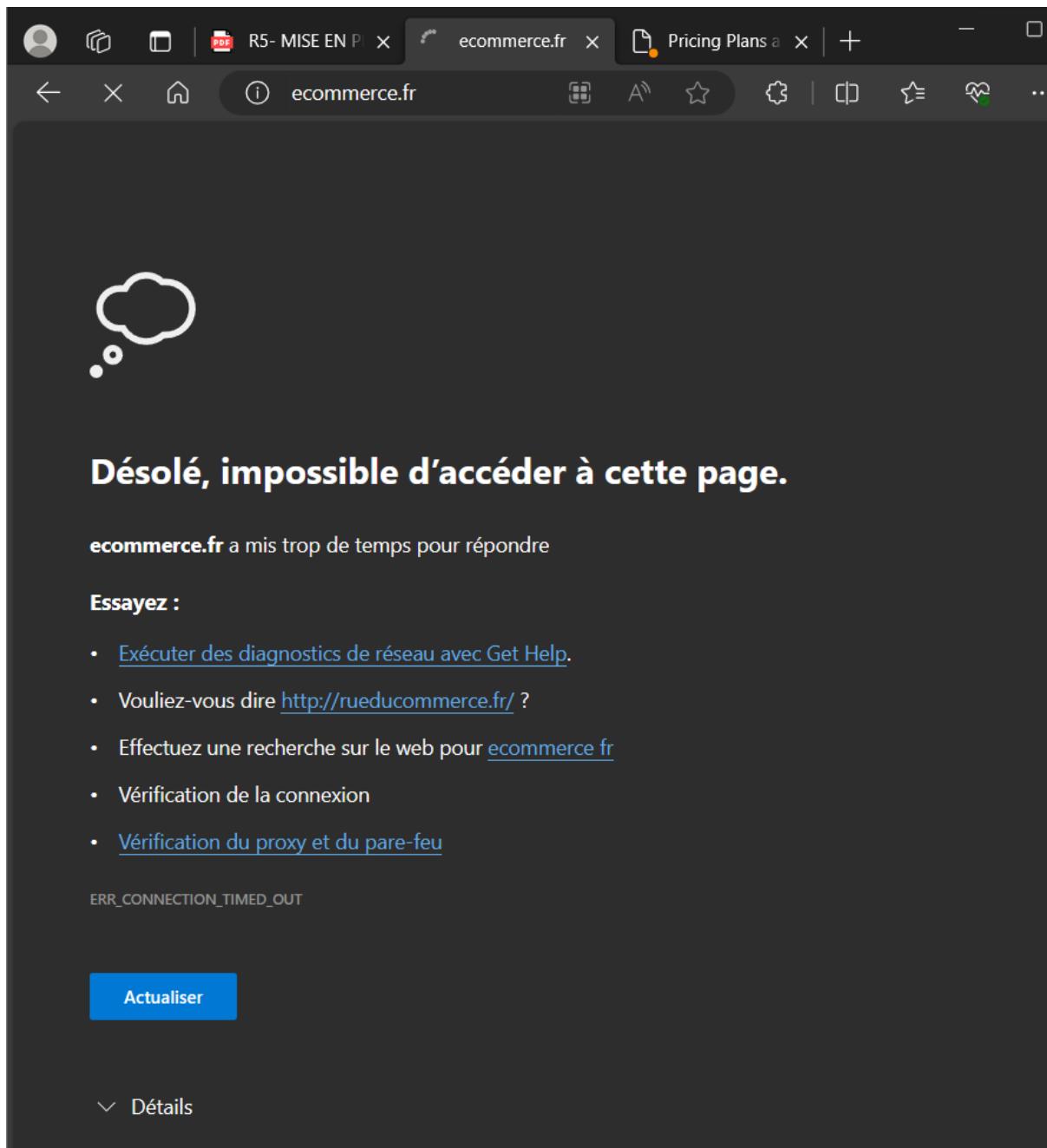
```
[root@localhost ~]# nano /etc/nginx/conf.d/site_commerce.uk.conf
[root@localhost ~]# firewall-cmd --remove-service=http
success
```

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 d'une VM HA01.



