

Alternate solution to Sideband connection

Use Case:

Sideband functionality in F5's solution provides an ability to call an external resource before a load balancing decision has been made. In many cases, the information, required for a load balancing decision, doesn't change too fast, so instead of making synchronous call via Sideband for each incoming connection, it makes sense to collect that information from an external resource periodically, and then use it for incoming connections management. This solution reduces the load on external system while increasing runtime performance on the load balancer.

F5 iRules:

```
when RULE_INIT {
    log local0. "rule init"
    set static::StatusSubtableName "Sessions_list"
    set static::sql_debug 1
    set static::Site1Name "LA"
    set static::SQL_response "</feed>"
    set static::domain_Netbios "BDD"
    set static::IIS_Timeout_Entry 60
}

when HTTP_REQUEST {
    set log_prefix "[IP::remote_addr]:[TCP::remote_port clientside] [IP::local_addr]:[TCP::local_port clientside]"
    if {$static::sql_debug > 1} { log local0. "<$log_prefix>: HTTP Request : [HTTP::uri]"}
}

when HTTP_RESPONSE {
    if {$static::sql_debug > 1} { log local0. "<$log_prefix>: HTTP response: [HTTP::status]"}
    HTTP::collect [HTTP::header Content-Length]
}
```

```

when HTTP_RESPONSE_DATA {
    if {$static::sql_debug > 1} { log local0. "<$log_prefix>: HTTP response data: [HTTP::payload]"}

    if {[HTTP::payload] contains $static::SQL_response } {
        if { not ( [findstr [HTTP::payload] "ModuleAndEventText>"] eq "" ) } {

            set user_array [split [HTTP::payload] "\n"]

            foreach line $user_array {
                if {$static::sql_debug > 1} { log local0. "<$log_prefix>: line: $line"}

                set user_string [findstr $line $static::domain_Netbios\ \ 0 " :ModuleAndEventText>"]

                if {$static::sql_debug > 1} { log local0. "<$log_prefix>: user_string: $user_string"}

                if { not ($user_string eq "") } {

                    set user_string [string tolower [findstr $user_string $static::domain_Netbios 0 " "]]

                    if {$static::sql_debug > 0} { log local0. "<$log_prefix>: Adding to the table: $user_string $static::Site1Name" }

                    table set -subtable $static::StatusSubtableName $user_string $static::Site1Name $static::IIS_Timeout_Entry

                }

            }

        }

        HTTP::respond 200 "OK"

    } else {

        log local0. "<$log_prefix>: Didn't get response from IIS Server"

        HTTP::respond 500 "ERROR"

    }

}

```

URL: <https://devcentral.f5.com/codeshare/alternative-solution-to-sideband-connection>

NetScaler Solution:

```
add policy httpCallout hc1 -IPAddress 10.102.58.183 -port 80 -
returnType TEXT -hostExpr HTTP.REQ.HOSTNAME -urlStemExpr
"/testsite/get_user_data/" -headers
UserName(HTTP.REQ.HEADER("Authorization").AFTER_STR("BASIC").B64DECO
DE.BEFORE_STR(":")) Domain(HTTP.REQ.HOSTNAME) -resultExpr
"HTTP.RES.BODY(1000)" -cacheForSecs 500
```

The simplified NetScaler solution here lets you create a callout to reach out to an external system to collect required data needed for making load balancing decision. Within the Callout configuration you can specify how long the response should be cached on NetScaler so that we are not reaching out to external server for every request. All this configuration is done in single Callout command and it can then be invoked using a policy bound to the respective vserver.