목차

HAProxy 설정 예제

```
global
    log /dev/log    local0
    log /dev/log    local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    tune.ssl.default-dh-param  2048
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # Default ciphers to use on SSL-enabled listening sockets.
    # For more information, see ciphers(1SSL). This list is from:
    #  https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
    # An alternative list with additional directives can be obtained from
    #  https://mozilla.github.io/server-side-tls/ssl-config-generator/?server=haproxy
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-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 no-sslv3 no-tlsv10 no-tlsv11 no-tls-tickets

    ssl-default-server-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
    #ssl-default-server-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
    ssl-default-server-options no-sslv3 no-tlsv10 no-tlsv11 no-tls-tickets

    # curl https://ssl-config.mozilla.org/ffdhe2048.txt > /path/to/dhparam.pem
    ssl-dh-param-file /etc/haproxy/ssl/dhparam.pem

defaults
    log    global
    mode    http
    option    httplog
    option    dontlognull
    option    forwardfor
    #option    transparent
        timeout connect 5000
        timeout client  50000
        timeout server  50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend web
    bind *:80
        #bind *:443 ssl crt /etc/haproxy/ssl/nas.sample.net.pem crt
/etc/haproxy/ssl/company.com.pem crt /etc/haproxy/ssl/office.com.pem crt
/etc/haproxy/ssl/dev.sample.net.pem crt
/etc/haproxy/ssl/jenkins.sample.net.pem
   # 인증서를 하나의 파일로 묶어서 서비스 하는 방법
    # crt-list.txt 예제 참조
        bind *:443 ssl crt-list /etc/haproxy/ssl/crt-list.txt ca-file
/etc/haproxy/ssl/ca.pem verify optional alpn h2,http/1.1
        #reqadd X-Forwarded-Proto:\ https

        http-request set-header X-SSL %[ssl_fc]
    http-request set-header X-Forwarded-Port %[dst_port]
    http-request add-header X-Forwarded-Proto https if { ssl_fc }
    http-response set-header Cache-Control no-cache,\ max-age="600"

    ## CertBot Let's Encrypt
    # Test URI to see if its a letsencrypt request
    acl letsencrypt-acl path_beg /.well-known/acme-challenge/
    use_backend letsencrypt-backend if letsencrypt-acl

    ## Host Setting
        acl is_nas.sample.net           hdr(host) -i nas.sample.net
        acl is_plex.sample.net      hdr(host) -i plex.sample.net
    acl is_meet.sample.net        hdr(host) -i meet.sample.net
        acl is_mon.sample.net       hdr(host) -i mon.sample.net
        acl is_db.sample.net        hdr(host) -i db.sample.net
        acl is_m.sample.net     hdr(host) -i m.sample.net

     # 호스트명이 XXX 으로 끝나는 모든 요청
        acl is_mydomain.kr                  hdr_end(host) -i mydomain.kr
        acl is_sample.net                 hdr_end(host) -i sample.net
        acl is_fatp.org                 hdr_end(host) -i fatp.org

     # 요청을 리다이렉션
        #acl is_redirect_nas             path -i /
        #redirect code 301 location /webman/index.cgi if is_redirect_nas
```

```
is_nas.office.com
        #redirect code 301 location http://nas.office.com/webman/index.cgi
if is_redirect_nas
        #redirect prefix /webman/index.cgi code 301 if is_nas.office.com
is_redirect_nas

    ## Backend Setting
        use_backend backend_company.com         if is_company.com or
is_office.com
        use_backend backend_nas.sample.net      if is_nas.sample.net
        use_backend backend_plex.sample.net     if is_plex.sample.net
        use_backend backend_meet.sample.net     if is_meet.sample.net
        use_backend backend_mon.sample.net      if is_mon.sample.net
    use_backend backend_dev.sample.net        if is_dev.sample.net
    use_backend backend_m.sample.net        if is_m.sample.net
        use_backend backend_mydomain.kr         if is_mydomain.kr
        use_backend backend_m.second.domain     if is_m.second.domain
        use_backend backend_j.second.domain     if is_j.second.domain
        use_backend backend_k.second.domain     if is_k.second.domain
        use_backend backend_l.second.domain     if is_l.second.domain
        use_backend backend_www.sample.net      if is_sample.net

        #use_backend backend_https_registry.sample.net  if
is_registry.sample.net { ssl_fc }
        #use_backend backend_registry.sample.net    if
is_registry.sample.net
   # 기본 백엔드 설정
        default_backend backend_deny

    http-response set-header Strict-Transport-Security max-age=63072000

backend backend_nas.sample.net
    redirect scheme https code 301 if !{ ssl_fc }
        server  static  192.168.0.8:5000    check

backend backend_dev.sample.net
    #redirect scheme https code 301 if !{ ssl_fc }
        server  static  192.168.0.27:80     check

backend backend_company.com
    redirect scheme https code 301 if !{ ssl_fc }
        server  static  192.168.0.21:80     check

backend backend_www.sample.net
    redirect scheme https code 301 if !{ ssl_fc }
        server  static  192.168.0.30:80     check

backend backend_www.fatp.org
    redirect scheme https code 301 if !{ ssl_fc }
        server  static  192.168.0.31:80     check
```

```
backend backend_meet.sample.net
    redirect scheme https code 301 if !{ ssl_fc }
        server  static 192.168.0.28:443 check ssl verify none

backend backend_mon.sample.net
    redirect scheme https code 301 if !{ ssl_fc }
        server  static  192.168.0.15:80     check

backend backend_plex.sample.net
    redirect scheme https code 301 if !{ ssl_fc }
        server  static  192.168.0.26:32400  check

#backend backend_https_registry.sample.net
#   #server server1 backend:3000 weight 1 maxconn 8192 check ssl verify none
#        server  static 192.168.0.30:443    check ssl verify none

# m.second.domain
backend backend_m.second.domain
    redirect scheme https code 301 if !{ ssl_fc }
        server  static  192.168.0.38:80     check

# LE Backend
backend letsencrypt-backend
    server letsencrypt 127.0.0.1:8888

## deny backend
backend backend_deny
    http-request deny deny_status 400

#### MySQL # -> iptime 에서 direct forwarding 하므로 필요없음
#listen  db.sample.net
#        bind *:3306
#        mode tcp
        #timeout client  10800s
        #timeout server  10800s
#        balance leastconn
        #option httpchk
        #option allbackups
        #default-server port 9200 inter 2s downinter 5s rise 3 fall 2
slowstart 60s maxconn 64 maxqueue 128 weight 100
#        server db.sample.net 192.168.0.20:3306 check
```

통합인증서 파일 설정 crt-list.txt

```
root@proxy:/etc/haproxy/ssl# cat crt-list.txt
/etc/haproxy/ssl/domain2.localdomain.pem
/etc/haproxy/ssl/domain.localdomain.pem
```

```
# 통합 인증서파일 생성방법
cat /etc/letsencrypt/live/domain.localdomain/cert.pem
/etc/letsencrypt/live/domain.localdomain/privkey.pem
/etc/letsencrypt/live/domain.localdomain/chain.pem >
/etc/haproxy/ssl/domain.localdomain.pem
cat /etc/letsencrypt/live/domain2.localdomain/cert.pem
/etc/letsencrypt/live/domain2.localdomain/privkey.pem
/etc/letsencrypt/live/domain2.localdomain/chain.pem >
/etc/haproxy/ssl/domain2.localdomain.pem
```

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/haproxy_%EC%84%A4%EC%A0%95_%EC%98%88%EC%A0%9C

Last update: **2024/07/19 09:41**