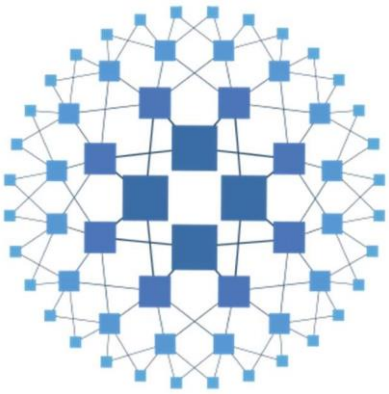
TP Mise en place d'un service Web



HAPROXY

Table des matières

#1 5
A. Installation de Rocky linux 9 5
B. Install service web 5
1. Installation de Nginx 5
2. Install Php-fpm 6
3. Install mariadb 7
C. Installation du moteur de base de données 8
D. Déploiement du site .fr 8
1. Création du repertoire pour le site web 8
2. Modification du fichier php-fpm.d/www.conf 9
3. Modifier fichier conf php 9
4. Implémentation du script dans le fichier datbase.sql 10
5. Execution du script 10
6. Modification du fichier de conf du site 10
7. Modification du fichier host sur le pc 11
8. Téléversement sur la machine des ressources 11
E. Déploiement du site .uk 12
F. Restriction des accès IP 14
1. Ajout dans le fichier de conf le fichier blockip.conf 14
G. Analyse des logs 16
1. Install de goaccess 16

2 18
A. Captation des trames avec Wireshark 18
B. Sécurisation d'un service WEB (nginx) 18
1. Création du fichier 'private' puis restriction de l'accès 18
2. Génération du certificat 19
3. Génération un groupe DiffieHellman fort 19
4. Modification des fichiers de conf des sites .fr et .uk 19
5. Test la configuration 20
6. Second test WireShark 21
7. Suppression de l'ancien protocole 21

3 22
A. Installation de 2 nouveaux serveurs rocky linux 9 22
1. Serveur web02 22
2. Install de HA01 24
B. Configuration de la base de données 25
1. Ouverture du port 3306 sur le serveur web01 25
2. On décommente la ligne 25
3. Création des users 25
4. Liaison a distance tester depuis le 2eme serveur web 26
C. Mise en place d'un serveur HA 26
1. Copie le fichier de configuration de haproxy 26
2. Modification du fichier de conf 26
3. Activation de Rsyslog 27
4. Autorisation dans SELinux 27
5. Activation de Haproxy et Rsyslog 27
6. Accès à l'interface 27
7. Configuration de l'https 28
8. Modification fichier conf haproxy 28
9. On constate que les 2 serveurs remontent dans le serveur haproxy 29
D. Exercices 30
1. On coupe un serveur web 30
2. On peut mettre différents modes 30

#4 31
A. Installation d'un 2ème Haproxy 31
1. Snapshot de la vm 31
2. Clone de la vm 31
3. Changement de carte réseau pour avoir une nouvelle IP 31
4. Renommer la machine 31
5. Ajout dans le fichier host pour chaque machine 31
B. Redondance Haproxy 31
1. Ouverture du port 80 31
2. Installation du pacemaker sur les 2 haproxy 32
3. Mise en place du cluster 32
4. Création d'un user 32
5. Authentification entre nos 2 nodes 33
6. Déploiement du cluster 33
7. Vérification des erreurs 34
8. Correction des erreurs 34
9. Re vérification 34
10. Création d'une adresse IP virtuel 34
11. Installation de l'agent sur les 2 serveurs 35

#1

A. Installation de Rocky linux 9

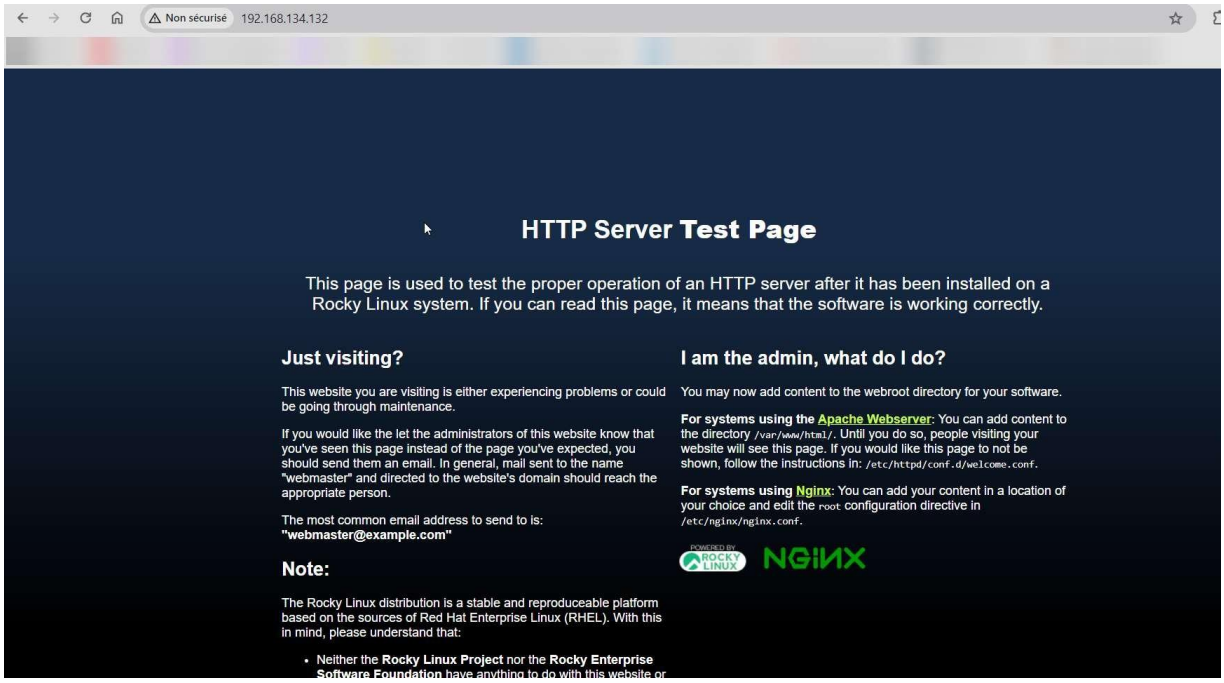
Mdp root : root

B. Install service web

1. Installation de Nginx

```
C:\WINDOWS\system32>ssh matthieu@192.168.134.132
matthieu@192.168.134.132's password:
Last login: Mon Feb 12 08:35:58 2024
[matthieu@localhost ~]$
[matthieu@localhost ~]$
[matthieu@localhost ~]$
[matthieu@localhost ~]$
[matthieu@localhost ~]$ sudo dnf install nginx
[sudo] password for matthieu:
$Last metadata expiration check: 0:09:28 ago on Mon Feb 12 08:29:12 2024.
Dependencies resolved.
================================================================================ Package                    Architecture
        Version                         Repository              Size
================================================================================Installing:
 nginx                          x86_64              1:1.20.1-14.el9_2.1               appstream               36 k
Installing dependencies:
 nginx-core                     x86_64              1:1.20.1-14.el9_2.1               appstream              565 k
 nginx-filesystem               noarch              1:1.20.1-14.el9_2.1               appstream              8.5 k
 rocky-logos-httpd              noarch              90.14-2.el9                       appstream               24 k

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

Total download size: 634 k
Installed size: 1.8 M
Is this ok [y/N]: y
Is this ok [y/N]: y
Downloading Packages:
(1/4): rocky-logos-httpd-90.14-2.el9.noarch.rpm                 256 kB/s |  24 kB     00:00
(2/4): nginx-filesystem-1.20.1-14.el9_2.1.noarch.rpm             83 kB/s | 8.5 kB     00:00
(3/4): nginx-1.20.1-14.el9_2.1.x86_64.rpm                       281 kB/s |  36 kB     00:00
(4/4): nginx-core-1.20.1-14.el9_2.1.x86_64.rpm                  2.9 MB/s | 565 kB     00:00
```

```
[matthieu@localhost ~]$ sudo systemctl start nginx
[matthieu@localhost ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Mon 2024-02-12 08:39:57 CET; 17s ago
    Process: 11655 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 11656 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 11657 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 11658 (nginx)
      Tasks: 3 (limit: 10940)
     Memory: 2.9M
        CPU: 28ms
     CGroup: /system.slice/nginx.service
             ├─11658 "nginx: master process /usr/sbin/nginx"
             ├─11659 "nginx: worker process"
             └─11660 "nginx: worker process"

Feb 12 08:39:57 localhost.localdomain systemd[1]: Starting The nginx HTTP and reverse proxy server...
Feb 12 08:39:57 localhost.localdomain nginx[11656]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Feb 12 08:39:57 localhost.localdomain nginx[11656]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Feb 12 08:39:57 localhost.localdomain systemd[1]: Started The nginx HTTP and reverse proxy server.
[matthieu@localhost ~]$ sudo firewall-cmd --permanent --add-service=http
success
[matthieu@localhost ~]$ sudo firewall-cmd --permanent --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client http ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[matthieu@localhost ~]$
```



2. Install Php-fpm :

https://docs.rockylinux.org/guides/web/php/

Sudo dnf install epel-release
Sudo dnf install http://rpms.remirepo.net/enterprise/remi-release-S.rpm
Sudo dnf install php
php -v

```
Complete!
[matthieu@localhost ~]$ php -v
PHP 8.0.30 (cli) (built: Aug  3 2023 17:13:08) ( NTS gcc x86_64 )
Copyright (c) The PHP Group
Zend Engine v4.0.30, Copyright (c) Zend Technologies
    with Zend OPcache v8.0.30, Copyright (c), by Zend Technologies
[matthieu@localhost ~]$
```

sudo dnf install php-fpm
sudo systemctl enable php-fpm
sudo systemctl start php-fpm
sudo systemctl status php-fpm

```
[matthieu@localhost ~]$ sudo dnf install php-fpm
Last metadata expiration check: 0:02:45 ago on Mon Feb 12 08:48:25 2024.
Package php-fpm-8.0.30-1.el9_2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[matthieu@localhost ~]$ sudo systemctl enable php-fpm
do systemctl start php-fpm
sudo systemctl status php-fpmCreated symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service → /usr/lib/systemd/system/php-fpm.service.
[matthieu@localhost ~]$ sudo systemctl start php-fpm
[matthieu@localhost ~]$ sudo systemctl status php-fpm
● php-fpm.service - The PHP FastCGI Process Manager
     Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; enabled; preset: disabled)
     Active: active (running) since Mon 2024-02-12 08:51:19 CET; 1s ago
   Main PID: 2198 (php-fpm)
     Status: "Ready to handle connections"
      Tasks: 6 (limit: 10940)
     Memory: 11.7M
        CPU: 43ms
     CGroup: /system.slice/php-fpm.service
             ├─2198 "php-fpm: master process (/etc/php-fpm.conf)"
             ├─2199 "php-fpm: pool www"
             ├─2200 "php-fpm: pool www"
             ├─2201 "php-fpm: pool www"
             ├─2202 "php-fpm: pool www"
             └─2203 "php-fpm: pool www"

Feb 12 08:51:19 localhost.localdomain systemd[1]: Starting The PHP FastCGI Process Manager...
Feb 12 08:51:19 localhost.localdomain systemd[1]: Started The PHP FastCGI Process Manager.
```

