LYCÉE ROBERT SCHUMAN METZ

# Projet SISR3-03: Répartition et équilibrage de charge sur une plate-forme Web avec réplication des données.

**Veynand-Saint Fiacre Lucille**

**14/12/2020**
**15/12/2020**
**29/12/2021**
**30/12/2021**

# Table des matières

# 1 Introduction

***Problématique :***

Mettre en place une configuration sur un serveur nommé HAProxy pour permettre de la répartition de charge sur 2 serveurs Web, préalablement installés avec réplication multi-maître des données.

# 2 Contexte de travail et gestion des configurations

**(voir *contexte.docx*)**

# 3 Gestion du travail en équipe et gestion du projet

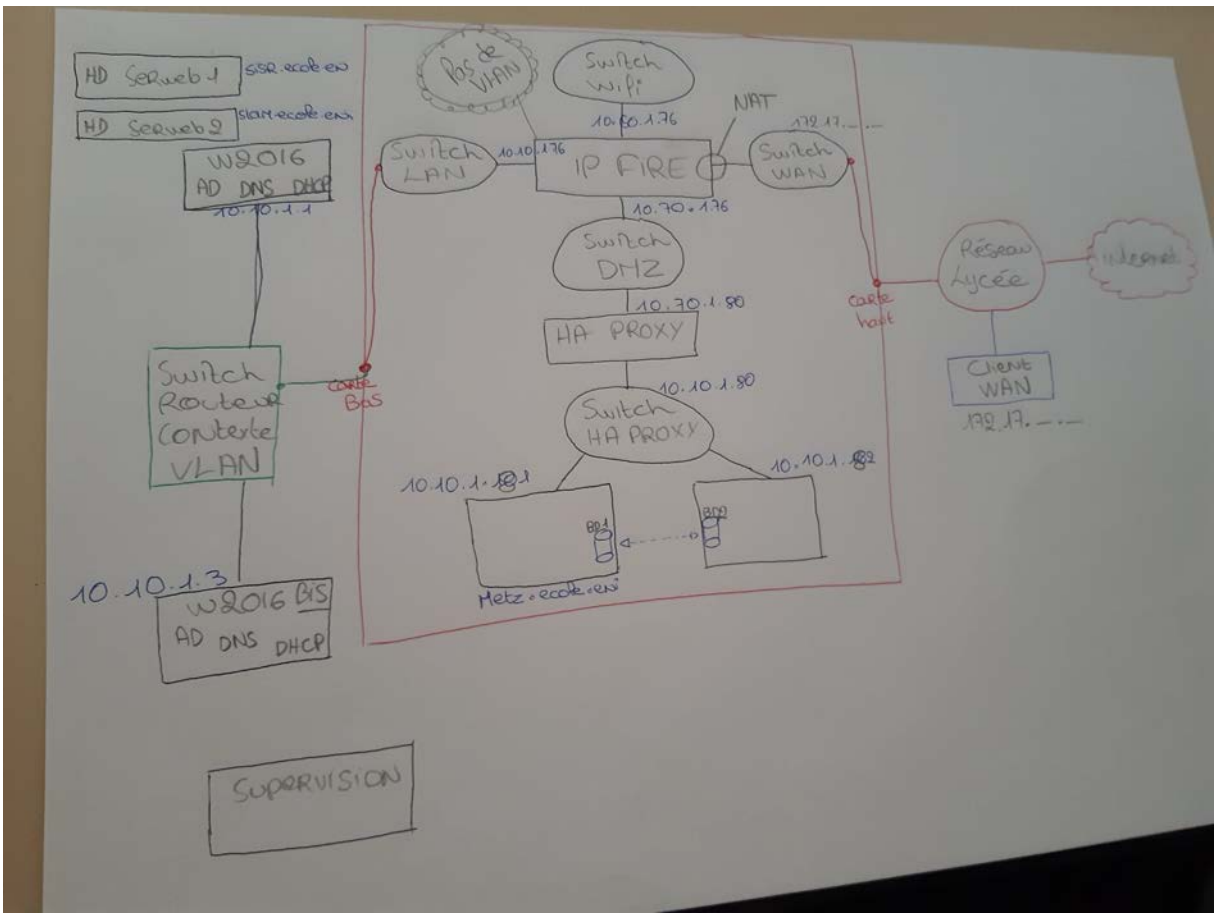
| | | |
|---|---|---|
| VEYNAND SAINT FIACRE Lucille | 14/12/2020<br><br>15/12/2020 | Jusqu'à question 12 |