## Installation d'haproxy sur la machine HA01 :

```
Iroot@localhost ~]# dnf install haproxy -y
Rocky Linux 9 - BaseOS
Rocky Linux 9 - AppStream
Rocky Linux 9 - Extras
Dependencies resolved:
=====
| Package           | Architecture | Version      | Repository | Size |
| ====== | =========== | ============ | =========== | =:= |
| Installing:     |             |             |             |       |
| haproxy          | x86_64      | 2.4.22-1.e19 | 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.e19.x86_64.rpm 7.5 MB/s | 2.2 MB  00:00
Total
Rocky Linux 9 - AppStream
Importing GPG key 0x3500275D:
Userid   : "Rocky Enterprise Software Foundation - Release key 2022 <rele...@rockylinux.org>"  

Fingerprint: Z1CB 256A E16F C54C 6E65 2949 7B2D 426D 3500 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 :
  Running scriptlets: haproxy-2.4.22-1.e19.x86_64 1/1
  Installing  : haproxy-2.4.22-1.e19.x86_64 1/1
  Running scriptlets: haproxy-2.4.22-1.e19.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.e19.x86_64 1/1
Installed:
  haproxy-2.4.22-1.e19.x86_64 1/1
Complete!
Iroot@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 ecommerce.uk on remarque bien que c notre service\_web02 qui nous donne la page :

The screenshot shows a browser window with two tabs open, both titled "Pricing Plans". The left tab is "Non sécurisé" and the right tab is "https://ecommerce.fr". Below the tabs, there's a navigation bar with icons for back, forward, search, and refresh. The main content area displays a table of plans:

Plan	Prix	Caractéristiques
Basic	\$5 per month	Full access Documentation Customers Support Free Updates Unlimited Domains
Standard	\$10 per month	Full access Documentation Customers Support

Below the table, there are "Sign Up" buttons for each plan.

On the right side of the screen, there's a terminal window titled "root@localhost:~". It shows a log entry from MariaDB:

```
Dec 06 13:33:14 localhost systemd[1]: Starting MariaDB 10.5 database server...
Dec 06 13:33:15 localhost mariadb-check-socket[866]: Socket file /var/lib/mysql/
Dec 06 13:33:15 localhost.localdomain mariadb-check-socket[866]: No process is using /var/lib/mysql/
Dec 06 13:33:16 localhost.localdomain mariadb-prepare-db-dir[912]: Database MariaDB is probab
Dec 06 13:33:16 localhost.localdomain mariadb-prepare-db-dir[912]: If this is not the case, y
Dec 06 13:33:19 localhost.localdomain systemd[1]: Started MariaDB 10.5 database server.
lines 1-22
```

Below the log, there's a command-line interface with several commands entered:

```
[root@localhost ~]# *c
[root@localhost ~]# *c
[root@localhost ~]# *
[root@localhost ~]# 
```

At the bottom of the terminal, there's a message:

```
root@ServiceWeb2:~
```

login as: root
root@192.168.111.128's password:
Activate the web console with: systemctl enable --now cockpit.socket

At the very bottom, there's a message from the system:

```
Last login: Wed Dec 6 13:33:18 2023
[root@ServiceWeb2 ~]# systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:mariadb(8)
              https://mariadb.com/kb/en/library/systemd/
[root@ServiceWeb2 ~]# nano /etc/nginx/conf.d/php-fpm.conf
site_commerce.fr.conf site_commerce.uk.conf
[root@ServiceWeb2 ~]# nano /etc/nginx/conf.d/site_commerce.fr.conf
site_commerce.uk.conf
[root@ServiceWeb2 ~]# nano /etc/nginx/conf.d/site_commerce.fr.conf
[root@ServiceWeb2 ~]# nano /etc/nginx/conf.d/site_commerce.uk.conf
[root@ServiceWeb2 ~]# 
```

## 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 ~]# 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 dhcpcv6-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
```

```
GNU nano 5.6.1          /etc/my.cnf.d/mariadb-server.cnf      Modified
pid-file=/run/mariadb/mariadb.pid

#
# * Galera-related settings
#
[galera]
# Mandatory settings
#wsrep_on=ON
#wsrep_provider=
#wsrep_cluster_address=
#binlog_format=row
#default_storage_engine=InnoDB
#innodb_autoinc_lock_mode=2
#
# Allow server to accept connections on all interfaces.
#
bind-address=0.0.0.0
#
```

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

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
[root@localhost ~]#
```

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 PricingSubscription]#
```

root@ServiceWeb2:/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"]='192.168.111.161';		
\$SETTINGS["mysql_database"]='web';		
\$SETTINGS["data_table"]='registrations';		
\$SETTINGS["paypal_address"]='email@domain.com';		
?>		