3. Install mariadb :

https://www.digitalocean.com/community/tutorials/how-to-install-mariadb-on-rocky-linux-9

sudo dnf install mariadb-server
sudo systemctl enable mariadb
sudo systemctl start mariadb
sudo systemctl status mariadb

sudo mysql_secure_installation

mdp root : root

mysqladmin -u root -p version

```
[matthieu@localhost ~]$ mysqladmin -u root -p version
Enter password:
mysqladmin  Ver 9.1 Distrib 10.5.22-MariaDB, for Linux on x86_64
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Server version          10.5.22-MariaDB
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/lib/mysql/mysql.sock
Uptime:                 2 min 2 sec

Threads: 1  Questions: 17  Slow queries: 0  Opens: 20  Open tables: 13  Queries per second avg: 0.139
[matthieu@localhost ~]$
```

C. Installation du moteur de base de données

Create database srvweb;
MariaDB> grant all privileges on srvweb.* TO 'matthieu'@'localhost' identified by 'PASSWORD';
Flush privileges;

Mdp pour utilisateur matthieu : PASSWORD

D. Déploiement du site .fr

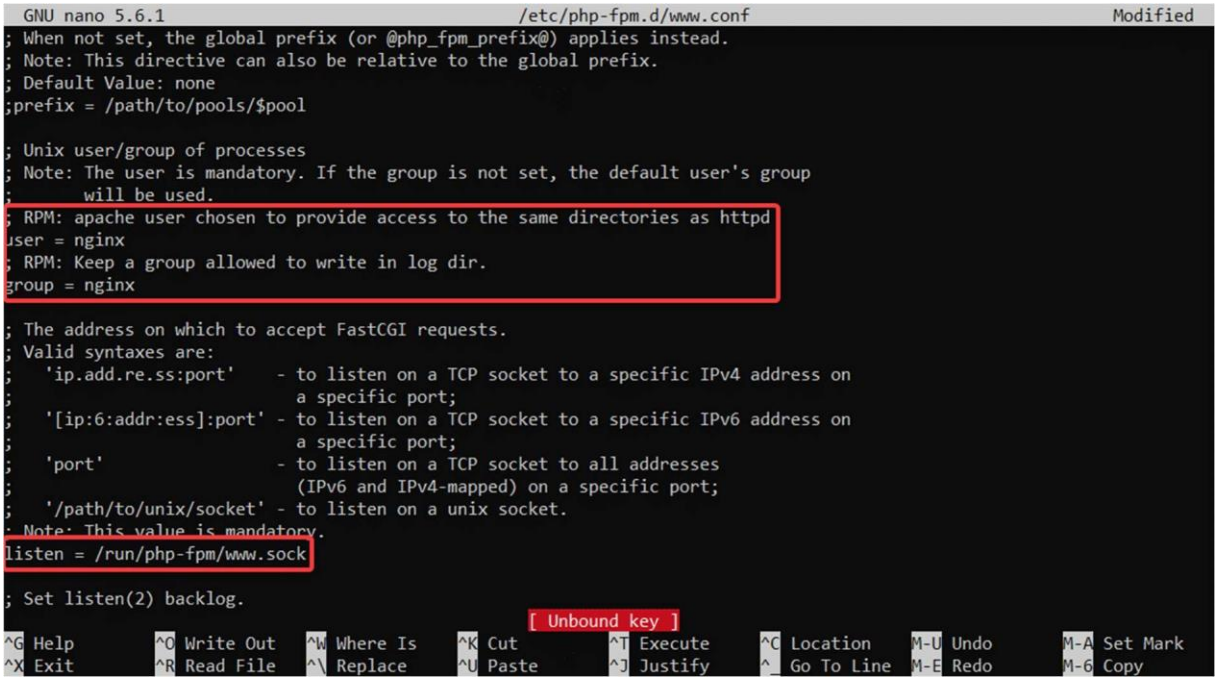
1. Création du repertoire pour le site web :

mkdir -p /var/www/ecommerce.fr/

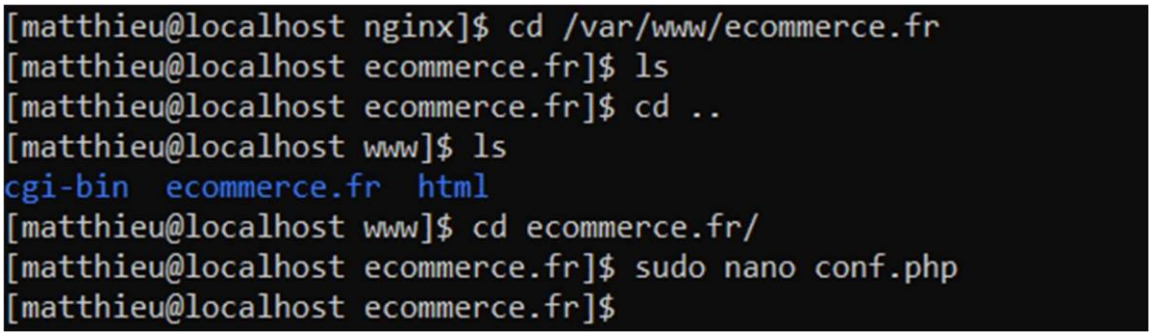
```
[matthieu@localhost nginx]$ sudo nano /etc/nginx/conf.d/ecommerce.fr
[matthieu@localhost nginx]$ [matthieu@localhost nginx]$ _
```

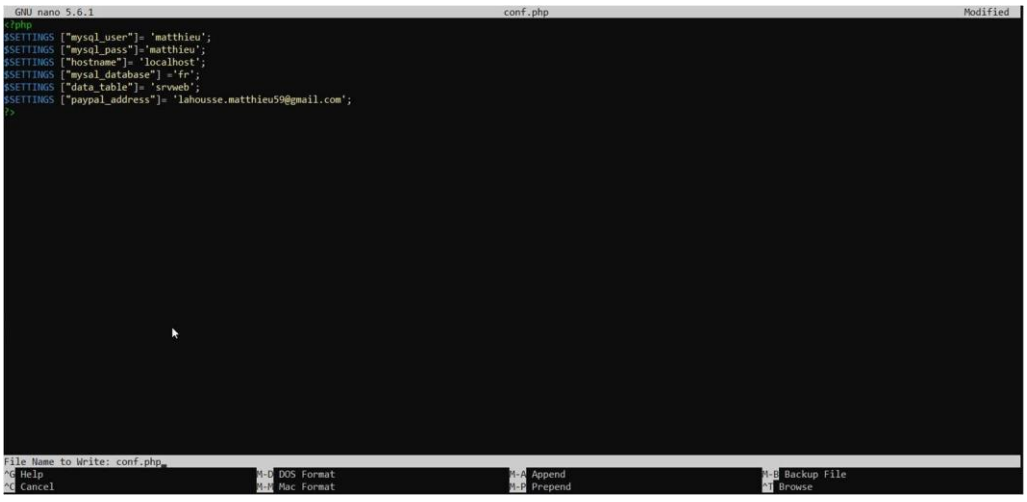
```
 GNU nano 5.6.1                                  /etc/nginx/conf.d/ecommerce.fr
server {
        listen 80;
        listen [::]:80;

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

        server_name ecommerce.fr ;
        location ~* \.php$ {
                fastcgi_pass unix:/run/php-fpm/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;
        }
}
                                        [ Read 21 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark    M-] To Bracket  M-Q Previous    ^B Back
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo      M-6 Copy        ^Q Where Was    M-W Next        ^F Forward
```

2. Modification du fichier php-fpm.d/www.conf

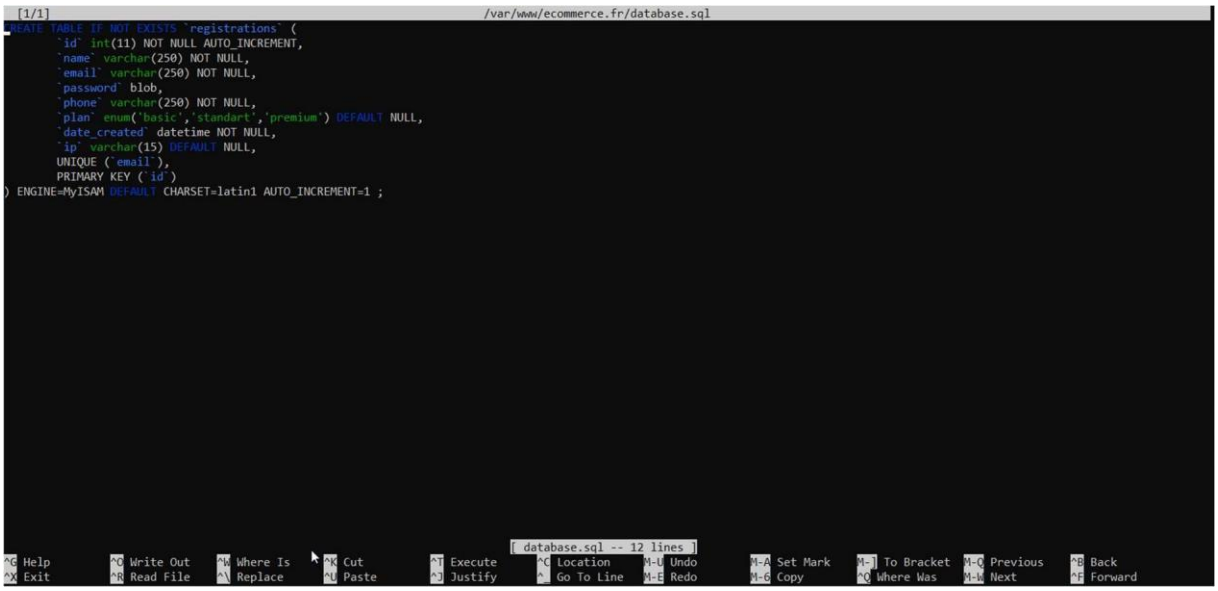


3. Modification fichier conf php





4. Implémentation du script dans le fichier datbase.sql



5. Execution du script

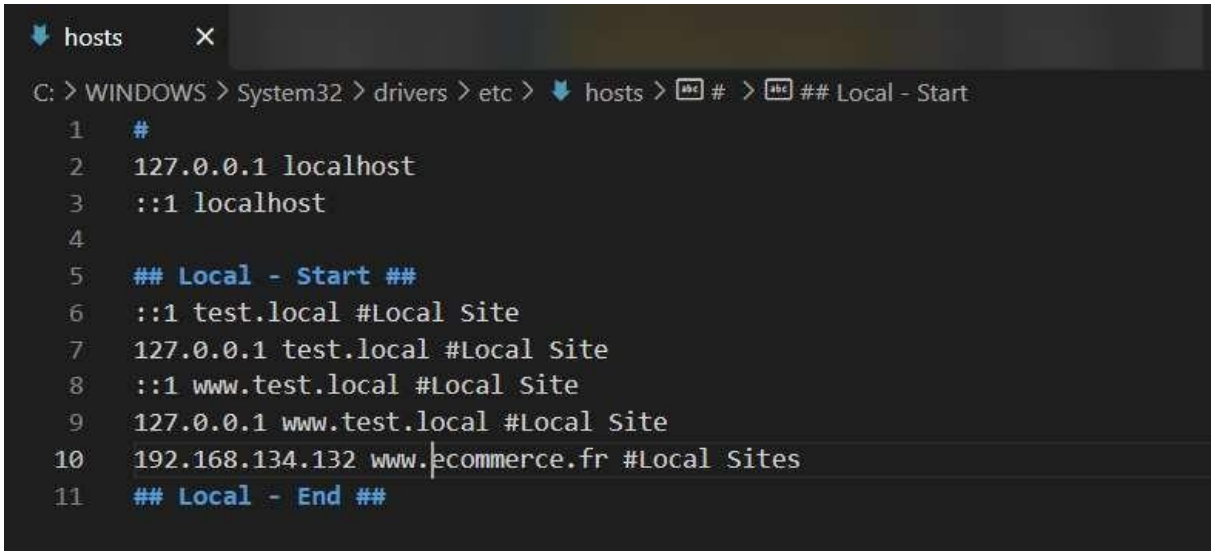
```
MariaDB [(none)]> use srvweb
Database changed
MariaDB [srvweb]> source /var/www/ecommerce.fr/database.sql
Query OK, 0 rows affected (0.051 sec)

MariaDB [srvweb]>
```

