FortiADC - SLB Script Deployment Guide

Version 5.2.x

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO GUIDE
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/support-and-training/training.html

NSE INSTITUTE
https://training.fortinet.com

FORTIGUARD CENTER
https://fortiguard.com/

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

July 17, 2019
FortiADC 5.2.x SLB Script Deployment Guide

TABLE OF CONTENTS

Change Log

Date	Change Description
2019-07-17	Second release.
2019-01-29	Initial release.

Introduction

FortiADC SLB supports Lua scripts to perform actions that are not currently supported by the built-in feature set. Scripts enable you to use predefined script commands and variables to manipulate the HTTP request/response or select a content route. The multi-script support feature enables you to use multiple scripts by setting their sequence of execution.

Here are FortiADC's predefined scripts and commands that you can copy and customize in the GUI/Server Load Balance/Scripting page.

Configuration Overview

The script used in the SLB/VS configuration is triggered when the associated virtual server receives an HTTP request or response. Then, it does the programmed action. The events in which you can create them are shown as below:

Event name	Description
RULE_INIT	The event is used to initialize global or static variables used within a script. It is triggered when a script is added or modified, or when the device starts up, or when the software is restarted.
VS_LISTENER_BIND	When a VS tries to bind.
TCP_ACCEPTED	When a TCP connection from a client is accepted
CLIENTSSL_HANDSHAKE	When a client-side SSL handshake is completed.
HTTP_REQUEST	The virtual server receives a complete HTTP request header.
HTTP_DATA_REQUEST	When an HTTP:collect command finishes processing on the server side of a connection.
SERVER_BEFORE_ CONNECT	When we are going to connect to the backend real server
SERVERSSL_HANDSHAKE	When a server-side SSL handshake is completed.
SERVER_CONNECTED	When Httproxy deem that the backend real server is connected
HTTP_RESPONSE	The virtual server receives a complete HTTP response header.
HTTP_DATA_RESPONSE	When an HTTP:collect command finishes processing on the server side of a connection.
SERVER_CLOSED	When Httproxy is going to terminate the backend real server connection
TCP_CLOSED	When a TCP connection from a client is to be closed
CLIENTSSL_RENEGOTIATE	When a client-side SSL renegotiation is completed.
SERVERSSL_RENEGOTIATE	When a server-side SSL renegotiation is completed.
AUTH_RESULT	When authentication(HTML Form / HTTP-basic) is done
COOKIE_BAKE	When FADC is done baking an authentication cookie

The examples of built-in predefined scripts are as follows:

Predefined script	Description
AES_DIGEST_SIGN_2F_ COMMANDS	Demonstrate how to use AES to encryption/decryption data and some tools to generate the digest.
AUTH_COOKIE_BAKE	Allows you to retrieve the baked cookie and edit the cookie content.

Predefined script	Description
AUTH_EVENTS_n_ COMMANDS	Used to get the information from authentication process.
CLASS_SEARCH_n_MATCH	Demonstrates how to use the class_match and class_search utility function.
COMPARE_IP_ADDR_2_ ADDR_GROUP_DEMO	Compares an IP address to an address group to determine if the IP address is included in the specified IP group. For example ,192.168.1.2 is included 192.168.1.0/24. Note: Do NOT use this script "as is". Instead, copy it and customize the IP address and the IP address group.
CONTENT_ROUTING_by_ URI	Routes to a pool member based on URI string matches. You should not use this script as is. Instead, copy it and customize the URI string matches and pool member names.
CONTENT_ROUTING_by_X_ FORWARDED_FOR	Routes to a pool member based on IP address in the X-Forwarded-For header. You should not use this script as is. Instead, copy it and customize the X-Fowarded-For header values and pool member names.
COOKIE_COMMANDS	Demonstrate the cookie command to get the whole cookie in a table and how to remove/insert/set the cookie attribute.
COOKIE_COMMANDS_ USAGE	Demonstrate the sub-function to handle the cookie attribute "SameSite" and others.
COOKIE_CRYPTO_ COMMANDS	Used to perform cookie encryption/decryption on behalf of the real server.
CUSTOMIZE_AUTH_KEY	Demonstrate how to customize the crypto key for authentication cookie.
GENERAL_REDIRECT_ DEMO	Redirects requests to a URL with user-defined code and cookie. Note: Do NOT use this script "as is". Instead, copy and customize the code, URL, and cookie.
GEOIP_UTILITY	Used to fetch the GEO information country and possible province name of an IP address.
HTTP_2_HTTPS_ REDIRECTION	Redirects requests to the HTTPS site. You can use this script without changes
HTTP_2_HTTPS_ REDIRECTION_FULL_URL	Redirects requests to the specified HTTPS URL. Note: This script can be used directly, without making any change.
HTTP_DATA_FETCH_SET_ DEMO	Collects data in HTTP request body or HTTP response body. In HTTP_ REQUEST or HTTP_RESPONSE, you could collect specified size data with “size” in collect().In HTTP_DATA_REQUEST or HTTP_DATA_RESPONSE. You could print the data use “content”, calculate data length with “size”, and rewrite the data with “set”.

