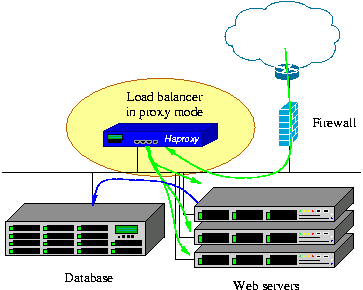
**

“HAproxy 는 http 어플리케이션과 TCP 통신에 있어 고가용성(High availability)과 부하분산 (load balancing)을 제공하는 프로그램으로 L4 스위치를 대체할 수 있다.

- linux22 for Linux 2.2  
— linux24 for Linux 2.4 and above (default)  
— linux24e for Linux 2.4 with support for a working epoll (> 0.21)  
— linux26 for Linux 2.6 and above  
— linux2628 for Linux 2.6.28, 3.x, and above (enables splice and tproxy) (2.6.32-431.e16.x86\_64)

1. 설치[centos 6버젼]

[root@localhost ~] # yum install openssl pcre-devel

[root@localhost ~] # cd /usr/local/src

[root@localhost ~] # wget <http://www.haproxy.org/download/1.6/src/haproxy-1.6.3.tar.gz>  
[root@localhost ~] # tar xzvf haproxy-1.6.3.tar.gz

[root@localhost ~] # cd haproxy-1.6.3

[root@localhost ~] make TARGET=linux2628 USE\_PCRE=1 USE\_OPENSSL=1 USE\_ZLIB=1

[root@localhost ~] # make install

install -d "/usr/local/sbin"

install haproxy "/usr/local/sbin"

install -d "/usr/local/share/man"/man1

install -m 644 doc/haproxy.1 "/usr/local/share/man"/man1

install -d "/usr/local/doc/haproxy"

for x in architecture close-options configuration cookie-options intro linux-syn-cookies lua management network-namespaces proxy-protocol; do \

install -m 644 doc/$x.txt "/usr/local/doc/haproxy" ; \

done

[root@localhost ~] # cp examples/haproxy.init /etc/init.d/haproxy

[root@localhost ~] cd /etc/init.d

[root@localhost ~] chmod 755 haproxy

[root@localhost ~] service haproxy

Usage: haproxy {start|stop|restart|reload|condrestart|status|check}

[root@localhost ~] ln –s /usr/local/sbin/haproxy /usr/sbin

[root@localhost ~] # cp examples/transparet\_proxy.cfg /etc/haproxy/haproxy.cfg

[root@localhost ~] service haproxy start

1. *설치[centos 7버젼]*

*yum install haproxy*

1. 설정 cat /etc/haproxy/haproxy.cfg

<https://cbonte.github.io/haproxy-dconv/configuration-1.6.html> 매뉴얼 참조

[global] : 전체적인 프로세스 관련된 부분을 설정

global

chroot /var/lib/haproxy

pidfile /var/run/haproxy.pid

maxconn 4000 --> 최대연결수

user haproxy

group haproxy

daemon --> 데몬형식으로 실행

[default] : global에 종속되는 섹션으로 proxt,mode,time out등을 설정

defaults

mode http --> tcp,http,health 3가지 모드가 있음

log global

option httplog

option dontlognull

option http-server-close

option forwardfor except 127.0.0.0/8

option redispatch

retries 3

timeout http-request 10s

timeout queue 1m

timeout connect 10s

timeout client 1m

timeout server 1m

timeout http-keep-alive 10s

timeout check 10s

maxconn 3000

[frontend] : client가 접속하는 front 설정

frontend f\_myapp\_ntlm --> frontend 섹션 이름설정

bind 192.168.17.184:80 --> 아파치 바인드설정과 같으며 해당 ip:port에서 listen

option http-keep-alive

default\_backend b\_myapp\_ntlm --> backend 서버 이름 지정

[backend] : front 접속 client가 실제 접속하는 real ip 설정

backend b\_myapp\_ntlm

option http-keep-alive

server s1 192.168.27.200:8888 check --> 실제 서비스할 real 서버의 ip:port parameter

* backend 서버가 2대 이상일 경우 balace 알고리즘 사용

balance roundrobin

server server1 ip1:port1

server server2 ip2:port2

[listen] : front+backend를 한번에 설정 --> frontend+backend와 listen중 하나만 설정

listen tcp\_service\_a :14900

mode tcp

option tcplog

option tcp-check

# tcp-check send PING

# tcp-check expect string PONG

balance roundrobin

server tcp\_service\_a\_01 xxx.xxx.xxx.001:4900 check

server tcp\_service\_a\_02 xxx.xxx.xxx.002:4900 check

listen stats :80

mode http

log global

maxconn 10

stats enable

stats refresh 30s

stats uri /haproxy.stats

1. 기동

haproxy -f /etc/haproxy/haproxy.cfg –c

Configuration file is valid

haproxy -f /etc/haproxy/haproxy.cfg (최초기동)

haproxy -f /etc/haproxy/haproxy.cfg -p /var/run/haproxy.pid -sf $(cat /var/run/haproxy.pid)(reload 설정)

netstat –anltp | grep haproxy

1. web통계