6. Modification du fichier de conf du site

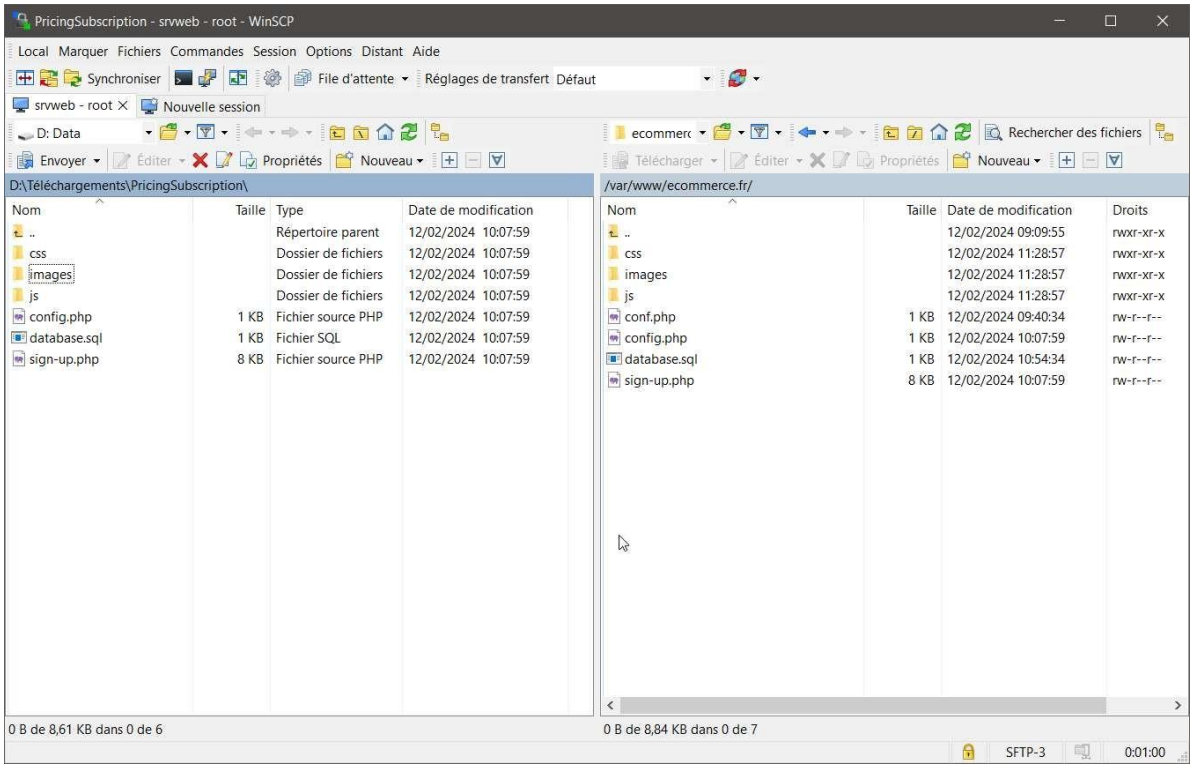
```
[matthieu@localhost ~]$ cd /etc/nginx/conf.d/
[matthieu@localhost conf.d]$ ls
ecommerce.fr  php-fpm.conf
[matthieu@localhost conf.d]$ cp ecommerce.fr ecommerce.fr.conf
```

```
[matthieu@localhost conf.d]$ sudo rm -rf ecommerce.fr
[matthieu@localhost conf.d]$ sudo systemctl restart nginx
[matthieu@localhost conf.d]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/nginx.service.d
             └─php-fpm.conf
     Active: active (running) since Wed 2024-02-28 17:03:28 CET; 18s ago
    Process: 1687 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 1688 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 1689 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 1690 (nginx)
      Tasks: 3 (limit: 10940)
     Memory: 3.2M
```

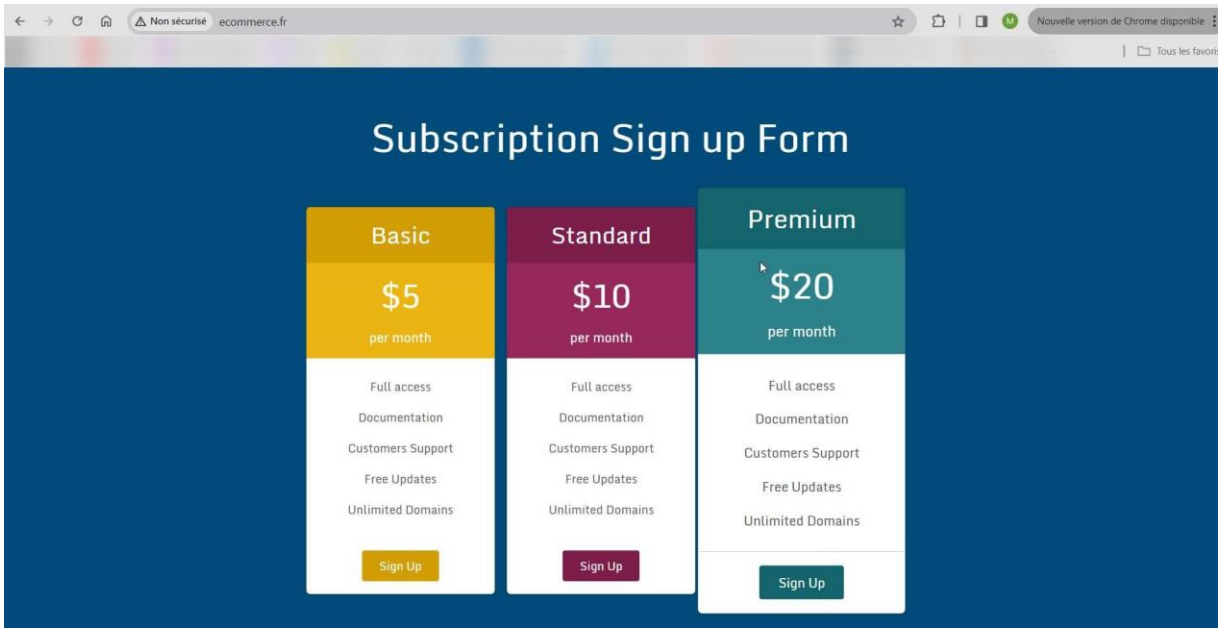
7. Modification du fichier host sur le pc :



8. Téléversement sur la machine des ressources :



Rendu du site :

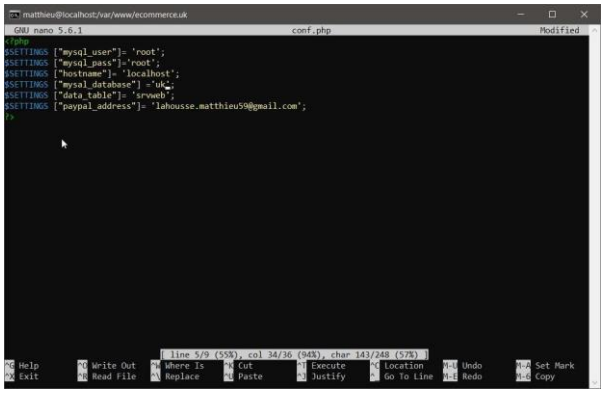


E. Déploiement du site .uk

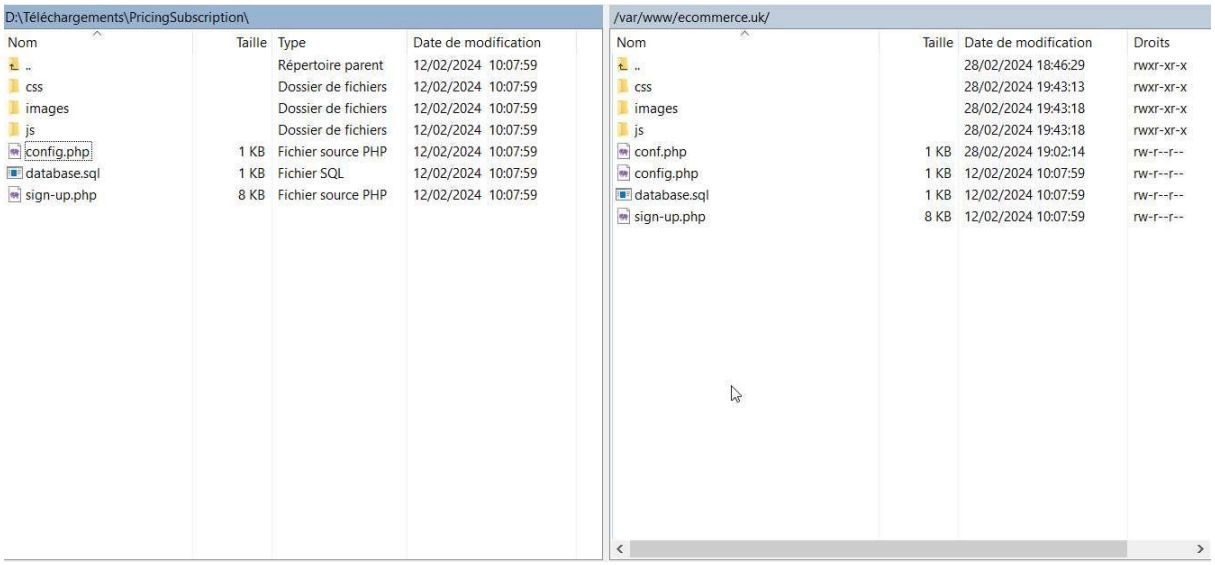
Même étape que pour le site .fr sans l'étape 2, pensez à bien changer les chemins en .uk et non .fr

```
[matthieu@localhost /]$ sudo mkdir -p /var/www/ecommerce.uk
[sudo] password for matthieu:
[matthieu@localhost /]$
```