Predefined script	Description
	Note: Do NOT use this script "as is". Instead, copy it and manipulate the collected data.
HTTP_DATA_FIND_ REMOVE_REPLACE_DEMO	Finds a specified string, removes a specified string, or replaces a specified string to new content in HTTP data. Note: Do NOT use this script "as is". Instead, copy it and manipulate the collected data.
INSERT_RANDOM_ MESSAGE_ID_DEMO	Inserts a 32-bit hex string into the HTTP header with a parameter “Message-ID”. Note: You can use the script directly, without making any change.
IP_COMMANDS	Used to get various types IP Address and port number between client and server side.
MANAGEMENT_ COMMANDS	Allow you to disable/enable rest of the events from executing.
MULTIPLE_SCRIPT_ CONTROL_DEMO_1	Uses demo_1 and demo_2 script to show how multiple scripts work. Demo_1 with priority 12 has a higher priority. Note: You could enable or disable other events. Do NOT use this script "as is". Instead, copy it and customize the operation.
MULTIPLE_SCRIPT_ CONTROL_DEMO_2	Uses demo_1 and demo_2 script to show how multiple scripts work. Demo_2 with priority 24 has a lower priority. Note: You could enable or disable other events. Do NOT use this script "as is". Instead, copy it and customize the operation
OPTIONAL_CLIENT_ AUTHENTICATION	Performs optional client authentication. Note: Before using this script, you must have the following four parameters configured in the client-ssl-profile: l client-certificate-verify—Set to the verify you'd like to use to verify the client certificate. l client-certificate-verify-option—Set to optional l ssl-session-cache-flag—Disable. l use-tls-tickets—Disable. l
REDIRECTION_by_STATUS_ CODE	Redirects requests based on the status code of server HTTP response (for example, a redirect to the mobile version of a site). Do NOT use this script "as is". Instead, copy it and customize the condition in the server HTTP response status code and the URL values.

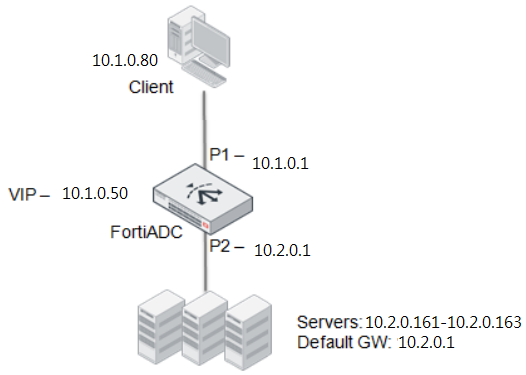
Predefined script	Description
REDIRECTION_by_USER_AGENT	Redirects requests based on User Agent (for example, a redirect to the mobile version of a site). You should not use this script as is. Instead, copy it and customize the User Agent and URL values
REWRITE_HOST_n_PATH	Rewrites the host and path in the HTTP request, for example, if the site is reorganized. You should not use this script as is. Instead, copy
REWRITE_HTTP_2_HTTPS_in_LOCATION	Rewrites HTTP location to HTTPS, for example, rewrite “Location:http://www.example.com” to “Location:https://www.example.com” Note: You can use the script directly, without making any change
REWRITE_HTTP_2_HTTPS_in_REFERER	Rewrites HTTP referer to HTTPS, for example, rewrite “Referer: http://www.example.com” to “Referer: https://www.example.com”. Note: You can use the script directly, without making any change.
REWRITE_HTTPS_2_HTTP_in_LOCATION	Rewrites HTTPS location to HTTP, for example, rewrite “Location:https://www.example.com” to “Location:http://www.example.com”. Note: You can use the script directly, without making any change.
REWRITE_HTTPS_2_HTTP_in_REFERER	Rewrites HTTPS referer to HTTP, for example, rewrite “Referer: https://www.example.com” to “Referer: http://www.example.com”. Note: You can use the script directly, without making any change
SNAT_COMMANDS	Allows you to overwrite client source address to a specific IP for certain clients, also support IPv4toIPv6 or IPv6toIPv4 type. Note: Make sure the flag SOURCE ADDRESS is selected in the HTTP or HTTPS type of profile.
SOCKOPT_COMMAND_USAGE	Allows user to customize the TCP_send buffer and TCP_receive buffer size.
SPECIAL_CHARACTERS_HANDLING_DEMO	Shows how to use those "magic characters" which have special meanings when used in a certain pattern. The magic characters are ( ) . % + - * ? [ ] ^ $
SSL_EVENTS_n_COMMANDS	Demonstrate how to fetch the SSL certificate information and some of the SSL connection parameters between server and client side.
TCP_EVENTS_n_COMMANDS	Demonstrate how to reject a TCP connection from a client in TCP_ACCEPTED event.
URL_UTILITY_COMMANDS	Demonstrate how to use those url tools to encode/decode/parser/compare .

