MYDIGITALSCHOOL ANGERS — BTS SIO SISR

Mise en œuvre d'un Reverse Proxyavec Load Balancing (HAProxy)

Fichier de configuration commenté

Rédacteur	Paul Barrault-Gautier
Établissement	MyDigitalSchool Angers
Formation	BTS SIO — option SISR (Session 2026)
Cadre	Travaux pratiques / Projet de formation
Date	Avril 2026
Version	1.0

1. Introduction

Ce document présente le fichier de configuration HAProxy utilisé dans le cadre de la mise en œuvre du reverse proxy avec load balancing chez Loste Tradi-France. Chaque directive est commentée pour en expliquer le rôle et la valeur choisie.

La configuration implémente les fonctionnalités suivantes :

- Load balancing HTTP/HTTPS avec algorithme roundrobin
- Terminaison SSL/TLS avec redirection HTTP → HTTPS
- Health checks actifs sur les serveurs back-end
- Page de statistiques HAProxy sécurisée
- En-têtes de sécurité HTTP injectés par le proxy
- Logs structurés vers syslog

2. Section global

La section global définit les paramètres du processus HAProxy lui-même.

```
global
    # Fichier de log : envoie vers rsyslog local (facility local2)
    log /dev/log local2

    # Répertoire chroot : isolation du processus HAProxy
    chroot /var/lib/haproxy

    # Socket de statistiques (utilisé par socat et hatop)
    stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
    stats timeout 30s

    # Utilisateur et groupe dédiés (sécurité)
    user haproxy
    group haproxy

    # Mode daemon (service en arrière-plan)
    daemon

    # Nombre de threads (égal au nombre de CPU disponibles)
    nbthread 2

    # Fichier PID
    pidfile /var/run/haproxy.pid

    # Taille maximale de la liste des certificats SSL en cache
    tune.ssl.default-dh-param 2048

    # Activer le support SSL natif
    ssl-default-bind-ciphers
ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA3
84:ECDHE-RSA-AES256-GCM-SHA384
    ssl-default-bind-ciphersuites
TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
    ssl-default-bind-options prefer-client-ciphers no-sslv3 no-tlsv10 no-tlsv11
```

3. Section defaults

La section defaults définit les paramètres hérités par tous les frontends et backends, sauf surcharge explicite.

```
defaults
```

```
    # Mode : HTTP (niveau 7) — utiliser 'tcp' pour les flux non-HTTP
    mode http

    # Héritage du log global
    log global

    # Options HTTP recommandées
    option httplog           # Log au format HTTP (méthode, code, durée...)
    option dontlognull       # Ne pas logger les connexions sans requête (health
checks)
    option forwardfor        # Ajouter l'en-tête X-Forwarded-For (IP réelle du
client)
    option http-server-close # Fermer la connexion côté serveur après chaque requête
    option redispatch        # Si un back-end est DOWN, renvoyer vers un autre

    # Timeouts (valeurs adaptées à une appli web standard)
    timeout connect  5s      # Délai de connexion au back-end
    timeout client   50s     # Délai d'inactivité côté client
    timeout server   50s     # Délai d'inactivité côté serveur
    timeout http-request 10s # Délai max pour recevoir une requête HTTP complète
    timeout http-keep-alive 2s
    timeout queue    5s      # Délai max en file d'attente si tous les back-ends sont
occupés

    # Nombre de tentatives avant de marquer un back-end DOWN
    retries 3

    # Page d'erreur personnalisée (optionnel)
    # errorfile 503 /etc/haproxy/errors/503.http
```

4. Frontend HTTP (port 80)

Le frontend HTTP écoute sur le port 80 et redirige automatiquement tout le trafic vers HTTPS (301 permanent).

```
frontend fe_http
    # Écoute sur toutes les interfaces, port 80
    bind *:80

    # Mode HTTP (hérité de defaults, explicitement rappelé)
    mode http

    # Redirection permanente HTTP -> HTTPS
    # %[hdr(host)] : récupère le nom d'hôte de la requête originale
    # %[capture.req.uri] : préserve l'URI complète
    redirect scheme https code 301 if !{ ssl_fc }
```