Modifier ceci dans les deux sites :

```
config.php    CSS/      database.sql   images/    JS/      sign-up.php
[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 :



```
[root@ServiceWeb2 ~]# mysql -u USER-web -h 192.168.111.161 -p
Enter password:
ERROR 2002 (HY000): Can't connect to MySQL server on '192.168.111.161' (115)
[root@ServiceWeb2 ~]# mysql -u USER-web -h 192.168.111.161 -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.5.22-MariaDB MariaDB Server

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

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

MariaDB [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| web           |
+-----+
2 rows in set (0.002 sec)

MariaDB [(none)]>
```

## 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. :

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

Nous allons modifier les paramètres pour ajouter nos serveurs :

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



```
#-----  
# 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        redispach  
    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" SysSock.Use="off") # provides support for local system logging (e.g. via logger command)
# Turn off message reception via local log socket;
# local messages are retrieved through imjournal now.
module(load="imjournal" 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
 # highlighted by a blue oval

# 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

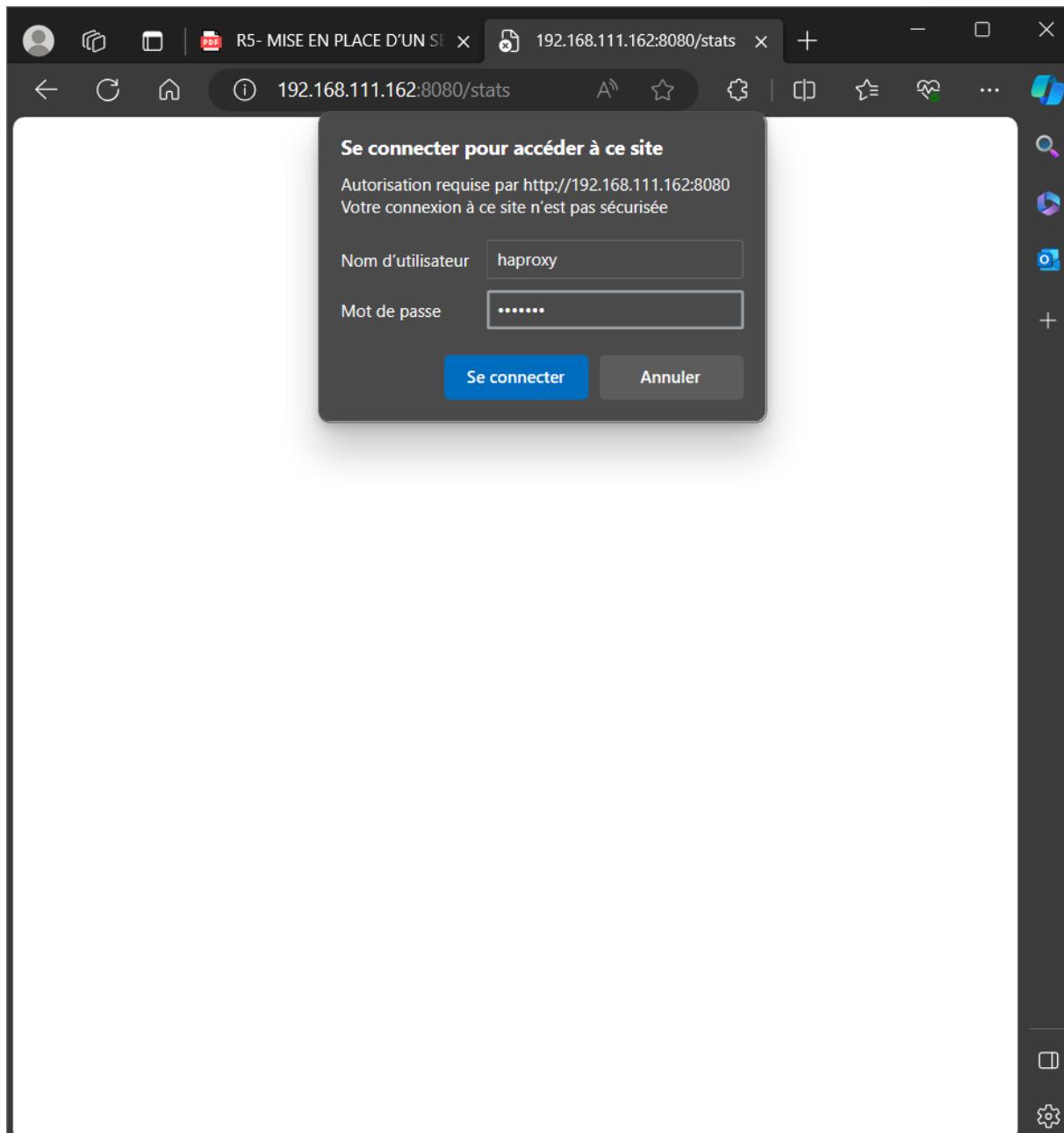
On ouvre le port 8080 :

```
[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 dhcpcv6-client ssh
  ports: 8080/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost ~]#
```

