

Stream Rewrite

Use Case:

This is example of simple reverse proxy task done using ISA server which is switching the requests from PROD to TEST setup. The same job can be done on load balancer using simple stream rewrite.

F5 iRules:

```
when HTTP_REQUEST {

    STREAM::disable

    # LTM does not uncompress response content, so if the server has compression enabled
    # and it cannot be disabled on the server, we can prevent the server from
    # sending a compressed response by removing the compression offerings from the client

    HTTP::header remove "Accept-Encoding"

    log local0. "Host: [HTTP::host]"
    if { [string tolower [HTTP::host]] equals "www.abc.com.int.xyz.com" } {
        HTTP::header replace "Host" "www.abc.com"
        pool POOL1
        log local0. "Setting Pool: POOL1"
    }

    when HTTP_RESPONSE {

        # Check if response type is text
        if {[HTTP::header value Content-Type] contains "text"}{

            # Replace http:// with https://

            STREAM::expression {@www.abc.com@www.abc.com.int.xyz.com@@www.def.com@www.def.com.int.xyz.com@}
```

```
# Enable the stream filter for this response only
STREAM::enable
}
}
```

URL: <https://devcentral.f5.com/questions/irule-to-redirect-traffic-as-well-as-changing-the-url>

NetScaler Solution:

```
add rewrite action replace_host_header replace
HTTP.REQ.HEADER("Host") "www.abc.com"
```

```
add rewrite policy policy_replace_host_header
HTTP.REQ.HEADER("Host").EQ("www.abc.com.int.xyz.com")
replace_host_header
```

Bind Rewrite policy to VSERVER or to Global rewrite bind point on Request Flow.

```
add rewrite action replace_http_response insert_after_all
HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH) "www.abc.com.int.xyz.com" -
pattern "www.abc.com"
```

```
add rewrite policy policy_replace_http_response
HTTP.RES.HEADER("Content-Type").CONTAINS("text")
replace_http_response
```

Bind Rewrite policy to specific VSERVER or to Global rewrite bind point on Response flow. Here we are replacing the Host header in the request before it goes to backend. On response path we are replacing the host header specific string in the HTTP body.