# 4 Activités compétences du référentiel du BTS SIO (reseaucerta.org)

| | |
|---|---|
| A1.3.1 Test d'intégration et d'acceptation d'un service | · C1.3.1.1 Mettre en place l'environnement de test du service<br>· C1.3.1.2 Tester le service<br>· C1.3.1.3 Rédiger le rapport de test |
| A1.4.3 Gestion des ressources | · C1.4.3.1 Recenser les ressources humaines, matérielles, logicielles et budgétaires nécessaires à l'exécution du projet et de ses tâches personnelles. |
| A2.1.2 Évaluation et maintien de la qualité d'un service | · C2.1.2.1 Analyser les indicateurs de qualité du service. |
| A2.2.3 Réponse à une interruption de service | · C2.2.3.1 Appliquer la procédure de continuité du service en mode dégradé<br>· C2.2.3.2 Appliquer la procédure de reprise du service |
| **A3.1.1 Proposition d'une solution d'infrastructure** | · **C3.1.1.1 Lister les composants matériels et logiciels nécessaires à la prise en charge des processus, des flux d'information et de leur rôle.**<br>· **C3.1.1.2 Caractériser les éléments d'interconnexion, les services, les serveurs et les équipements terminaux nécessaires.** |

| | |
|---|---|
| | · **C3.1.1.3 Caractériser les éléments permettant d'assurer la qualité et la sécurité des services**. |
| **A3.1.3 Prise en compte du niveau de sécurité nécessaire à une infrastructure** | · C3.1.3.2 Proposer une solution de sécurité compatible avec les contraintes techniques, financières, juridiques et organisationnelles.<br>· C3.1.3.3 Décrire une solution de sécurité et les risques couverts. |
| **A3.2.1 Installation et configuration d'éléments d'infrastructure** | · **C3.2.1.1 Installer et configurer un élément d'interconnexion, un service, un serveur, un équipement terminal utilisateur**<br>· **C3.2.1.3 Installer et configurer des éléments de sécurité permettant d'assurer la protection du système informatique.** |
| **A3.2.3 Mise à jour de la documentation technique d'une solution d'infrastructure** | · C3.2.3.1 Repérer les éléments de la documentation à mettre à jour.<br>· C3.2.3.2 Mettre à jour la documentation. |
| **A5.1.2 Recueil d'informations sur une configuration et ses éléments** | · C5.1.2.1 Renseigner les événements relatifs au cycle de vie d'un élément de la configuration<br>· C5.1.2.2 Actualiser les caractéristiques des éléments de la configuration. |
| **A5.1.3 Suivi d'une configuration et de ses éléments** | · C5.1.3.1 Contrôler et auditer les éléments de la configuration<br>· C5.1.3.2 Reconstituer un historique des modifications effectuées sur les éléments de la configuration<br>· C5.1.3.3 Identifier les éléments de la configuration à modifier ou à remplacer. |
| **A5.2.3 Repérage des compléments de formation ou d'auto-formation utiles à l'acquisition de nouvelles compétences** | · C5.2.3.1 Identifier les besoins de formation pour mettre en œuvre une technologie, un composant, un outil ou une méthode. |
| **A5.2.4 Étude d'une technologie, d'un composant, d'un outil ou d'une méthode** | · C5.2.4.1 Se documenter à propos d'une technologie, d'un composant, d'un outil ou d'une méthode. |

# 5 Le Maquettage




# 6 Le projet en détail.

## 1. Installer et configurer les deux serveurs web sécurisé.

On l'a déjà configuré le serveur web sécurisé dans le projet 1 de SISR3
On clone alors la VM serveurwebsecu1
Et on les nomment respectivement serveurwebsecu1 et serveurwebsecu2

```
root@serwebsecu1:/home/sisr# uname -a
Linux serwebsecu1 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u6 x86_64 GNU/Linux
root@serwebsecu1:/home/sisr#
```

```
root@serwebsecu2:/home/sisr# uname -a
Linux serwebsecu2 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u6 x86_64 GNU/Linux
```

Et on leur donne respectivement les adresses IP suivantes

```
root@serwebsecu1:/home/sisr# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:1f:44:36
          inet adr:10.10.1.101  Bcast:10.10.1.255  Masque:255.255.255.0
          adr inet6: fe80::215:5dff:fe1f:4436/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:135 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:15589 (15.2 KiB)  TX bytes:29538 (28.8 KiB)
```

```
root@serwebsecu2:/home/sisr# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:1f:44:35
          inet adr:10.10.1.102  Bcast:10.10.1.255  Masque:255.255.255.0
          adr inet6: fe80::215:5dff:fe1f:4435/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:122 errors:0 dropped:0 overruns:0 frame:0
          TX packets:90 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:23699 (23.1 KiB)  TX bytes:21620 (21.1 KiB)
```

## 2. Donner la configuration des fichiers DNS, WEB, SSL des deux serveurs.

Vu que l'on a répliqué les serveurs on ne va que donner la configuration de serwebsecu1

### Configuration du serveur DNS pour serwebsecu1

```
root@serwebsecu1:/home/sisr# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 127.0.0.1
root@serwebsecu1:/home/sisr# cat /etc/bind/named.conf.local
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "ecole.eni" {
        type master;
        file "/etc/bind/ecole.eni";
};

zone "1.10.10.in-addr.arpa" {
        type master;
        file "/etc/bind/ecole.eni_inv";
};
root@serwebsecu1:/home/sisr# cat /etc/bind/ecole.eni
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
sisr    IN      A       10.10.1.101
slam    IN      A       10.10.1.101
root@serwebsecu1:/home/sisr# cat /etc/bind/ecole.eni_inv
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
101     IN      PTR     sisr
101     IN      PTR     slam
```