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



**HAProxy**

**Statistics Report for pid 1407**

> General process information

pid = 1407 (process #1, nbproc = 1, nbthread = 1)  
 uptime = 0d 0h17m23s  
 system limits: memmax = unlimited; ulimit-n = 8034  
 maxsock = 8034; maxconn = 4000; maxpipes = 0  
 current connns = 1; current pipes = 0/0; conn rate = 1/sec; bit rate = 5.166 kbps  
 Running tasks: 0/19; idle = 100 %

Display option:

- Scope :
- [Hide 'DOWN'](#)
- [Refresh now](#)
- [CSV export](#)
- [JSON export](#)

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

stats												Server													
	Queue			Session rate			Sessions			Bytes		Denied		Errors		Warnings									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	
Frontend				1	2	-	1	2	3 000	3			469	263	0	0	1					OPEN			
Backend	0	0		0	0		0	0	300	0	0	0s	469	263	0	0		0	0	0	0	17m23s UP		0/0	0

main												Server													
	Queue			Session rate			Sessions			Bytes		Denied		Errors		Warnings									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	
Frontend				0	0	-	0	0	3 000	0			0	0	0	0	0					OPEN			

static												Server												
	Queue			Session rate			Sessions			Bytes		Denied		Errors		Warnings								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	
static	0	0	-	0	0		0	0	-	0	0	?	0	0		0	0	0	0	0	17m23s DOWN	L4CON in 0ms	1/1	
Backend	0	0		0	0		0	0	300	0	0	?	0	0	0	0	0	0	0	0	17m23s DOWN		0/0	

app												Server													
	Queue			Session rate			Sessions			Bytes		Denied		Errors		Warnings									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk		
app1	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	17m22s DOWN	L4CON in 0ms		
app2	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	17m22s DOWN	L4CON in 0ms		
app3	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	17m22s DOWN	L4CON in 0ms		
app4	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	17m21s DOWN	L4CON in 0ms		
service_web	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	17m23s UP	L6OK in 3ms		
service_web02	0	0	-	0	0		0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	17m23s UP	L6OK in 2ms		
Backend	0	0		0	0		0	0	300	0	0	?	0	0	0	0	0	0	0	0	0	17m23s UP			

## Sécurisation

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



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

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



```
root@localhost:~          /etc/haproxy/haproxy.cfg      Modified |  
GNU nano 5.6.1  
#-----  
# Example configuration for a possible web application. See the  
# full configuration options online.  
#  
#     https://www.haproxy.org/download/1.8/doc/configuration.txt  
#-----  
#-----  
# Global settings  
#-----  
global  
    # to have these messages end up in /var/log/haproxy.log you will  
    # need to:  
    #  
    # 1) configure syslog to accept network log events. This is done  
    # by adding the '-r' option to the SYSLOGD_OPTIONS in  
    # /etc/sysconfig/syslog  
    #  
    # 2) configure local2 events to go to the /var/log/haproxy.log  
    # file. A line like the following can be added to  
    # /etc/sysconfig/syslog  
    #  
    #     local2.*           /var/log/haproxy.log  
    #  
log      127.0.0.1 local2  
  
chroot   /var/lib/haproxy  
pidfile  /var/run/haproxy.pid  
maxconn  4000  
user     haproxy  
group    haproxy  
daemon  
    maxsslconn 256  
    tune.ssl.default-dh-param 2048  
  
#-----  
# main frontend which proxys to the backends  
#-----  
frontend main  
    bind *:80  
    acl url_static      path_beg      -i /static /javascript /stylesheets  
    acl url_static      path_end      -i .jpg .gif .png .css .js  
  
    use_backend static      if url_static  
    default_backend app  
        bind *:443 ssl crt /etc/pki/tls/certs/haproxy.pem
```