Predefined script	Description
USE_REQUEST_HEADERS_ in_OTHER_EVENTS	Stores a request header value in an event and uses it in other events. For example, you can store a URL in a request event, and use it in a response event. Note: Do NOT use this script "as is". Instead, copy it and customize the content you want to store, use collect() in HTTP_REQUEST to trigger HTTP_DATA_REQUEST,or use collect() in HTTP_ RESPONSE to trigger HTTP_DATA_ RESPONSE.
UTILITY_FUNCTIONS_ DEMO	Demonstrates how to use the basic string operations and random number/alphabet, time, MD5, SHA1, SHA2, BASE64, BASE32, table to string conversion, network to host conversion utility function.

Content routes based on a URI string

The content routing feature has rules that match HTTP requests to content routes based on a Boolean AND combination of match conditions. If you want to select routes based on a Boolean OR, you can configure multiple rules. The content routing rules table is consulted from top to bottom until one matches.

Topology



Create a script object

1. Go to Server Load Balance > Scripting
2. Click **Create New** to display the configuration editor
3. Complete the configuration as below:

```
when HTTP_REQUEST{
uri = HTTP:uri_get()
if uri:find("news") then
LB:routing("SP1")
debug("uri %s \n", uri);
elseif uri:find("finance") then
LB:routing("SP2")
debug("uri %s \n", uri);
elseif uri:find("game") then
LB:routing("SP3")
debug("uri %s \n", uri);
end
}
```

4. **Save** the configuration.

Create a content route rule

1. Go to Server Load Balance > Virtual Server.
2. Click the Content Routing tab.
3. Click Create New to display the configuration editor.
4. Complete the configuration as described below:

Name	Type	Real Server	Match Condition Count	
SP1	Layer 7	Root-RS1-HTTP	0	
SP2	Layer 7	Root-RS2-HTTP	0	
SP3	Layer 7	Root-RS3-HTTP	0	

5. **Save** the configuration.

Liking the script to the virtual server

1. Go to Server Load Balance > Virtual Server

2. Click one of the VS to display the configuration windows.

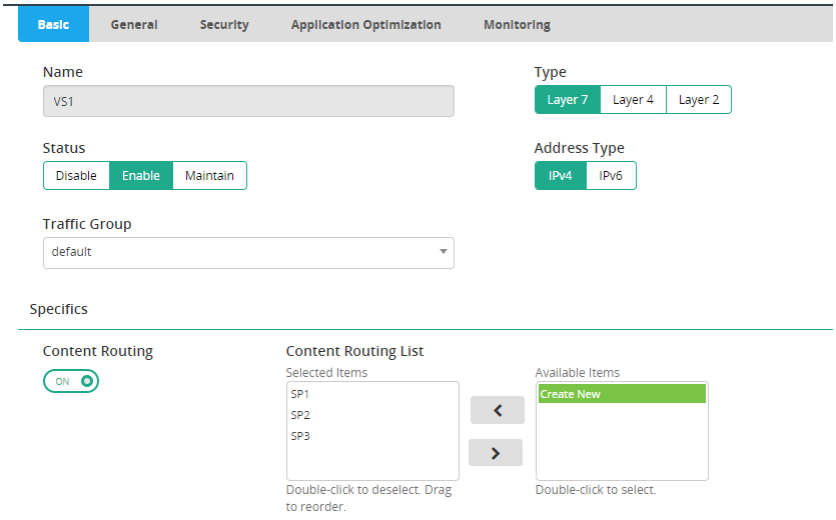
3. **Enable** content routing and select the content route configuration objects in the tab “Basic.”