**Configuration du serveur DNS pour serwebsecu2**

```
root@serwebsecu1:/home/sisr# cat /etc/bind/named.conf.local
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "ecole.eni" {
        type master;
        file "/etc/bind/ecole.eni";
};

zone "1.10.10.in-addr.arpa" {
        type master;
        file "/etc/bind/ecole.eni_inv";
};
root@serwebsecu2:/home/sisr# cat /etc/bind/ecole.eni
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
sisr    IN      A       10.10.1.102
slam    IN      A       10.10.1.102
root@serwebsecu2:/home/sisr# cat /etc/bind/ecole.eni_inv
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
102     IN      PTR     sisr
102     IN      PTR     slam
```

**Configuration du serveur Apache pour serwebsecu1**

```
root@serwebsecu1:/home/sisr# cat /etc/apache2/apache2.conf | egrep -v "(^#.*)"



LockFile ${APACHE_LOCK_DIR}/accept.lock

PidFile ${APACHE_PID_FILE}

Timeout 300

KeepAlive On

MaxKeepAliveRequests 100

KeepAliveTimeout 5


<IfModule mpm_prefork_module>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers      10
    MaxClients          150
    MaxRequestsPerChild   0
</IfModule>

<IfModule mpm_worker_module>
    StartServers          2
    MinSpareThreads      25
    MaxSpareThreads      75
    ThreadLimit          64
    ThreadsPerChild      25
    MaxClients          150
    MaxRequestsPerChild   0
</IfModule>

<IfModule mpm_event_module>
    StartServers          2
    MinSpareThreads      25
```

```
    MaxSpareThreads      75
    ThreadLimit          64
    ThreadsPerChild      25
    MaxClients          150
    MaxRequestsPerChild   0
</IfModule>

User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}


AccessFileName .htaccess

<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
    Satisfy all
</Files>

DefaultType None


HostnameLookups Off

ErrorLog ${APACHE_LOG_DIR}/error.log

LogLevel warn

Include mods-enabled/*.load
Include mods-enabled/*.conf

Include ports.conf

LogFormat "%v:%p %h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %0" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

```
root@serwebsecu1:/home/sisr# cat /etc/apache2/sites-available/default
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        NameVirtualHost 10.10.1.101:80
        DocumentRoot /var/www
        DirectoryIndex accueil.html index.html
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
#               Order allow,deny
#               Allow from all
        </Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
root@serwebsecu1:/home/sisr# cat /etc/apache2/sites-available/default-ssl
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
        NameVirtualHost 10.10.1.101:443
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www
        DirectoryIndex accueil.html index.html
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined

        #   SSL Engine Switch:
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on

        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
```

```
#   the ssl-cert package. See
#   /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
#   If both key and certificate are stored in the same file, only the
#   SSLCertificateFile directive is needed.
SSLCertificateFile    /etc/apache2/certs/private_key.crt
SSLCertificateKeyFile /etc/apache2/certs/private_key.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCACertificatePath /etc/ssl/certs/
#SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs for client
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCARevocationPath /etc/apache2/ssl.crl/
#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
```

```
#   issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   Access Control:
#   With SSLRequire you can do per-directory access control based
#   on arbitrary complex boolean expressions containing server
#   variable checks and other lookup directives.  The syntax is a
#   mixture between C and Perl.  See the mod_ssl documentation
#   for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#            and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#            and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#            and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#            and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20       ) \
#           or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#     Translate the client X.509 into a Basic Authorisation.  This means that
#     the standard Auth/DBMAuth methods can be used for access control.  The
#     user name is the `one line' version of the client's X.509 certificate.
#     Note that no password is obtained from the user. Every entry in the user
#     file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#     This exports two additional environment variables: SSL_CLIENT_CERT and
#     SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#     server (always existing) and the client (only existing when client
#     authentication is used). This can be used to import the certificates
#     into CGI scripts.
#   o StdEnvVars:
#     This exports the standard SSL/TLS related `SSL_*' environment variables.
#     Per default this exportation is switched off for performance reasons,
#     because the extraction step is an expensive operation and is usually
#     useless for serving static content. So one usually enables the
#     exportation for CGI and SSI requests only.
#   o StrictRequire:
#     This denies access when "SSLRequireSSL" or "SSLRequire" applied even
```