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 dhcpcv6-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 :



Alexandre LESPAGNOL AL

InPrivate Non sécurisé | https://192.168.111.162

Visionneuse de certificats : haproxy

Général Détails

Émis pour

Nom commun (CN)	haproxy
Organisation (O)	haproxy
Unité d'organisation (UO)	admin

Émis par

Nom commun (CN)	haproxy
Organisation (O)	haproxy
Unité d'organisation (UO)	admin

Période de validité

Date d'émission	jeudi 7 décembre 2023 à 09:03:52
Date d'expiration	vendredi 6 décembre 2024 à 09:03:52

Empreintes digitales SHA-256

Certifikat	aad9ad673da4024132c2961111f00c253a35abe5fb2d63b9c20930cb931a6564
Clé publique	e01a7d5118882479effc4d1aea3c601f0429fbf233c4423e0b10538d4d4a496a

## 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 et le service web 2 en vert :



app	System Metrics												Resource Utilization																			
	Queue			Session			Sessions			Bytes			Denied			Errors			Warnings			Status										
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Ret	Redis	Chk	Server	Wght	Act	Bck	Chk	Dwn	Downtime	Thrshld			
app1	0	-	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	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	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	0	0	13m50s	DOWN	L4CON in 0ms	1/1	Y	-	1	1	13m50s	-	
service_web	0	0	-	0	2	-	0	3	-	5	5	5m30s	229 265	129 282	0	0	0	0	0	0	0	3m2s	DOWN	*LATOUT in 2000ms	1/1	Y	-	0	3	1	3m2s	-
service_web2	0	0	-	0	3	-	3	3	-	7	7	44s	4 217	218 765	0	0	0	0	0	0	0	13m1s	UP	L6CON in 3ms	1/1	Y	-	0	0	0	0s	-
Backend	0	0	-	0	4	-	3	5	300	12	12	44s	233 482	348 647	0	0	0	0	0	0	0	13m1s	UP	L1CON in 0ms	1/1	Y	-	0	0	0	0	-

## 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
```



The screenshot shows two terminal windows side-by-side. Both windows are running the nano text editor on the /etc/hosts file. The top window is titled "root@ServiceWeb:~" and the bottom window is titled "root@ServiceWeb2:~". Both windows display the same content:

```
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
192.168.111.130 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
```

The bottom window shows the command-line interface with the following prompt and command:

```
[root@HAProxy02 ~]# curl -k https://ecommerce.fr
```

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

```
[root@HAProxy02 ~]# curl -k https://ecommerce.fr
```



```
[root@HAProxy02 ~]# curl -k https://ecommerce.ti
<!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) {
                $('#signUpForm').find('input[name="Plan"]').val($(this).data("plan"));
                $('#signUpForm').find('input[name="Price"]').val($(this).data("price"));
            });
            $('#signUpForm').validate({
                errorPlacement: function(error, element) {
                    if (element.attr('name') == 'Terms') {
                        error.insertAfter(element.parent());
                    }
                }
            });
        });
    </script>
```

## Installation de Pacemaker

Sur chaque serveur HA proxy, il est nécessaire de :