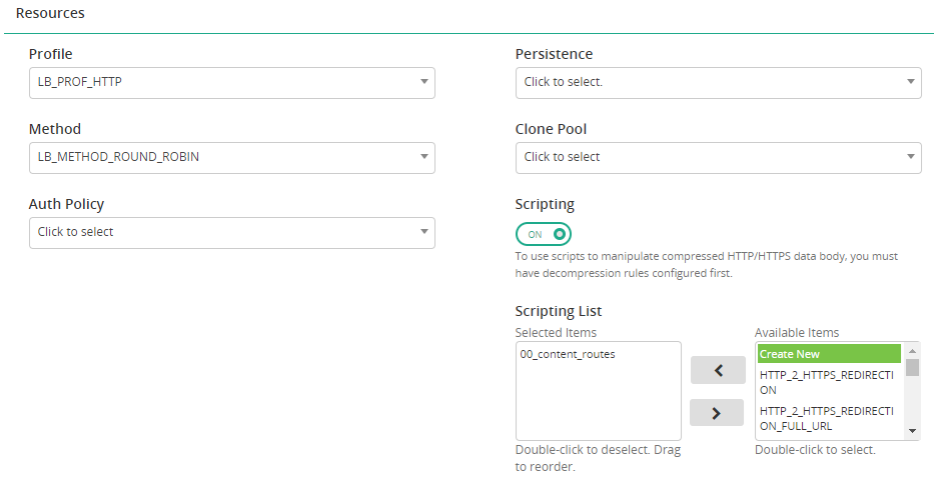
3. Click the tab “General.”

4. Tap the Scripting toggle on.

5. In Scripting List, select “00_content_routes” from the Available Items and move it to the Selected Items column.

6. Click **Save** to save the configuration.





Confirm that the log printed in the console and routing works well

1. Connect your management computer to the FortiADC

2. Enable the diagnose debug output for httproxy_script:

```
diagnose debug module httproxy scripting set
diagnose debug enable
```

3. Send a HTTP request(http://10.1.0.50/news) to VS from client and you will see the "uri /news" printed on the screen and see the content of the RS1.

4. Send a HTTP request(http://10.1.0.50/finance) to VS from client and you will see the "uri /finance" printed on the screen and see the content of the RS2.

5. Send a HTTP request(http://10.1.0.50/game) to VS from client and you will see the "uri /game" printed on the screen and see the content of the RS3.

```
(M) VM02 (root) #
(M) VM02 (root) # diagnose debug module httproxy scripting

(M) VM02 (root) # diagnose debug enable

(M) VM02 (root) #
(M) VM02 (root) # uri /news

(M) VM02 (root) # uri /finance

(M) VM02 (root) # uri /game

(M) VM02 (root) #
```

```
[root@Client1 ~]#
[root@Client1 ~]# curl http://10.1.0.50/news
SERVER1    NEWS     SERVER1
SERVER1    NEWS     SERVER1
SERVER1    NEWS     SERVER1
SERVER1    NEWS     SERVER1
SERVER1    NEWS     SERVER1
SERVER1    NEWS     SERVER1
SERVER1    NEWS     SERVER1
[root@Client1 ~]# curl http://10.1.0.50/finance
SERVER2    Finance    SERVER2
SERVER2    Finance    SERVER2
SERVER2    Finance    SERVER2
SERVER2    Finance    SERVER2
SERVER2    Finance    SERVER2
[root@Client1 ~]# curl http://10.1.0.50/game
SERVER3    game   SERVER3
SERVER3    game   SERVER3
SERVER3    game   SERVER3
SERVER3    game   SERVER3
SERVER3    game   SERVER3
SERVER3    game   SERVER3
```