```
#     under a "Satisfy any" situation, i.e. when it applies access is denied
#     and no other module can change it.
#   o OptRenegotiate:
#     This enables optimized SSL connection renegotiation handling when SSL
#     directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#     This forces an unclean shutdown when the connection is closed, i.e. no
#     SSL close notify alert is send or allowed to received.  This violates
#     the SSL/TLS standard but is needed for some brain-dead browsers. Use
#     this when you receive I/O errors because of the standard approach where
#     mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#     This forces an accurate shutdown when the connection is closed, i.e. a
#     SSL close notify alert is send and mod_ssl waits for the close notify
#     alert of the client. This is 100% SSL/TLS standard compliant, but in
#     practice often causes hanging connections with brain-dead browsers. Use
#     this only for browsers where you know that their SSL implementation
#     works correctly.
#   Notice: Most problems of broken clients are also related to the HTTP
#   keep-alive facility, so you usually additionally want to disable
#   keep-alive for those clients, too. Use variable "nokeepalive" for this.
#   Similarly, one has to force some clients to use HTTP/1.0 to workaround
#   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
#   "force-response-1.0" for this.
BrowserMatch "MSIE [2-6]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
# MSIE 7 and newer should be able to use keepalive
```

```
        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
</IfModule>
root@serwebsecu1:/home/sisr# cat /etc/apache2/sites-available/sisr
<VirtualHost 10.10.1.101:80>
        DocumentRoot "/var/www/sisr"
        DirectoryIndex sisr.php
        ServerName sisr.ecole.eni
<Directory /var/www/sisr>
        AllowOverride AuthConfig
</Directory>
</VirtualHost>

<VirtualHost 10.10.1.101:443>
        SSLEngine on
        SSLCertificateFile /etc/apache2/certs/private_key.crt
        SSLCertificateKeyFile /etc/apache2/certs/private_key.key
        DocumentRoot "/var/www/sisr"
        DirectoryIndex sisr.php
        ServerName sisr.ecole.eni
<Directory /var/www/sisr>
        AllowOverride AuthConfig
</Directory>
</VirtualHost>
root@serwebsecu1:/home/sisr# cat /etc/apache2/sites-available/slam
<VirtualHost 10.10.1.101:80>
        DocumentRoot "/var/www/slam"
        DirectoryIndex slam.php
        ServerName slam.ecole.eni
<Directory /var/www/slam>
        AllowOverride AuthConfig
</Directory>
</VirtualHost>

<VirtualHost 10.10.1.101:443>
        SSLEngine on
        SSLCertificateFile /etc/apache2/certs/private_key.crt
        SSLCertificateKeyFile /etc/apache2/certs/private_key.key
        DocumentRoot "/var/www/slam"
        DirectoryIndex slam.php
        ServerName slam.ecole.eni
<Directory /var/www/slam>
        AllowOverride AuthConfig
</Directory>
</VirtualHost>
```

**Configuration du serveur Apache pour serwebsecu2**

```
root@serwebsecu1:/home/sisr# cat /etc/apache2/apache2.conf | egrep -v "(^#.*)"



LockFile ${APACHE_LOCK_DIR}/accept.lock

PidFile ${APACHE_PID_FILE}

Timeout 300

KeepAlive On

MaxKeepAliveRequests 100

KeepAliveTimeout 5


<IfModule mpm_prefork_module>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers      10
    MaxClients          150
    MaxRequestsPerChild   0
</IfModule>

<IfModule mpm_worker_module>
    StartServers          2
    MinSpareThreads      25
    MaxSpareThreads      75
    ThreadLimit          64
    ThreadsPerChild      25
    MaxClients          150
    MaxRequestsPerChild   0
</IfModule>

<IfModule mpm_event_module>
    StartServers          2
    MinSpareThreads      25
```

```
    MaxSpareThreads      75
    ThreadLimit          64
    ThreadsPerChild      25
    MaxClients          150
    MaxRequestsPerChild   0
</IfModule>