```
root@Haproxy:~  
[root@Haproxy ~]# login as: root  
[root@192.168.111.162's password:  
Last login: Thu Dec 14 08:01:55 2023  
[root@Haproxy ~]# dnf --enablerepo=HighAvailability -y install pacemaker pcs  
Error: Unknown repo: 'HighAvailability'  
[root@Haproxy ~]# ^C  
[root@Haproxy ~]# dnf --enablerepo=HighAvailability -y install pacemaker pcs  
Error: Unknown repo: 'HighAvailability'  
[root@Haproxy ~]# dnf --enablerepo=highavailability -y install pacemaker pcs  
Rocky Linux 9 - BaseOS 12 kB/s | 4.1 kB 00:00  
Rocky Linux 9 - BaseOS 3.0 MB/s | 2.2 MB 00:00  
Rocky Linux 9 - AppStream 1.0 kB/s | 4.5 kB 00:04  
Rocky Linux 9 - AppStream 9.4 MB/s | 7.4 MB 00:00  
Rocky Linux 9 - High Availability 498 kB/s | 246 kB 00:00  
Rocky Linux 9 - Extras 9.4 kB/s | 2.9 kB 00:00  
Dependencies resolved.  
=====  
Package Arch Version Repository Size  
=====  
Installing:  
pacemaker x86_64 2.1.6-10.1.el9_3 highavailability 459 k  
pcs x86_64 0.11.6-3.el9 highavailability 7.8 M  
Upgrading:  
root@Haproxy02:~  
[root@Haproxy02 ~]# login as: root  
[root@192.168.111.130's password:  
Last login: Thu Dec 14 08:02:07 2023  
[root@Haproxy02 ~]# dnf --enablerepo=highavailability -y install pacemaker pcs  
Rocky Linux 9 - High Availability 597 kB/s | 246 kB 00:00  
Dependencies resolved.  
=====  
Package Arch Version Repository Size  
=====  
Installing:  
pacemaker x86_64 2.1.6-10.1.el9_3 highavailability 459 k  
pcs x86_64 0.11.6-3.el9 highavailability 7.8 M  
Upgrading:  
keyutils-libs x86_64 1.6.3-1.el9 baseos 31 k  
libsss_certmap x86_64 2.9.1-4.el9_3.1 baseos 93 k  
libsss_idmap x86_64 2.9.1-4.el9_3.1 baseos 44 k  
libsss_nss_idmap x86_64 2.9.1-4.el9_3.1 baseos 48 k  
libsss_sudo x86_64 2.9.1-4.el9_3.1 baseos 38 k  
nspr x86_64 4.35.0-3.el9_2 appstream 134 k  
nss x86_64 3.90.0-3.el9_2 appstream 704 k  
nss-softokn x86_64 3.90.0-3.el9_2 appstream 380 k  
nss-softokn-freebl x86_64 3.90.0-3.el9_2 appstream 326 k  
nss-sysinit x86_64 3.90.0-3.el9_2 appstream 19 k
```

On active le démarrage sur les deux serveurs haproxy :



The screenshot shows two terminal windows side-by-side. Both windows are running on the root user of Haproxy servers.

**Terminal 1 (Haproxy):**

```
[root@Haproxy:~]
quota-nls-1:4.06-6.el9.noarch
resource-agents-4.10.0-44.el9_3.0.1.x86_64
rocky-logos-90.14-2.el9.x86_64
rpcbind-1.2.6-5.el9.x86_64
ruby-3.0.4-160.el9_0.x86_64
ruby-default-gems-3.0.4-160.el9_0.noarch
ruby-libs-3.0.4-160.el9_0.x86_64
rubygem-bigdecimal-3.0.0-160.el9_0.x86_64
rubygem-bundler-2.2.33-160.el9_0.noarch
rubygem-io-console-0.5.7-160.el9_0.x86_64
rubygem-json-2.5.1-160.el9_0.x86_64
rubygem-psych-3.3.2-160.el9_0.x86_64
rubygem-rdoc-6.3.3-160.el9_0.noarch
rubygem-rexml-3.2.5-160.el9_0.noarch
rubygems-3.2.33-160.el9_0.noarch
sssd-nfs-idmap-2.9.1-4.el9_3.1.x86_64
tar-2:1.34-6.el9_1.x86_64

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

**Terminal 2 (Haproxy02):**

```
[root@Haproxy02:~]
quota-nls-1:4.06-6.el9.noarch
resource-agents-4.10.0-44.el9_3.0.1.x86_64
rocky-logos-90.14-2.el9.x86_64
rpcbind-1.2.6-5.el9.x86_64
ruby-3.0.4-160.el9_0.x86_64
ruby-default-gems-3.0.4-160.el9_0.noarch
ruby-libs-3.0.4-160.el9_0.x86_64
rubygem-bigdecimal-3.0.0-160.el9_0.x86_64
rubygem-bundler-2.2.33-160.el9_0.noarch
rubygem-io-console-0.5.7-160.el9_0.x86_64
rubygem-json-2.5.1-160.el9_0.x86_64
rubygem-psych-3.3.2-160.el9_0.x86_64
rubygem-rdoc-6.3.3-160.el9_0.noarch
rubygem-rexml-3.2.5-160.el9_0.noarch
rubygems-3.2.33-160.el9_0.noarch
sssd-nfs-idmap-2.9.1-4.el9_3.1.x86_64
tar-2:1.34-6.el9_1.x86_64

Complete!
[root@Haproxy02 ~]# systemctl enable --now pcasd
Created symlink /etc/systemd/system/multi-user.target.wants/pcasd.service → /usr/lib/systemd/system/pcasd.service.
[root@Haproxy02 ~]#
```