5. Frontend HTTPS (port 443)

Le frontend HTTPS assure la terminaison SSL et le routage vers le backend applicatif.

```
frontend fe_https
    # Terminaison SSL sur le port 443
    # ssl : active la terminaison TLS
    # crt : chemin vers le fichier PEM (cert + clé)
    # alpn h2,http/1.1 : active HTTP/2
    bind *:443 ssl crt /etc/haproxy/certs/mondomaine.fr.pem alpn h2,http/1.1

    mode http
```

```
    # Injection des en-têtes de sécurité HTTP
    http-response set-header Strict-Transport-Security "max-age=63072000;
includeSubDomains; preload"
    http-response set-header X-Frame-Options "SAMEORIGIN"
    http-response set-header X-Content-Type-Options "nosniff"
    http-response set-header Referrer-Policy "strict-origin-when-cross-origin"

    # ACL : définition de règles de routage (exemple multi-application)
    acl host_app1 hdr(host) -i app1.mondomaine.fr
    acl host_app2 hdr(host) -i app2.mondomaine.fr

    # Routage selon le domaine
    use_backend be_app1 if host_app1
    use_backend be_app2 if host_app2

    # Backend par défaut
    default_backend be_app1
```

6. Backend applicatif

Le backend définit les serveurs cibles et la politique de répartition de charge.

```
backend be_app1
    mode http

    # Algorithme de load balancing
    # roundrobin : distribution circulaire (poids égaux par défaut)
    # Autres options : leastconn (moins de connexions), source (affinité IP)
    balance roundrobin

    # Health check HTTP actif
    # HAProxy envoie une requête GET /health toutes les 2 secondes
    option httpchk GET /health HTTP/1.1\r\nHost:\ app1.mondomaine.fr
    http-check expect status 200

    # Cookie de session (sticky session) - décommentez si nécessaire
    # cookie SERVERID insert indirect nocache

    # Serveurs back-end
    # check : active le health check
    # inter 2000 : fréquence du check (ms)
    # rise 2 : nb de checks OK pour repasser UP
    # fall 3 : nb de checks KO pour passer DOWN
    # weight 1 : poids (égal pour tous = roundrobin uniforme)
    server web-back-01 192.168.10.21:8080 check inter 2000 rise 2 fall 3 weight 1
    server web-back-02 192.168.10.22:8080 check inter 2000 rise 2 fall 3 weight 1
    server web-back-03 192.168.10.23:8080 check inter 2000 rise 2 fall 3 weight 1

    # Serveur de backup : utilisé uniquement si tous les autres sont DOWN
    # server web-backup 192.168.10.30:8080 backup
```

7. Page de statistiques HAProxy

```
listen stats
    # Écoute uniquement sur l'interface de management (pas sur WAN)
    bind 192.168.10.10:8404
```

```
    mode http
    stats enable

    # URI d'accès à la page de statistiques
    stats uri /haproxy-stats

    # Rafraîchissement automatique toutes les 5 secondes
    stats refresh 5s

    # Authentification HTTP Basic (remplacer par des valeurs sécurisées)
    stats auth admin:MotDePasseStrong!

    # Masquer la version HAProxy pour des raisons de sécurité
    stats hide-version

    # Afficher le nom de la page
    stats show-node
    stats show-legends
```

8. Résumé des directives clés

Directive	Section	Description
`balance roundrobin`	backend	Répartition circulaire équitable
`option forwardfor`	defaults / frontend	Transmet l'IP client via X-Forwarded-For
`option httpchk`	backend	Health check HTTP actif sur les back-ends
`redirect scheme https`	frontend fe_http	Redirection HTTP → HTTPS (301)
`bind *:443 ssl crt`	frontend fe_https	Terminaison SSL avec certificat PEM
`acl / use_backend`	frontend	Routage conditionnel multi-application
`server ... backup`	backend	Serveur de secours (failover)