User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}


AccessFileName .htaccess

<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
    Satisfy all
</Files>

DefaultType None


HostnameLookups Off

ErrorLog ${APACHE_LOG_DIR}/error.log

LogLevel warn

Include mods-enabled/*.load
Include mods-enabled/*.conf

Include ports.conf

LogFormat "%v:%p %h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %0" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

```
root@serwebsecu2:/home/sisr# cat /etc/apache2/sites-available/default
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        NameVirtualHost 10.10.1.102:80
        DocumentRoot /var/www
        DirectoryIndex accueil.html index.html
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
#               Order allow,deny
#               Allow from all
        </Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
root@serwebsecu2:/home/sisr# cat /etc/apache2/sites-available/default-ssl
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
        NameVirtualHost 10.10.1.102:443
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www
        DirectoryIndex accueil.html index.html
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined

        #   SSL Engine Switch:
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on

        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
```

```
#   the ssl-cert package. See
#   /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
#   If both key and certificate are stored in the same file, only the
#   SSLCertificateFile directive is needed.
SSLCertificateFile    /etc/apache2/certs/private_key.crt
SSLCertificateKeyFile /etc/apache2/certs/private_key.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCACertificatePath /etc/ssl/certs/
#SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs for client
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCARevocationPath /etc/apache2/ssl.crl/
#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
```

```
#   issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   Access Control:
#   With SSLRequire you can do per-directory access control based
#   on arbitrary complex boolean expressions containing server
#   variable checks and other lookup directives.  The syntax is a
#   mixture between C and Perl.  See the mod_ssl documentation
#   for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#            and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#            and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#            and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#            and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20       ) \
#           or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#     Translate the client X.509 into a Basic Authorisation.  This means that
#     the standard Auth/DBMAuth methods can be used for access control.  The
#     user name is the `one line' version of the client's X.509 certificate.
#     Note that no password is obtained from the user. Every entry in the user
#     file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#     This exports two additional environment variables: SSL_CLIENT_CERT and
#     SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#     server (always existing) and the client (only existing when client
#     authentication is used). This can be used to import the certificates
#     into CGI scripts.
#   o StdEnvVars:
#     This exports the standard SSL/TLS related `SSL_*' environment variables.
#     Per default this exportation is switched off for performance reasons,
#     because the extraction step is an expensive operation and is usually
#     useless for serving static content. So one usually enables the
#     exportation for CGI and SSI requests only.
#   o StrictRequire:
#     This denies access when "SSLRequireSSL" or "SSLRequire" applied even
```

```
#     under a "Satisfy any" situation, i.e. when it applies access is denied
#     and no other module can change it.
#   o OptRenegotiate:
#     This enables optimized SSL connection renegotiation handling when SSL
#     directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#     This forces an unclean shutdown when the connection is closed, i.e. no
#     SSL close notify alert is send or allowed to received.  This violates
#     the SSL/TLS standard but is needed for some brain-dead browsers. Use
#     this when you receive I/O errors because of the standard approach where
#     mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#     This forces an accurate shutdown when the connection is closed, i.e. a
#     SSL close notify alert is send and mod_ssl waits for the close notify
#     alert of the client. This is 100% SSL/TLS standard compliant, but in
#     practice often causes hanging connections with brain-dead browsers. Use
#     this only for browsers where you know that their SSL implementation
#     works correctly.
#   Notice: Most problems of broken clients are also related to the HTTP
#   keep-alive facility, so you usually additionally want to disable
#   keep-alive for those clients, too. Use variable "nokeepalive" for this.
#   Similarly, one has to force some clients to use HTTP/1.0 to workaround
#   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
#   "force-response-1.0" for this.
BrowserMatch "MSIE [2-6]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
# MSIE 7 and newer should be able to use keepalive
```

```
        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
</IfModule>
root@serwebsecu2:/home/sisr# cat /etc/apache2/sites-available/sisr
<VirtualHost 10.10.1.102:80>
        DocumentRoot "/var/www/sisr"
        DirectoryIndex sisr.php
        ServerName sisr.ecole.eni
<Directory /var/www/sisr>
        AllowOverride AuthConfig
</Directory>
</VirtualHost>

<VirtualHost 10.10.1.102:443>
        SSLEngine on
        SSLCertificateFile /etc/apache2/certs/private_key.crt
        SSLCertificateKeyFile /etc/apache2/certs/private_key.key
        DocumentRoot "/var/www/sisr"
        DirectoryIndex sisr.php
        ServerName sisr.ecole.eni
<Directory /var/www/sisr>
        AllowOverride AuthConfig
</Directory>
</VirtualHost>
```

**3.** Donner la configuration de la réplication des données.

**Réplication des données sur serwebsecu1**

```
mysql> show slave status\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 10.10.1.62
                  Master_User: replicateur
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000003
          Read_Master_Log_Pos: 1990
               Relay_Log_File: mysqld-relay-bin.000003
                Relay_Log_Pos: 425
        Relay_Master_Log_File: mysql-bin.000003
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 1990
              Relay_Log_Space: 2439
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 2
1 row in set (0.00 sec)
```

**Réplication des données sur serwebsecu2**

```
mysql> show slave status \G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 10.10.1.61
                  Master_User: replicateur
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000004
          Read_Master_Log_Pos: 2157
               Relay_Log_File: mysqld-relay-bin.000003
                Relay_Log_Pos: 253
        Relay_Master_Log_File: mysql-bin.000004
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 2157
              Relay_Log_Space: 2606
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 1
1 row in set (0.00 sec)
```