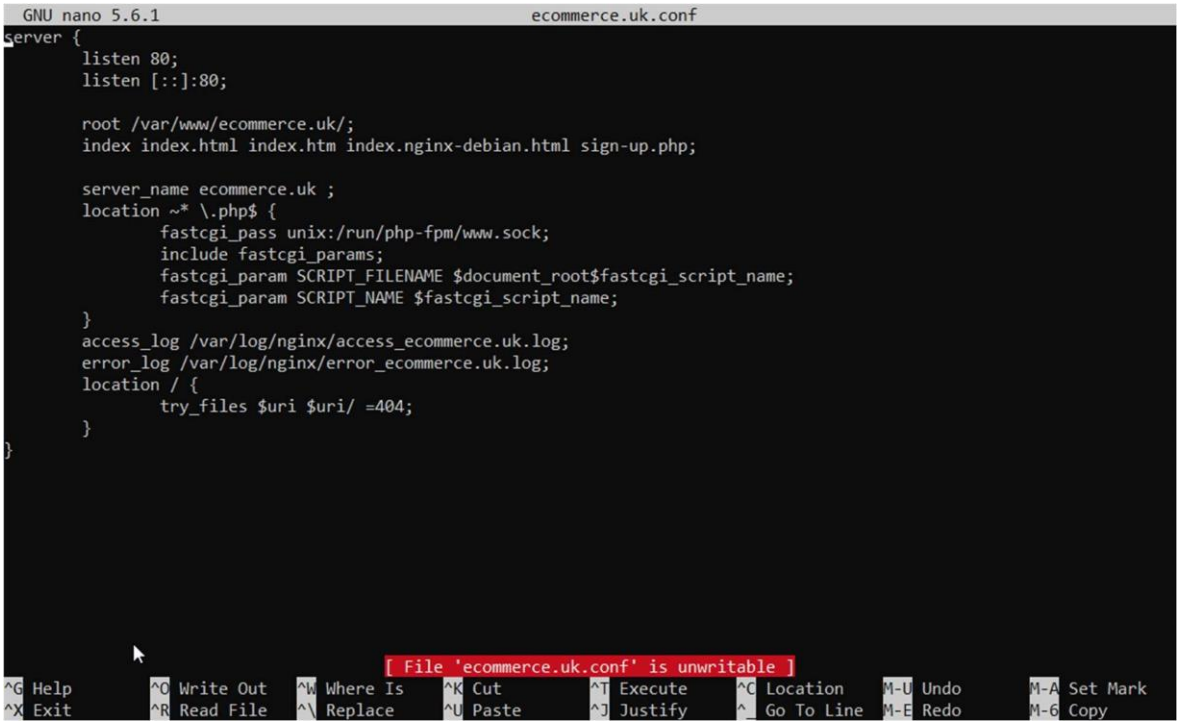
```
[matthieu@localhost ecommerce.fr]$ cd /var/www/ecommerce.uk/
[matthieu@localhost ecommerce.uk]$ sudo nano conf.php
[matthieu@localhost ecommerce.uk]$
```



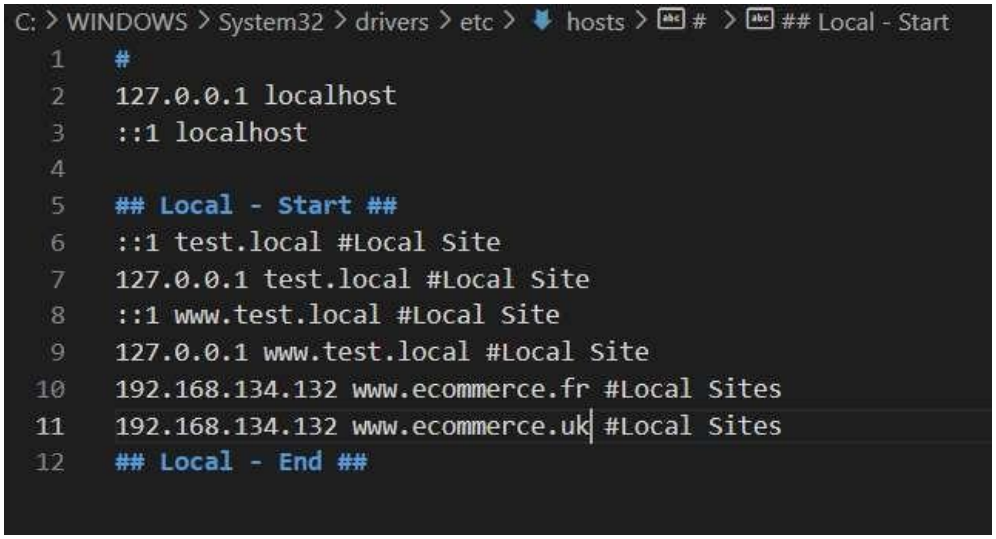
Transfert avec winscp des fichiers conf



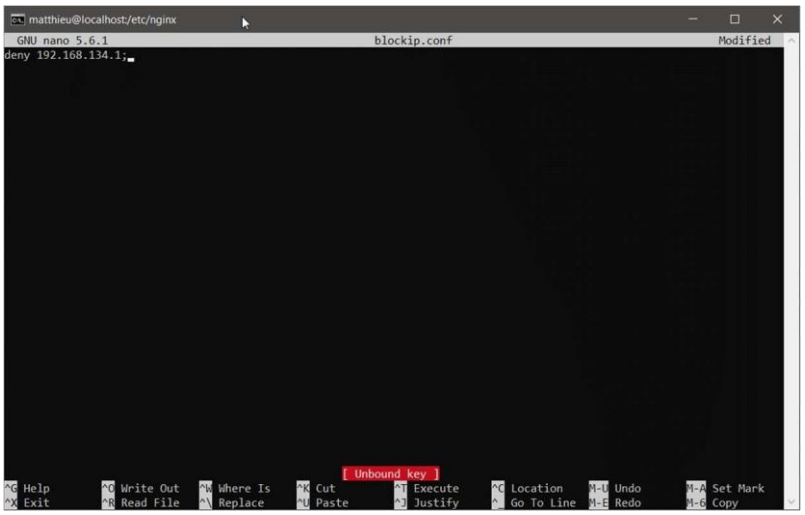
Modification du fichier conf



Ajout dans le fichier host

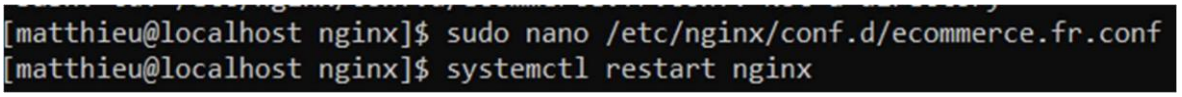


Création du fichier blockip.conf



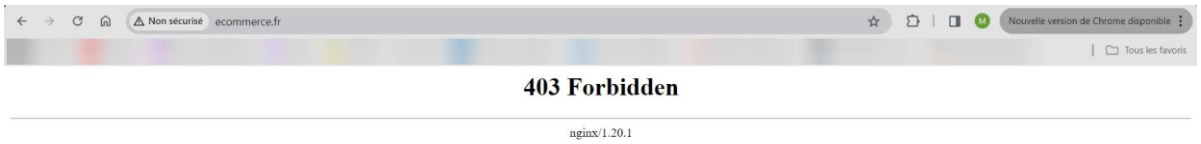
F. Restriction des accès IP

1. Ajout dans le fichier de conf le fichier blockip.conf



```
  GNU nano 5.6.1                          /etc/nginx/conf.d/ecommerce.fr.conf
server {
        listen 80;
        listen [::]:80;

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

        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;
        }
}


                                [ Read 22 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo     M-6 Copy
```

Adresse ip bloqué sur le site ecommerce.fr



403 Forbidden

nginx/1.20.1

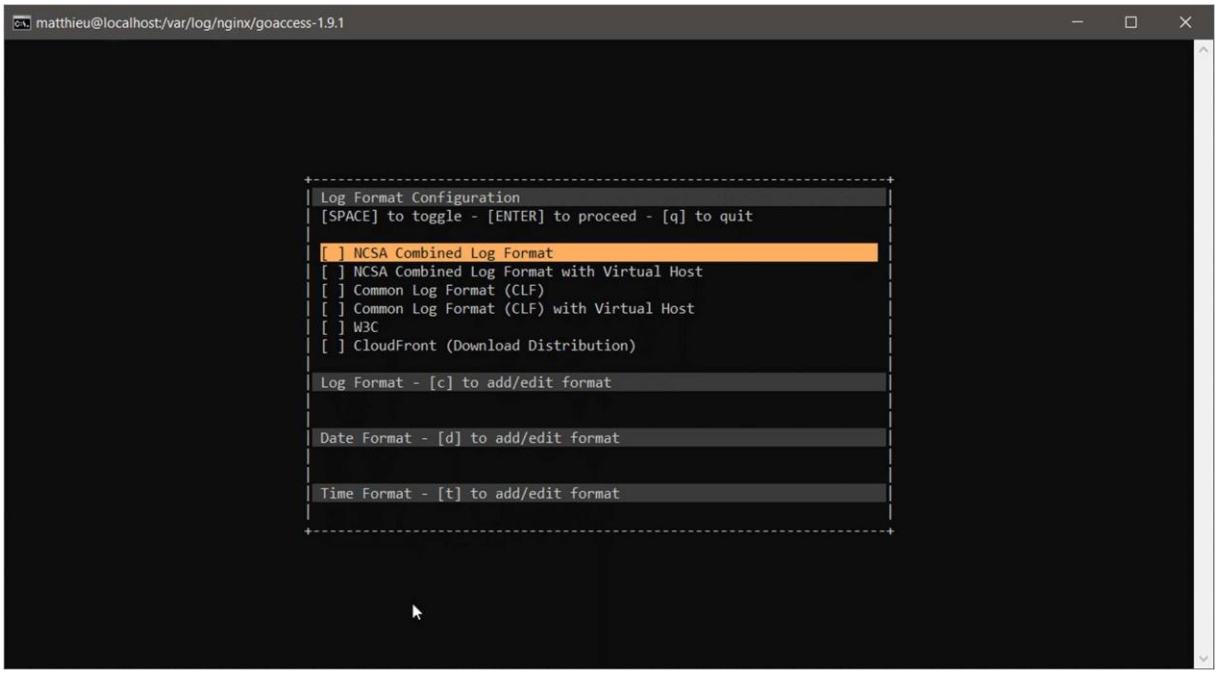
G. Analyse des logs

1. Installation de goaccess

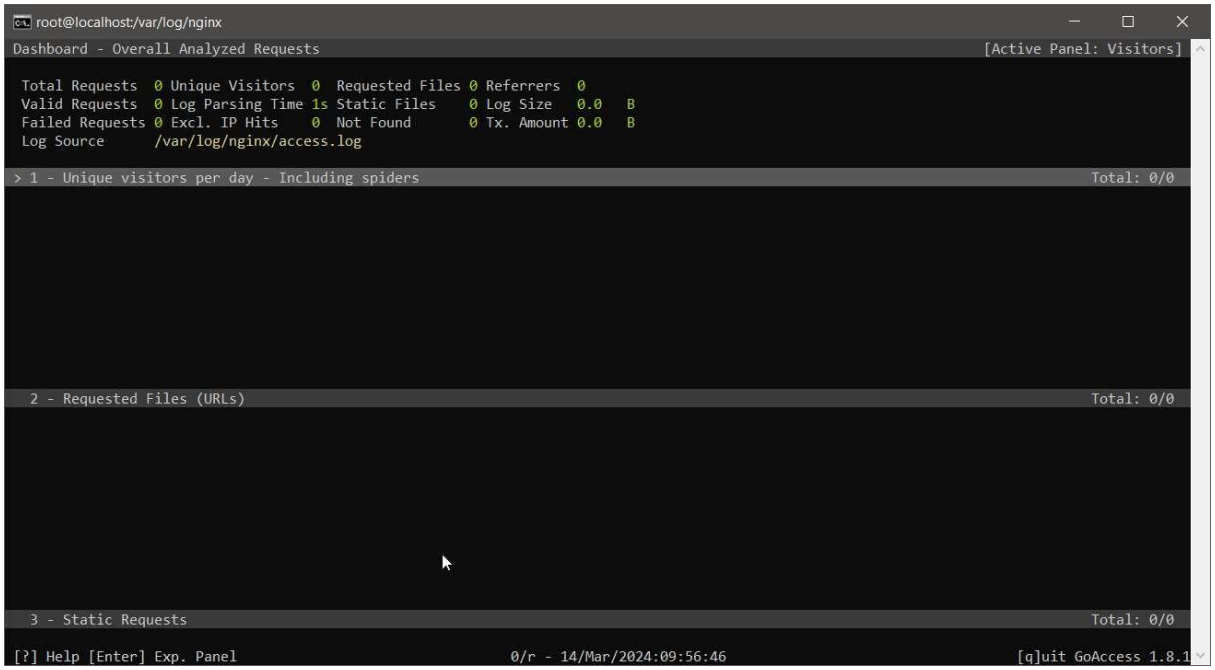
Goaccess est un outil qui permets de lire plus facilement les logs

