ALOHA LOAD BALANCER
REWRITING HTTP REQUESTS

"APPNOTE" #0007 — REWRITING HTTP REQUESTS

This application note is intended to help you apply rules for rewriting HTTP requests within the ALOHA Load Balancer solution.

CONSTRAINT

In order to rewrite requests, you need to understand regular expressions.

OBJECTIVE

Replace the "/jpg/" folder with "/images/", while maintaining the components before and after that folder.

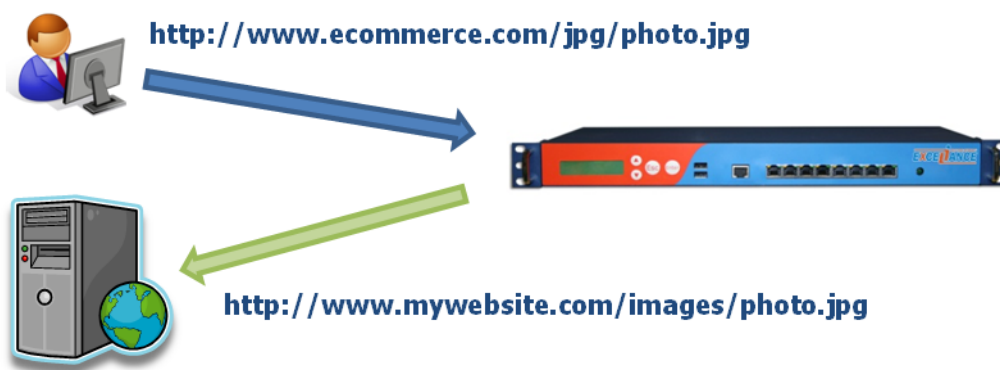Replace any host name in the http header with "www.mywebsite.com".

COMPLEXITY

★★☆☆☆

VERSIONS CONCERNED

V 3.x and later

TARGET NETWORK DIAGRAM

FUNCTIONS TO USE

In order to rewrite a request, use the "reqrep" and "reqirep" keywords with the following syntax:

```
reqrep  <search> <string> [{if | unless} <cond>]

reqirep <search> <string> [{if | unless} <cond>]    (ignored case)
```

<search> is the regular expression applied both to the HTTP headers and to the request. This is an extended regular expression. Grouped parentheses are supported, and the backslash character is not required. All spaces and known separators must be escaped using the backslash "\". The template is then applied to the entire line.

<string> is the entire line to be added. All spaces and known separators must be escaped using the backslash "\". You can refer to groups on corresponding patterns by using "\N", where "N" is an integer between 0 and 9.

<cond> is an optional corresponding condition produced from an ACL. Thus you can ignore this rule when the other conditions are not met.

Any line with a correspondence extended by a regular expression in the "search" argument of a request (in both the request and the header) will be completely replaced by the "string" argument. This is most commonly used to rewrite URLs or domain names in the "host" field of headers, for instance.

Important:

The "reqrep" keyword is strictly case-sensitive, while "repirep" is case insensitive.

The **<cond>** condition is available only from version v3.5.x and later.

EXTRACT OF THE LB LEVEL7 CONFIGURATION

```
######## The first public address as seen by the clients
frontend frt
 bind  10.0.32.10:80                 # address:port to listen to
 mode http
 log global                          # use global log parameters
 option httplog                      # Enable HTTP logging
 # Replace the "jpg" folder with the "images" folder
 reqrep ^([^\ ]*)\ /jpg/(.*)     \1\ /images/\2
 # Replace any host name in the header with "www.mysite.com"
  reqirep ^Host:\  Host:\ www.mysite.com
 maxconn 4000                        # max conn per instance
 timeout client 25s                  # maximum client idle time (ms)
 default_backend bck                 # send everything to this backend by default


####### This backend manages the servers and the load balancing algorithm
backend bck
 balance roundrobin                  # roundrobin | source | uri | leastconn
 mode http
 log global                          # use global log parameters
 option httplog                      # Enable HTTP logging
 cookie  SERVERID insert indirect nocache  # provide persistence with cookie
 option  httpchk HEAD /              # how to check those servers
 option forwardfor except 127.0.0.1/8      # add X-Forwarded-For except local
 fullconn 4000                       # dynamic limiting below
 timeout server 25s                  # max server's response time (ms)
 server srv1 10.0.32.101:80 cookie s1 weight 10 maxconn 100 check inter 1000 fall 3
 server srv2 10.0.32.102:80 cookie s2 weight 10 maxconn 100 check inter 1000 fall 3
```