**4.** Tester les services DNS WEB SSL Mysql et la réplication de base de données sur les deux serveurs.

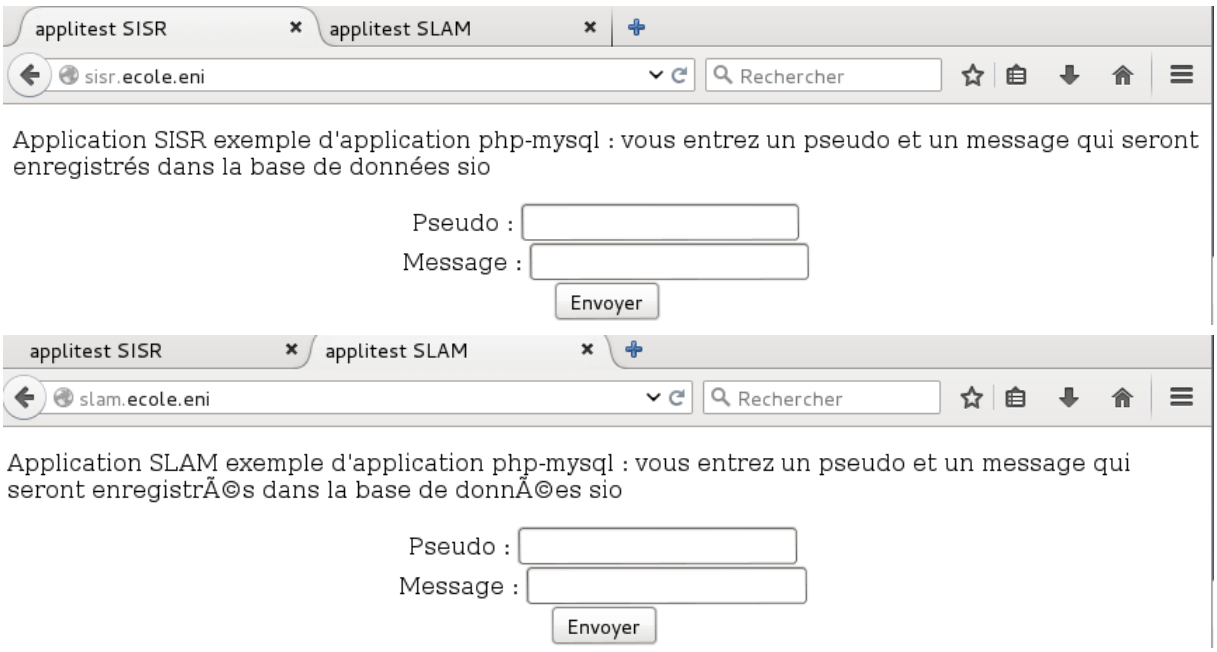
**Test du service DNS sur serwebsecu1**

```
root@serwebsecu1:/home/sisr# nslookup sisr-secu.ecole.eni
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   sisr-secu.ecole.eni
Address: 10.10.1.61

root@serwebsecu1:/home/sisr# nslookup slam-secu.ecole.eni
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   slam-secu.ecole.eni
Address: 10.10.1.61

root@serwebsecu1:/home/sisr# nslookup 10.10.1.61
Server:         127.0.0.1
Address:        127.0.0.1#53

61.1.10.10.in-addr.arpa name = sisr-secu.1.10.10.in-addr.arpa.
61.1.10.10.in-addr.arpa name = slam-secu.1.10.10.in-addr.arpa.
```

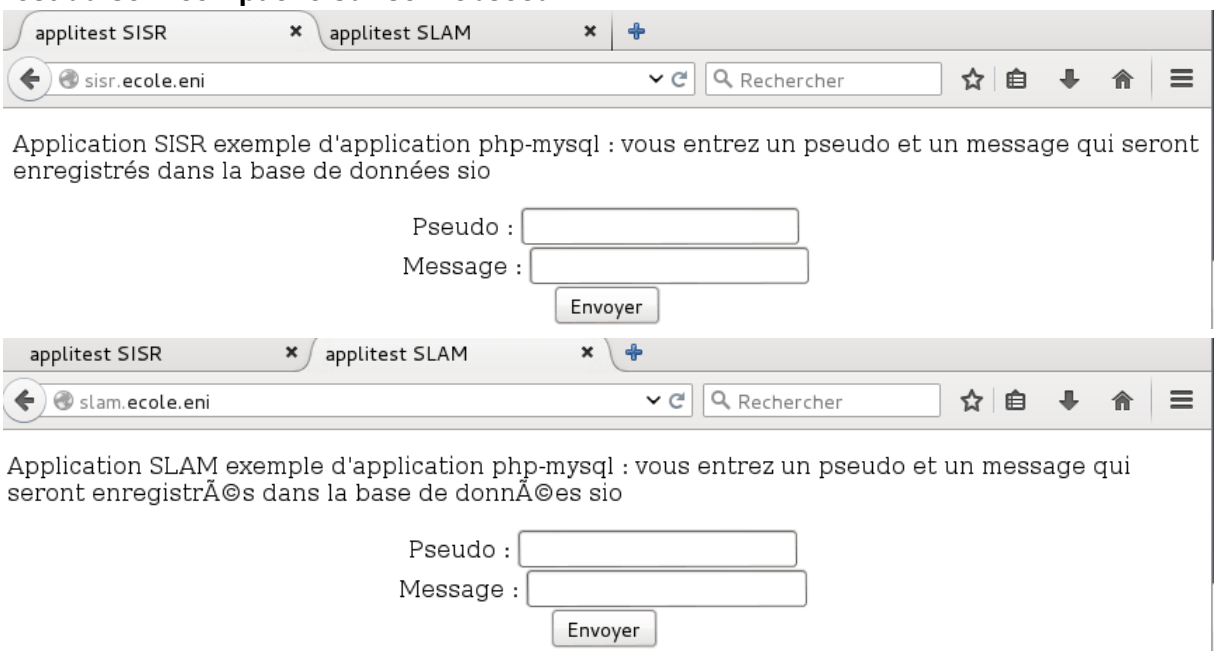
**Test du service DNS sur serwebsecu2**

```
root@serwebsecu1:/home/sisr# nslookup sisr-secu.ecole.eni
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   sisr-secu.ecole.eni
Address: 10.10.1.62

root@serwebsecu1:/home/sisr# nslookup slam-secu.ecole.eni
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   slam-secu.ecole.eni
Address: 10.10.1.62

root@serwebsecu1:/home/sisr# nslookup 10.10.1.62
Server:         127.0.0.1
Address:        127.0.0.1#53

62.1.10.10.in-addr.arpa name = sisr-secu.1.10.10.in-addr.arpa.
62.1.10.10.in-addr.arpa name = slam-secu.1.10.10.in-addr.arpa.
```