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

```
goaccess -f /var/log/nginx/accesss.log
```



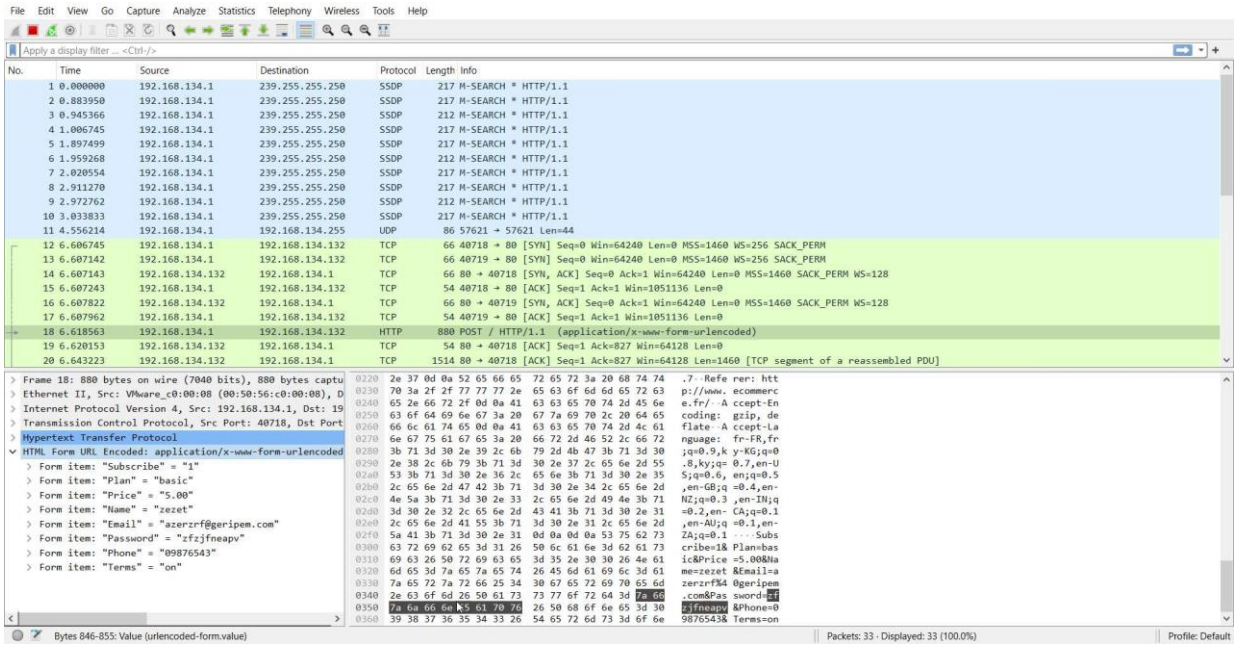
Sélectionnez le premier (barre espace), puis faite 'entrer'


root@localhost:/var/log/nginx
Dashboard - Overall Analyzed Requests
[Active Panel: Visitors]
Total Requests 0 Unique Visitors 0 Requested Files 0 Referrers 0
Valid Requests 0 Log Parsing Time 1s Static Files 0 Log Size 0.0 B
Failed Requests 0 Excl. IP Hits 0 Not Found 0 Tx. Amount 0.0 B
Log Source /var/log/nginx/access.log
> 1 - Unique visitors per day - Including spiders
Total: 0/0
2 - Requested Files (URLs)
Total: 0/0
3 - Static Requests
Total: 0/0
[?] Help [Enter] Exp. Panel
0/r - 14/Mar/2024:09:56:46
[q]uit GoAccess 1.8.1

2

A. Captation des trames avec Wireshark

On observe qu'avec l'outil Wireshark on peut lire dans les trames en clair toutes les infos de connexion



B. Sécurisation d'un service WEB (nginx)

1. Création du fichier 'private' puis restriction de l'accès

```
[matthieu@localhost ~]$ sudo mkdir /etc/ssl/private
[sudo] password for matthieu:
[matthieu@localhost ~]$ chmod 700 /etc/ssl/private
chmod: changing permissions of '/etc/ssl/private': Operation not permitted
[matthieu@localhost ~]$ sudo chmod 700 /etc/ssl/private
[matthieu@localhost ~]$
```

2. Génération du certificat

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

```
Country Name (2 letter code) [XX]:fr
State or Province Name (full name) []:France
Locality Name (eg, city) [Default City]:lille
Organization Name (eg, company) [Default Company Ltd]:Gaston Berger
Organizational Unit Name (eg, section) []:bts sio
Common Name (eg, your name or your server's hostname) []:srv web
Email Address []:matthieu.lahousse@gastonberger.fr
```

3. Génération un groupe DiffieHellman fort

Il permet la négociation de PFS (Perfact Forward Secrecy)

```
openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

```
[matthieu@localhost ~]$ sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
Generating DH parameters, 2048 bit long safe prime
.........................................+.+..........+.............................................+.........................................
...........................................................................................................................+.................
.............................................................................................................................................
..................++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*
++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*
[matthieu@localhost ~]$
```

4. Modification des fichiers de conf des sites .fr et .uk

Edition des fichiers de configuration (/etc/nginx/conf .d) de chaque site pour intégrer le protocole https.

```
matthieu@localhost:/etc/nginx/conf.d
GNU nano 5.6.1                    ecommerce.fr.conf                    Modified
        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/;
        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;
}
File Name to Write: ecommerce.fr.conf
^G Help          M-D DOS Format     M-A Append      M-B Backup File
^C Cancel        M-M Mac Format     M-P Prepend     ^T Browse
```

```
matthieu@localhost:/etc/nginx/conf.d
 GNU nano 5.6.1                    ecommerce.uk.conf                    Modified
}


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/;
        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;
}

}
Save modified buffer?
 Y Yes
 N No           ^C Cancel
```

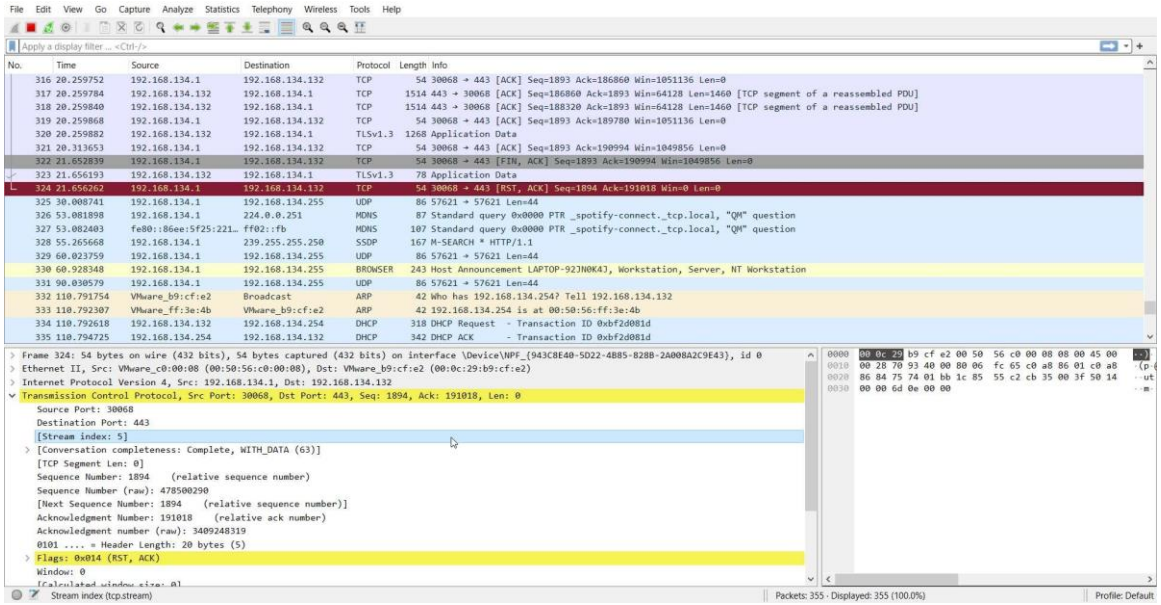
5. Test la configuration

```
[matthieu@localhost conf.d]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[matthieu@localhost conf.d]$
```

On obtient bien l'ensemble de nos infos dans le certificat

6. Second test WireShark

On re-observe les trames avec WireShark et on voit que s'est chiffré maintenant



7. Suppression de l'ancien protocole

Et on supprime maintenant le protocole http