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 PaceMaker.

```
[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 dhcpcv6-client high-availability ssh
[root@HAProxy ~]#
systemctl.service.
[root@HAProxy02 ~]# firewall-cmd --add-service=high-availability --permanent
success
[root@HAProxy02 ~]# firewall-cmd --reload
success
[root@HAProxy02 ~]# firewall-cmd --list-services
cockpit dhcpcv6-client high-availability ssh
[root@HAProxy02 ~]#
```

```
root@HAProxy:~
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 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 ~]#  
HAProxy: Starting Cluster...  
[root@HAProxy ~]# pcs cluster enable --all  
HAProxy: Cluster Enabled  
HAProxy02: Cluster Enabled  
[root@HAProxy ~]#
```

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 ~]#
```



```
[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 been defined
(unpack_resources)      error: Either configure some or disable STONITH with the stonith-enabled option
(unpack_resources)      error: NOTE: Clusters with shared data need STONITH to ensure data integrity
crm_verify: Errors found during check: config not valid
```

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 cidr_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 qlen 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:fe7:8858/64 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/haproxy
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 :

```
[root@HAProxy ~]# cd /usr/lib/ocf/resource.d/heartbeat
[root@HAProxy heartbeat]# sudo curl -O https://raw.githubusercontent.com/thisismitch/cluster-agents/master/haproxy
  % Total    % Received % Xferd  Average Speed   Time     Time     Current
               Dload  Upload Total Spent   Left Speed
100  5384  100  5384    0      0 16772      0 --:--:-- --:--:-- --:--:-- 16772
[root@HAProxy heartbeat]# chmod +x haproxy
[root@HAProxy heartbeat]# [root@HAProxy02:/usr/lib/ocf/resource.d/heartbeat
Error: missing value of 'pcs' option
[root@HAProxy02 ~]# crm_verify -L -V
[root@HAProxy02 ~]# crm_verify -L -V
[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
[root@HAProxy02 ~]# pcs resource create virtual_ip ocf:heartbeat:IPAddr2 ip=192.168.111.163 cidr_netmask=24 op monitor interval=30s
[root@HAProxy02 ~]# pcs status resources
  * virtual_ip (ocf:heartbeat:IPAddr2):           Started HAProxy
[root@HAProxy02 ~]# cd /usr/lib/ocf/resource.d/heartbeat
[root@HAProxy02 heartbeat]# sudo curl -O https://raw.githubusercontent.com/thisismitch/cluster-agents/master/haproxy
  % Total    % Received % Xferd  Average Speed   Time     Time     Current
               Dload  Upload Total Spent   Left Speed
100  5384  100  5384    0      0 36378      0 --:--:-- --:--:-- --:--:-- 36378
[root@HAProxy02 heartbeat]# chmod +x haproxy
```

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



```
root@HAPROXY:/usr/lib/ocf/resource.d/heartbeat
apache
conntrackd
corosync-qnetd
crypt
CTDB
db2
Delay
dhcpd
Dummy
ethmonitor
exportfs
Filesystem
galera
garbd
haproxy
iface-vlan
IPaddr2
IPsrcaddr
iSCSILogicalUnit
iSCSITarget
LVM-activate
lvmlockd
MailTo

root@HAProxy02:/usr/lib/ocf/resource.d/heartbeat
apache
conntrackd
corosync-qnetd
crypt
CTDB
db2
Delay
dhcpd
Dummy
ethmonitor
exportfs
Filesystem
galera
garbd
haproxy
iface_vlan
IPaddr2
IPsrcaddr
iSCSILogicalUnit
iSCSITarget
LVM-activate
lvmlockd
MailTo
```

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/haproxy conffile=/etc/haproxy/haproxy.cfg op monitor interval=10s
```

Nous affectons notre ressource à notre cluster :

```
/haprox config=/etc/haproxy/haproxy.cfg op monitor interval=10s
[root@HAPROXY heartbeat]# pcs resource group add HAproxyGroup virtual_ip haproxy
[root@HAPROXY heartbeat]#
```



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
[root@HAProxy cluster]#
```