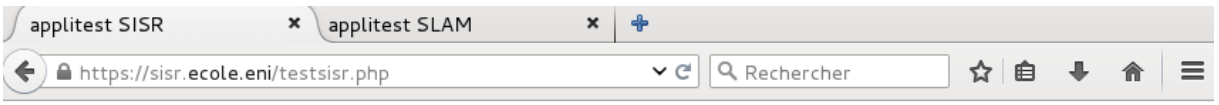
**Test du service Apache sur serwebsecu1**




**Test du service Apache sur serwebsecu2**




**Test de réplication des données de la base de données**




Application SISR exemple d'application php-mysql : vous entrez un pseudo et un message qui seront enregistrés dans la base de données sio

Pseudo :
Message :
Envoyer

**Hamid** envoie le message: Les domaines de diffusion de serwebsecu2

**Hamid** envoie le message: Les domaines de diffusion de serwebsecu1

```
mysql> select * from sisr;
+----+--------+-----------------------------------------+
| id | pseudo | message                                 |
+----+--------+-----------------------------------------+
|  1 | Hamid  | Les domaines de diffusion de serwebsecu1 |
|  2 | Hamid  | Les domaines de diffusion de serwebsecu2 |
+----+--------+-----------------------------------------+
2 rows in set (0.00 sec)
```

On tente de supprimer la table SISR à partir de serwebsecu1

Résultat sur serwebsecu1:

```
mysql> select * from sisr;
Empty set (0.00 sec)
```

Résultat sur serwebsecu2:

```
mysql> select * from sisr;
Empty set (0.00 sec)
```

**5.** Configurer le serveur HAPROXY d'adresse 10.10.1.61/24 (sur le réseau privé) et 10.70.1.61/24 (sur le réseau public)

```
root@HAProxy:/home/sisr# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:15:5d:1f:44:52 brd ff:ff:ff:ff:ff:ff
    inet 10.70.1.63/24 brd 10.70.1.255 scope global eth0
    inet6 fe80::215:5dff:fe1f:4452/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:15:5d:1f:44:53 brd ff:ff:ff:ff:ff:ff
    inet 10.10.1.63/24 brd 10.10.1.255 scope global eth1
    inet6 fe80::215:5dff:fe1f:4453/64 scope link
       valid_lft forever preferred_lft forever
```

**6.** Installer et démarrer HAproxy.

[....] Restarting haproxy: haproxy[ALERT] 236/215943 (8688) : config : no <listen> line. Nothing to do !
[ALERT] 236/215943 (8688) : Fatal errors found in configuration.  failed!

Il est nécessaire de procéder à une configuration minimale.

HAProxy est déjà installé de base

Pour activer HAProxy, il faut mettre la valeur « 1 » à la variable ENABLED qu'on vient de créer dans le fichier  /etc/default/haproxy.

```
root@sisr:/home/sisr# cat /etc/default/haproxy
# Defaults file for HAProxy
#
# This is sourced by both, the initscript and the systemd unit file, so do not
# treat it as a shell script fragment.

# Change the config file location if needed
#CONFIG="/etc/haproxy/haproxy.cfg"

# Add extra flags here, see haproxy(1) for a few options
#EXTRAOPTS="-de -m 16"

ENABLED=1
```

On fait ensuite la commande service haproxy restart

```
root@sisr:/home/sisr# service haproxy restart
[ ok ] Restarting haproxy: haproxy.
```

**7.** En examinant le script de démarrage d'HAProxy, dont un extrait est présenté ci-dessous, expliquez pourquoi il est nécessaire de modifier la variable « ENABLED » du fichier /etc/default/haproxy.