```
[matthieu@localhost ~]$ sudo firewall-cmd --remove-service=http
[sudo] password for matthieu:
success
[matthieu@localhost ~]$ sudo firewall-cmd --list-service
cockpit dhcpv6-client https ssh
```

3

A. Installation de 2 nouveaux serveurs rocky linux 9

1. Serveur web02

a. Snapshot de la première machine
b. Clone de la machine pour la création de srv web 02

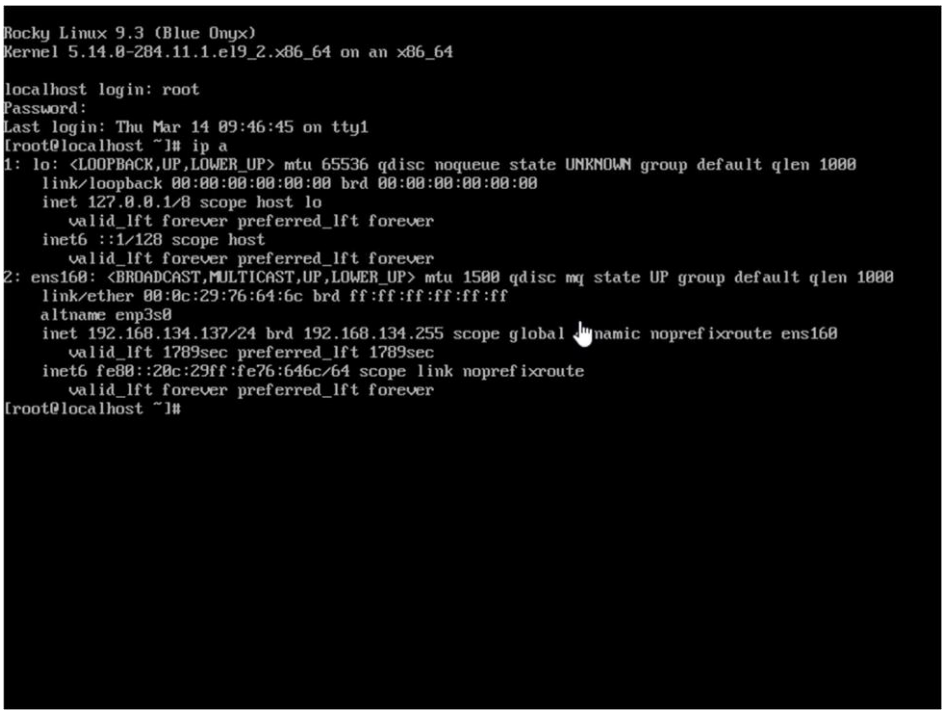
Avant le démarrage, enlever et remettre la carte réseau pour changer d'ip

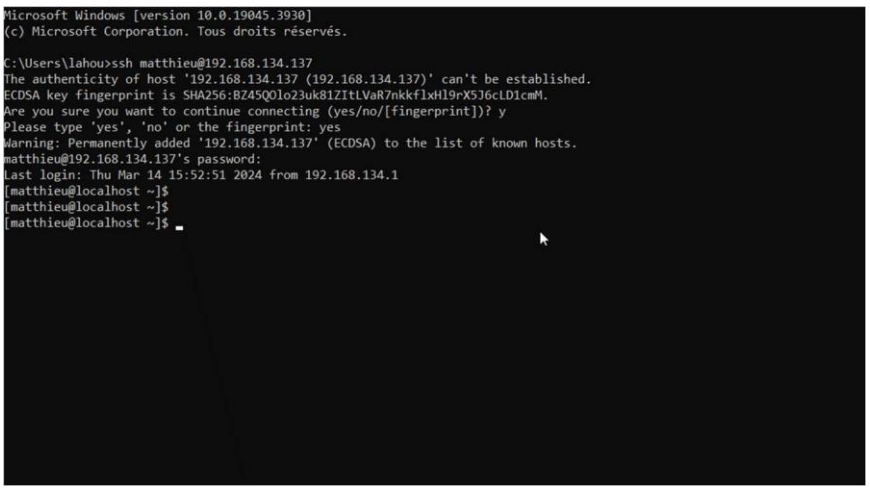
c. Se connecter

Obtenir la nouvelle ip

```
ip a
```

d. Connexion en ssh



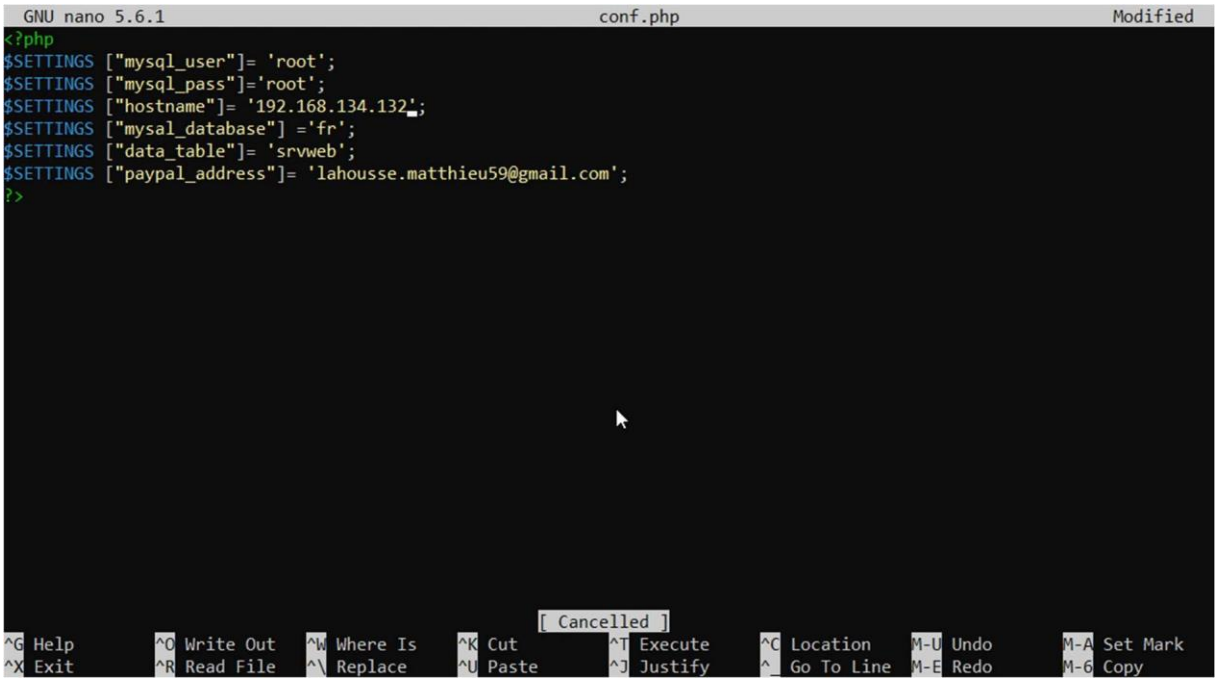


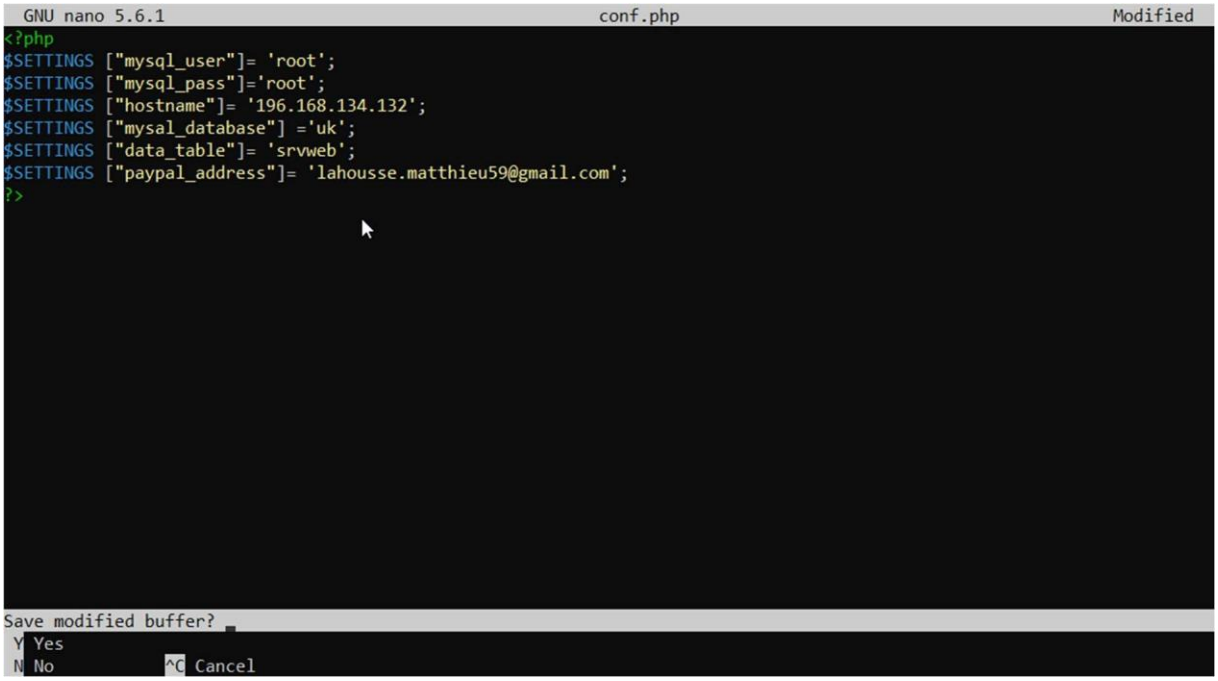
e. Suppression du service MariaDB (et ainsi BDD)

f. Lien vers le premier serveur web

On change le localhost pour mettre l'adresse de l'autre serveur

/var/www/ecommerce.fr/conf.php





2. Install de HA01

Installation classique d'une rocky 9

Activation du ssh pour un aspect pratique

B. Configuration de la base de données

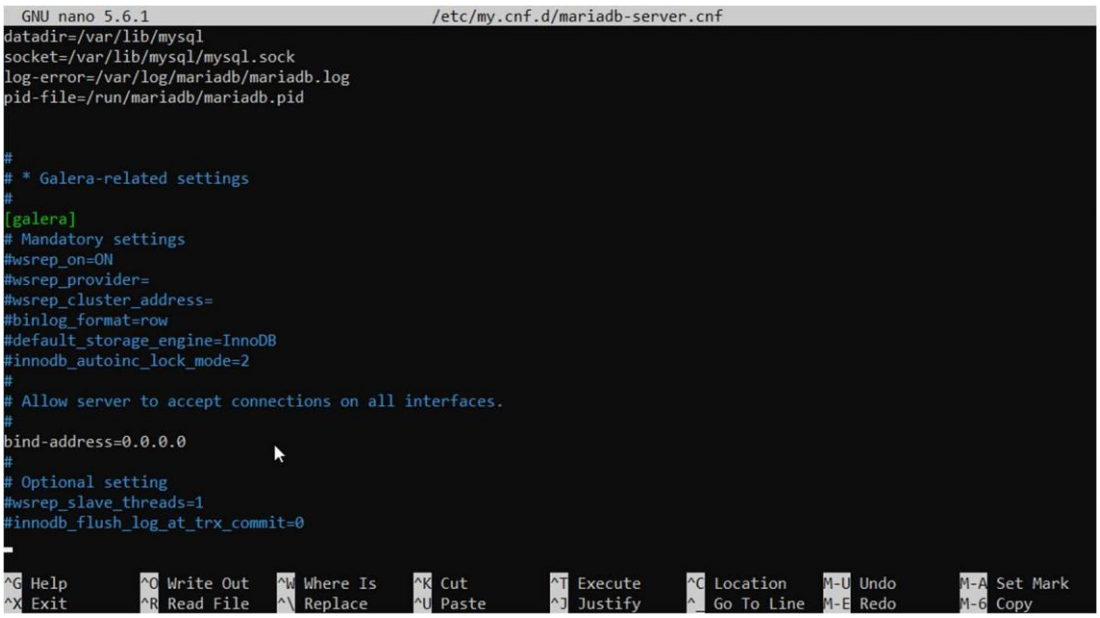
1. Ouverture du port 3306 sur le serveur web01

```
[matthieu@localhost ~]$ sudo firewall-cmd --permanent --add-port=3306/tcp
[sudo] password for matthieu:
success
[matthieu@localhost ~]$ sudo firewall-cmd --reload
success
```

2. On décommente la ligne

```
nano /etc/my.cnf.d/mariadb-server.cnf
```

```
bind-address=0.0.0.0
```



3. Création des users

```
MariaDB [(none)]> create user 'user_fr'@'192.168.134.137' identified by 'root';
Query OK, 0 rows affected (0.028 sec)

MariaDB [(none)]> create user 'user_uk'@'192.168.134.137' identified by 'root';
Query OK, 0 rows affected (0.001 sec)
```

```
MariaDB [(none)]> grant all on srvweb.* to 'user_fr'@'192.168.134.137';
Query OK, 0 rows affected (0.025 sec)