```
root@HAPROXY:/var/log/cluster
GNU nano 5.6.1                               corosync.log
Dec 14 09:08:01 [13693] HAProxy corosync notice  [MAIN   ] Corosync Cluster Engine 3.1.7 star>
Dec 14 09:08:01 [13693] HAProxy corosync info    [MAIN   ] Corosync built-in features: dbus s>
Dec 14 09:08:01 [13693] HAProxy corosync notice  [TOTEM  ] Initializing transport (Kronosnet).>
Dec 14 09:08:03 [13693] HAProxy corosync info    [TOTEM  ] totemknet initialized
Dec 14 09:08:03 [13693] HAProxy corosync info    [KNET   ] pmtud: MTU manually set to: 0
Dec 14 09:08:03 [13693] HAProxy corosync info    [KNET   ] common: crypto_nss.so has been loa>
Dec 14 09:08:03 [13693] HAProxy corosync notice  [SERV   ] Service engine loaded: corosync co>
Dec 14 09:08:03 [13693] HAProxy corosync info    [QB     ] server name: cmap
Dec 14 09:08:03 [13693] HAProxy corosync notice  [SERV   ] Service engine loaded: corosync co>
Dec 14 09:08:03 [13693] HAProxy corosync info    [QB     ] server name: cfg
Dec 14 09:08:03 [13693] HAProxy corosync notice  [SERV   ] Service engine loaded: corosync cl>
Dec 14 09:08:03 [13693] HAProxy corosync info    [QB     ] server name: cpg
Dec 14 09:08:03 [13693] HAProxy corosync notice  [SERV   ] Service engine loaded: corosync pr>
Dec 14 09:08:03 [13693] HAProxy corosync notice  [QUORUM] Using quorum provider corosync_vot>
Dec 14 09:08:03 [13693] HAProxy corosync notice  [VOTEQ  ] Waiting for all cluster members. C>
Dec 14 09:08:03 [13693] HAProxy corosync notice  [SERV   ] Service engine loaded: corosync vo>
Dec 14 09:08:03 [13693] HAProxy corosync info    [QB     ] server name: votequorum
Dec 14 09:08:03 [13693] HAProxy corosync notice  [SERV   ] Service engine loaded: corosync cl>
Dec 14 09:08:03 [13693] HAProxy corosync info    [QB     ] server name: quorum
[ Read 47 lines ]
^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File   ^V Replace    ^U Paste      ^J Justify   ^L Go To Line
```

Sur nos deux serveurs haproxy on start haproxy :



```
[root@HAPROXY:~]
● haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/usr/lib/systemd/system/haproxy.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
lines 1-3/3 (END)
^C
[root@HAPROXY ~]# systemctl start haproxy
[root@HAPROXY ~]#
Message from syslogd@localhost at Dec 14 10:21:35 ...
haproxy[18375]: backend static has no server available!

[root@HAPROXY ~]# systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/usr/lib/systemd/system/haproxy.service; disabled; vendor preset: disabled)
  Active: active (running) since Thu 2023-12-14 10:21:35 CET; 1min 18s ago
    Process: 18371 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -f $CFGDIR -c -q $OPTIONS (code=exited, status=0/OK)
   Main PID: 18373 (haproxy)
      Tasks: 2 (limit: 10892)
     Memory: 12.3M
        CPU: 250ms
      CGroup: /system.slice/haproxy.service
          ├─18373 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d>
          └─18375 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d>

[root@HAPROXY02:~]
rich rules:
[root@HAPROXY02 ~]# systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/usr/lib/systemd/system/haproxy.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
lines 1-3/3 (END)
^C
[root@HAPROXY02 ~]# systemctl start haproxy
[root@HAPROXY02 ~]# systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/usr/lib/systemd/system/haproxy.service; disabled; vendor preset: disabled)
  Active: active (running) since Thu 2023-12-14 10:24:07 CET; 5s ago
    Process: 5225 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -f $CFGDIR -c -q $OPTIONS (code=exited, status=0/OK)
   Main PID: 5227 (haproxy)
      Tasks: 3 (limit: 10885)
     Memory: 12.7M
        CPU: 56ms
      CGroup: /system.slice/haproxy.service
          ├─5227 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d>
          └─5229 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d>

Dec 14 10:24:07 HAPROXY02 haproxy[5227]: [NOTICE]      (5227) : New worker #1 (5229) forked
Dec 14 10:24:07 HAPROXY02 haproxy[5229]: [WARNING]    (5229) : Server static/static is DOWN, r>
```

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



Inscrис Toi Beau Gosse

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

**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
   * quiesce        (ocf:heartbeat:quiesce):          Stopped

[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 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
  link/ether 00:0c:29: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:fed:6936/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
[root@HAProxy02 ~]#
```