Car il vérifie si la variable ENABLED est définie à 1, s'il elle ne l'est pas le service de ne démarre pas

**8.** Procédez à une première configuration d'HAProxy avec mise en place des statistiques.

```
        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http

frontend proxypublic
        bind 10.70.1.63:80
        default_backend fermeweb

backend fermeweb
        balance roundrobin
        option httpclose
        option httpchk HEAD / HTTP/1.0
        server serwebsecu1 10.10.1.61:80 weight 66 check
        server serwebsecu2 10.10.1.62:80 weight 33 check
        stats refresh 30s
        stats auth admin:admin
        stats uri /stats
```

On redémarre ensuite le service HAProxy

```
root@sisr:/home/sisr# service haproxy restart
[ ok ] Restarting haproxy: haproxy.
```

**9.** Proposez et réalisez des tests (tests de non régression compris) permettant de vérifier que la solution est opérationnelle.

```
root@sisr:/home/sisr# haproxy -c -f /etc/haproxy/haproxy.cfg
Configuration file is valid
```

```
root@sisr:/home/sisr# netstat -tpnl | grep haproxy
tcp        0      0 10.70.1.200:80          0.0.0.0:*               LISTEN
2601/haproxy
```

**10.** Procédez à une nouvelle configuration d'HAProxy en faisant l'hypothèse que le serveur initialement de secours HAProxy*serweb2* est trois fois moins puissant que le serveur maître.

```
        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http

frontend proxypublic
        bind 10.70.1.63:80
        default_backend fermeweb

backend fermeweb
        balance roundrobin
        option httpclose
        option httpchk HEAD / HTTP/1.0
        server serwebsecu1 10.10.1.61:80 weight 66 check
        server serwebsecu2 10.10.1.62:80 weight 33 check
        stats refresh 30s
        stats auth admin:admin
        stats uri /stats
```

On vérifie ensuite le fichier de configuration et on redémarre le service.

```
root@HAProxy:/home/sisr# haproxy -c -f /etc/haproxy/haproxy.cfg
Configuration file is valid
root@HAProxy:/home/sisr# service haproxy restart
[ ok ] Restarting haproxy: haproxy.
root@HAProxy:/home/sisr#
```

**11.** Proposez et réalisez des tests permettant de vérifier que la solution est opérationnelle.

On reload la page de multiple fois: une fois sur trois ont le serveurweb2

SERWEB1 Application sisr : exemple d'application php-mysql : vous entrez un pseudo et un message qui seront enregistrÃ©s dans la base de donnÃ©es sio table sisr

Pseudo :
Message :
Envoyer

SERWEB1 Application sisr : exemple d'application php-mysql : vous entrez un pseudo et un message qui seront enregistrÃ©s dans la base de donnÃ©es sio table sisr

Pseudo :
Message :
Envoyer

SERWEB2 Application sisr : exemple d'application php-mysql : vous entrez un pseudo et un message qui seront enregistrÃ©s dans la base de donnÃ©es sio table sisr

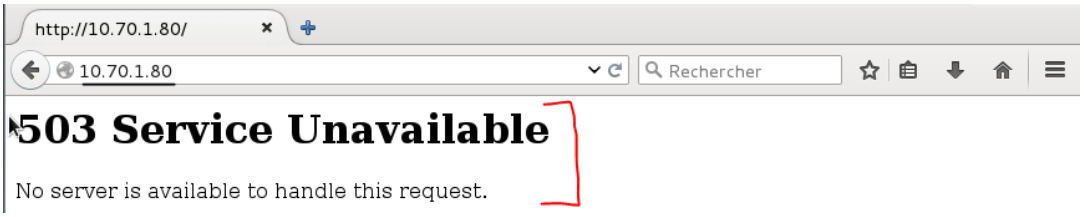
Pseudo :
Message :
Envoyer

**12.** Revenez à une répartition de charge égalitaire.

```
frontend proxypublic
        bind 10.70.1.63:80
        default_backend fermeweb

backend fermeweb
        balance roundrobin
        option httpclose
        option httpchk HEAD / HTTP/1.0
        server serwebsecu1 10.10.1.61:80 check
        server serwebsecu2 10.10.1.62:80 check
        stats refresh 30s
        stats auth admin:admin
        stats uri /stats
```

**13.** Vérifiez si le changement de serveur ne pose pas de problème lorsque l'utilisateur s'authentifie. Auquel cas, configurez HAProxy pour pouvoir utiliser l'application.

On se rend compte que l'on à une erreur 503



On a commenté la ligne avec option httpchk, on a aussi ajouter “cookie w1”/”cookie w2”, et “serwebsecu_mdp insert indirect”, cela permettra que si un client se connecte sur un des deux serveurs, sa session sera persistante

```
frontend proxypublic
        bind 10.70.1.61:80
        default_backend fermeweb

backend fermeweb
        balance roundrobin
        option httpclose
#       option httpchk HEAD / HTTP/1.0
        server serwebsecu1 10.10.1.61:80 cookie W1 check
        server serwebsecu2 10.10.1.62:80 cookie W2 check
        cookie serwebsecu_mdp insert indirect
```

**14.** Intégrer au serveur HAProxy un filtre applicatif via les « ACL ».

**15.** Donnez l’accès aux applications http et https

**16.** Ajouter un 3ème serveur HAProxyserweb3 à votre parc pour assurer la haute disponibilité, la redondance et la réplication des données.

**17.** Tester la haute disponibilité, avec des arrêts de serveurs.

**18.** Installer le serveur DNS sur le serveur HAProxy, et tester les accès web avec des noms FQDN.