MariaDB [(none)]> grant all on srvweb_uk.* to 'user_uk'@'192.168.134.137';
Query OK, 0 rows affected (0.001 sec)
```

4. Liaison à distance testée depuis le 2eme serveur web :

```
[matthieu@localhost ecommerce.uk]$ mysql -u user_fr -h 192.168.134.132 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.5-10.5.22-MariaDB MariaDB Server

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server versi
on for the right syntax to use near 'database' at line 1
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| srvweb             |
+--------------------+
2 rows in set (0.00 sec)
```

C. Mise en place d'un serveur HA

1. Copie le fichier de configuration de haproxy

cd /etc/haproxy
cp haproxy.cfg copiehaproxy.cfg

2. Modification du fichier de conf

```
GNU nano 5.6.1                                  haproxy.cfg
#---------------------------------------------------------------------
# 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

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

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

#---------------------------------------------------------------------
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#---------------------------------------------------------------------
defaults
        mode          tcp

        listen stat

        bind *:8080

        mode http

        stats enable

        stats hide-version

        stats uri /stats
                                [ Read 90 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo      M-A Set Mark  M-] To Bracket  M-Q Previous
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo      M-6 Copy      ^Q Where Was    M-W Next
```

3. Activation de Rsyslog

Dans le répertoire

```
/etc/rsyslog.conf
```

Modification du fichier de conf de Rsyslog

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

4. Autorisation dans SELinux

```
setsebool -P haproxy_connect_any 1
```

5. Activation de Haproxy et Rsyslog

```
[matthieu@localhost haproxy]$ sudo systemctl enable haproxy
Created symlink /etc/systemd/system/multi-user.target.wants/haproxy.service → /usr/lib/systemd/system/haproxy.service.
[matthieu@localhost haproxy]$ sudo systemctl enable rsyslog
```

6. Accès à l'interface

Ouverture du port 8080

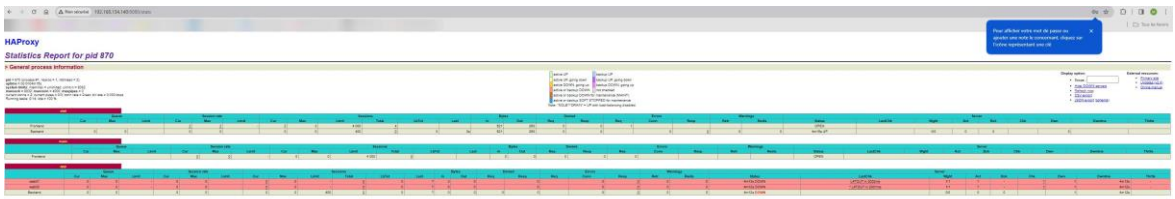
```
firewall-cmd --permanent --add-port=8080/tcp
```

Et on se connecte :

http://192.168.134.140:8080/stats

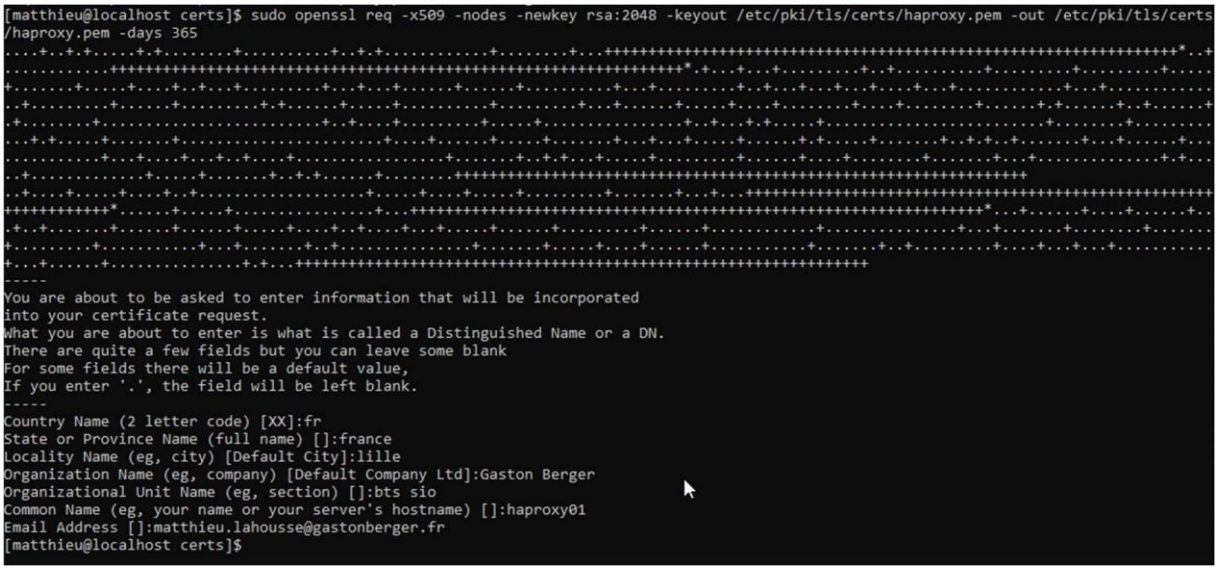
Identifiant : haproxy

Mot de passe : haproxy



7. Configuration de l'https

Cd /etc/pki/tls/certs
openssl req -x509 -nodes -newkey rsa:2048 -keyout /etc/pki/tls/certs/haproxy.pem -out /etc/pki/tls/certs/haproxy.pem -days 365

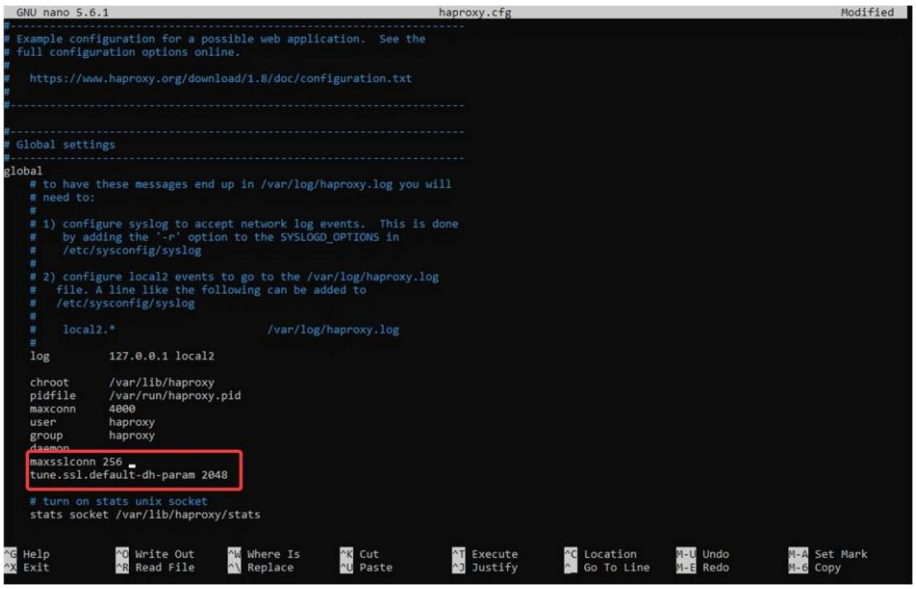


8. Modification fichier conf haproxy

cd /etc/haproxy

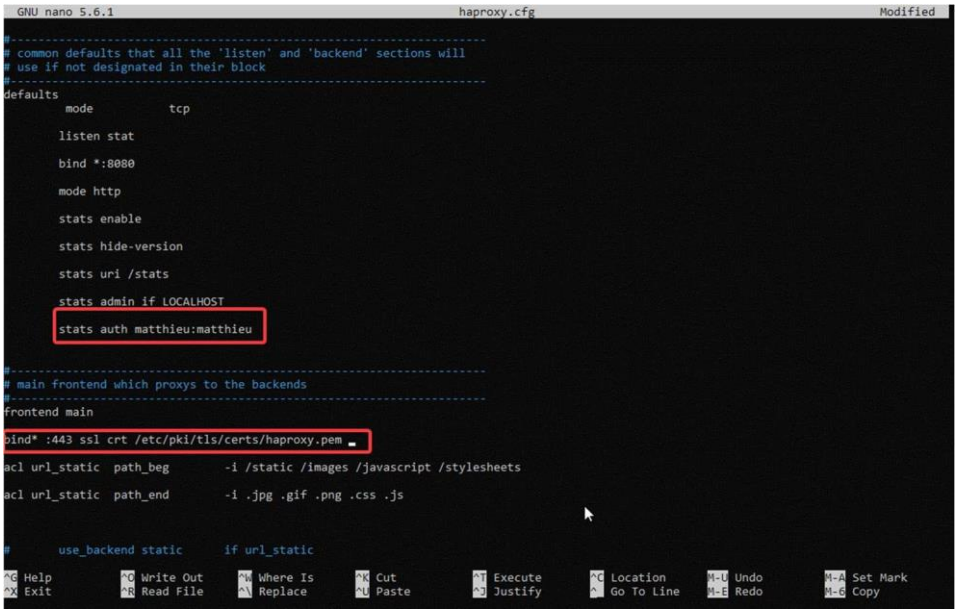
Et on rajoute les 2 lignes dans la section globale

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



Et dans la section front

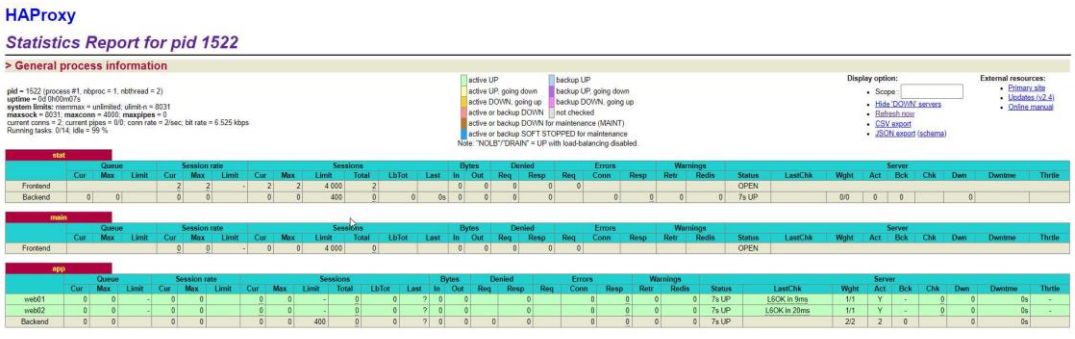
```
bind *:443 ssl crt /etc/pki/tls/certs/haprox y.pem
```



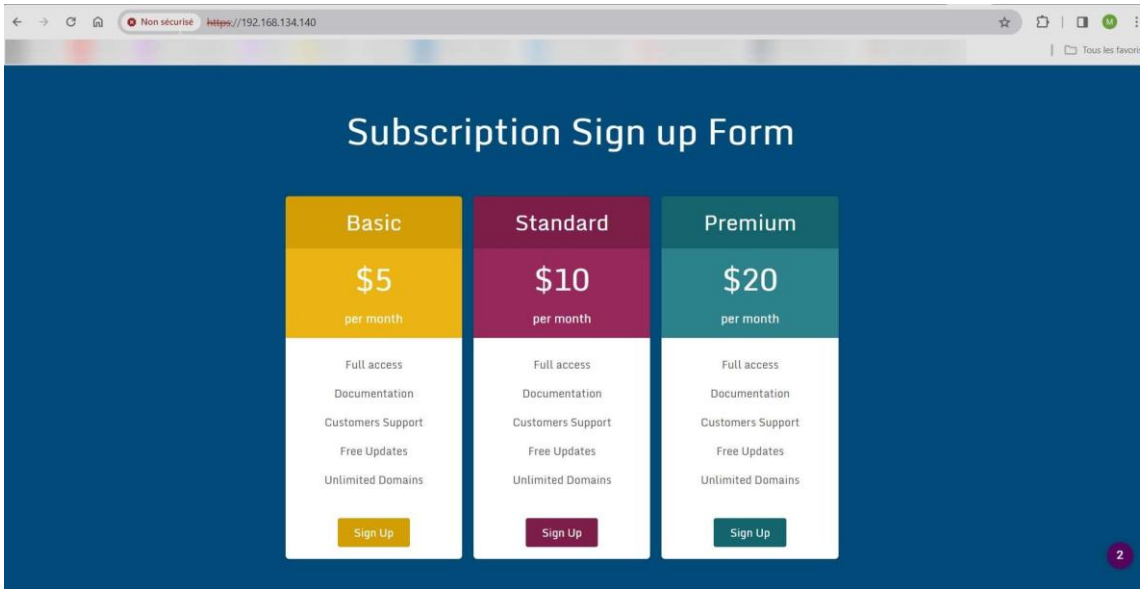
Et on redémarre le service

```
[matthieu@localhost ~]$ sudo firewall-cmd --permanent --add-port=443/tcp
success
[matthieu@localhost ~]$ sudo firewall-cmd --permanent --add-service=https
success
[matthieu@localhost ~]$ sudo firewall-cmd --reload
success
```

9. On constate que les 2 serveurs remontent dans le serveur haproxy

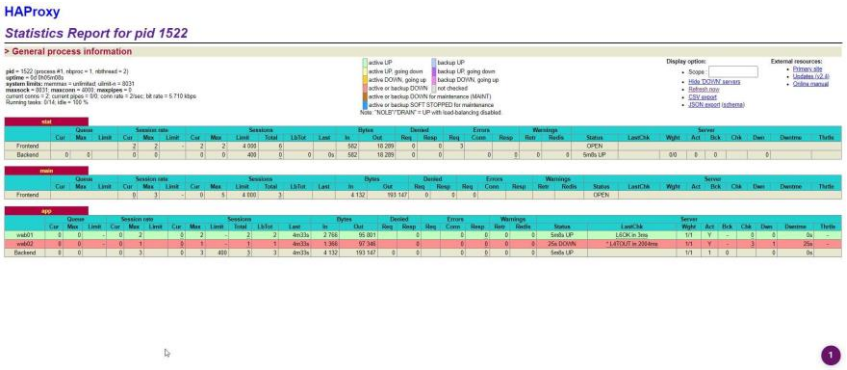


On peut également accéder au site web depuis l'adresse IP du ha01



D. Exercices

1. On coupe un serveur web



Il apparait donc en rouge

2. On peut mettre différents modes :
   - Round robin: répartit les requêtes à tour de rôle entre les serveurs disponibles.

   Balance roundrobin
   - Least connections: dirige les requêtes vers le serveur avec le moins de connexions actives. **Balance leastconn**
   - Weighted least connections: fonctionne comme least connections mais permet de pondérer la charge en fonction de la capacité des serveurs**. balance weighted_least_connections**

   server srv1 100
   server srv2 50
   - Source IP: affecte les requêtes d'une même adresse IP au même serveur.
   - URI: répartit les requêtes selon l'URI demandée.

#4

A. Installation d'un 2ème Haproxy

1. Snapshot de la vm
2. Clone de la vm
3. Changement de carte réseau pour avoir une nouvelle IP
4. Renommer la machine
5. Ajout dans le fichier host pour chaque machine

```
::1             localhost loca
192.168.134.132 web01
192.168.134.137 web02
192.168.134.141 HA02
```

On verifie avec un curl

```
success
[matthieu@localhost ~]$ curl web02
<!DOCTYPE HTML>
<html>
        <head>
                <title>Pricing Plans and Subscription Payment | 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-scale=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"));
                                });
```

B. Redondance Haproxy

1. Ouverture du port 80

Sur les 2 haproxy on ouvre le port 80

```
[matthieu@localhost ~]$ sudo firewall-cmd --permanent --add-port=80/tcp
success
[matthieu@localhost ~]$ sudo firewall-cmd --reload
success
[matthieu@localhost ~]$
```

2. Installation du pacemaker sur les 2 haproxy

```
dnf --enablerepo=highavailability -y install pacemaker pcs
```

```
  perl-Socket-4:2.031-4.el9.x86_64                perl-Storable-1:3.21-460.el9.x86_64
  perl-Symbol-1.08-480.el9.noarch                 perl-Term-ANSIColor-5.01-461.el9.noarch
  perl-Term-Cap-1.17-460.el9.noarch               perl-Text-ParseWords-3.30-460.el9.noarch
  perl-Text-Tabs+Wrap-2013.0523-460.el9.noarch    perl-Time-Local-2:1.300-7.el9.noarch
  perl-TimeDate-1:2.33-6.el9.noarch               perl-URI-5.09-3.el9.noarch
  perl-base-2.27-480.el9.noarch                   perl-constant-1.33-461.el9.noarch
  perl-if-0.60.800-480.el9.noarch                 perl-interpreter-4:5.32.1-480.el9.x86_64
  perl-libnet-3.13-4.el9.noarch                   perl-libs-4:5.32.1-480.el9.x86_64
  perl-mro-1.23-480.el9.x86_64                    perl-overload-1.31-480.el9.noarch
  perl-overloading-0.02-480.el9.noarch            perl-parent-1:0.238-460.el9.noarch
  perl-podlators-1:4.14-460.el9.noarch            perl-subs-1.03-480.el9.noarch
  perl-vars-1.05-480.el9.noarch                   pkgconf-1.7.3-10.el9.x86_64
  pkgconf-m4-1.7.3-10.el9.noarch                  pkgconf-pkg-config-1.7.3-10.el9.x86_64
  python3-cffi-1.14.5-5.el9.x86_64                python3-cryptography-36.0.1-4.el9.x86_64
  python3-lxml-4.6.5-3.el9.x86_64                 python3-ply-3.11-14.el9.0.1.noarch
  python3-pycparser-2.20-6.el9.noarch             python3-pycurl-7.43.0.6-8.el9.x86_64
  python3-pyparsing-2.4.7-9.el9.noarch            python3-pyyaml-5.4.1-6.el9.x86_64
  python3-setuptools-53.0.0-12.el9.noarch         quota-1:4.06-6.el9.x86_64
  quota-nls-1:4.06-6.el9.noarch                   resource-agents-4.10.0-44.el9_3.0.1.x86_64
  rocky-logos-90.15-2.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_3.5.x86_64
  tar-2:1.34-6.el9_1.x86_64

Complete!
[matthieu@localhost ~]$
```

3. Mise en place du cluster

On active le service au démarrage

```
[matthieu@localhost ~]$ sudo systemctl enable pcsd
[sudo] password for matthieu:
Created symlink /etc/systemd/system/multi-user.target.wants/pcsd.service → /usr/lib/systemd/system/pcsd.service.
[matthieu@localhost ~]$
```

4. Création d'un user

user : 'hacluster'

mdp : tphaproxy

```
[matthieu@localhost ~]$ sudo passwd hacluster
Changing password for user hacluster.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[matthieu@localhost ~]$
```

5. Authentification entre nos 2 nodes :

```
[matthieu@ha02 ~]$ sudo pcs host auth HA01 HA02
[sudo] password for matthieu:
Username: hacluster
Password:
HA02: Authorized
HA01: Authorized
[matthieu@ha02 ~]$
```

6. Déploiement du cluster

A faire que ha01

pcs cluster setup ha_cluster HA01 HA02

```
[matthieu@ha01 ~]$ sudo pcs cluster setup ha_cluster HA01 HA02
No addresses specified for host 'HA01', using 'ha01'
No addresses specified for host 'HA02', using 'ha02'
Destroying cluster on hosts: 'HA01', 'HA02'...
HA02: Successfully destroyed cluster
HA01: Successfully destroyed cluster
Requesting remove 'pcsd settings' from 'HA01', 'HA02'
HA01: successful removal of the file 'pcsd settings'
HA02: successful removal of the file 'pcsd settings'
Sending 'corosync authkey', 'pacemaker authkey' to 'HA01', 'HA0
HA01: successful distribution of the file 'corosync authkey'
HA01: successful distribution of the file 'pacemaker authkey'
HA02: successful distribution of the file 'corosync authkey'
HA02: successful distribution of the file 'pacemaker authkey'
Sending 'corosync.conf' to 'HA01', 'HA02'
HA01: successful distribution of the file 'corosync.conf'
HA02: successful distribution of the file 'corosync.conf'
Cluster has been successfully set up.
[matthieu@ha01 ~]$
```

Pcs cluster start –all	Pour lancer le cluster
Pcs cluster enable –all	Pour lancer le cluster à chaque démarrage

```
[matthieu@ha01 ~]$ sudo pcs cluster enable --all
HA01: Cluster Enabled
HA02: Cluster Enabled
[matthieu@ha01 ~]$
```

7. Vérification des erreurs

```
crm_verify -L -V
```

```
[matthieu@ha01 ~]$ sudo 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
[matthieu@ha01 ~]$
```

8. Correction des erreurs

```
[matthieu@ha01 ~]$ sudo pcs property set stonith-enabled=false
[matthieu@ha01 ~]$ sudo pcs property set no-quorum-policy=ignore
```

9. Re vérification

```
[matthieu@ha01 ~]$ sudo crm_verify -L -V
[matthieu@ha01 ~]$
```

10. Création d'une adresse IP virtuel

Création d'une adresse IP virtuelle permettant de rediriger le trafic sur l'un ou l'autre nœud

```
pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=192.168.134.130 cidr_netmask=24 op monitor interval=30s
```

```
[matthieu@ha01 ~]$ sudo pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=192.168.134.130 cidr_net
mask=24 op monitor interval=30s
[matthieu@ha01 ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:3d:9d:44 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.134.140/24 brd 192.168.134.255 scope global dynamic noprefixroute ens160
       valid_lft 1180sec preferred_lft 1180sec
    inet 192.168.134.130/24 brd 192.168.134.255 scope global secondary ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe3d:9d44/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[matthieu@ha01 ~]$
```

```
[matthieu@ha01 ~]$ sudo pcs status resources
  * virtual_ip  (ocf:heartbeat:IPaddr2):        Started HA01
[matthieu@ha01 ~]$
```

11. Installation de l'agent sur les 2 serveurs

```
[matthieu@ha01 heartbeat]$ sudo curl -O https://raw.githubusercontent.com/thisismitch/clusteragents/master/haproxy
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    14  100    14    0     0     50      0 --:--:-- --:--:-- --:--:--    50
```

Et le rendre exécutable

```
[matthieu@ha01 heartbeat]$ sudo chmod +x haproxy
```

Afin de voir l'ensemble des services :

pcs resource agents ocf:heartbeat

```
[matthieu@ha01 heartbeat]$ pcs resource agents ocf: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
mysql
nagios
named
nfsnotify
nfsserver
nginx
NodeUtilization
openstack-cinder-volume
openstack-floating-ip
openstack-info
openstack-virtual-ip
oraasm
oracle
```