AVI Networks®

Out of Band Request Processing via DataScripts

Avi Technical Reference (v20.1)

Out of Band Request Processing via DataScripts

view online

Overview

When a virtual service receives a request, either the whole request, or parts of the request, or an entirely different request can be sent to a third-party end point to validate for a specific criteria. The action on the original request is based on the response thus received. Starting with Avi Vantage version 20.1.3, the processing of certain incoming client requests can be paused to retrieve information from an external resource through DataScripts.

An external resource can be either a single server (that may belong to a third-party entity) or a pool of servers. The servers are accessed either by an IP address or an FQDN.

Note: The out-of-band request processing feature is currently under tech preview.

This article explains the DataScript `avi.requests.xxx`, which enables:

- Pausing the incoming client requests
- Configuring a new HTTP request
- Sending the request to a third party pool
- Waiting for a response from a third-party pool
- Resuming the initial client request, or closing it, or reposing with a local redirect/response

Note: Ensure that the third party or the external entity is configured as a pool in Avi Vantage.

Some use cases for external request validation are: * To query to an LDAP to fetch attributes and insert them as headers to a backend server * Converting token names into user names from a third party server before sending the information to the server * Real-time look up to database instead of using a static database to block users from specific countries

## DataScript

Function	avi.requests.get/post/put/head
Description	Send an external HTTP request to an external (third-party) server within the DataScript, whenever an incoming c Allows to pause a client request and configuring a custom HTTP request (or forward the original client request), s then use that response to make load balancing decisions for the initial client request when sending it to the backe
Functions	The HTTP method will be the function name instead of a parameter as follows: • `avi.requests.get(pool [, server [, port]], URI [, req_args])` • `avi.requests.post(pool [, server [,port]], URI [, req_args])` • `avi.requests.head(pool [, server [,port]], URI [, req_args])` • `avi.requests.put(pool [, server [,port]], URI [, req_args])`

Events

- http_auth
- http_post_auth
- http_req
- http_req_body

Parameters

- Pool: Name of external pool to select (<String>)
- Server: Name of external server optional
- Port: Port of external server optional
- Method: HTTP method name in a string. For example, get, post, head, put, patch, and delete optional
- URI: URI of the external request (Path + query strings)(<String>)
- req_args: Additional optional request configurations optional

```
<b>Headers</b>: Table of request headers to send. The values must be either a number or string.(&
Avi will insert in headers that are required for HTTP, but aren?t provided like Host, Content-Leng
<b>Body</b>: Request body, if this the POST request (&lt;String&gt;)<br>
```

If the method is POST but the body is not provided (or body is nil), then by default, the client?s request b
otherwise a [500] error is displayed)

Value Returned

The table contains the following values:

- Status: Response Status Code (<Number>)
- Headers: Table of response headers (<Table>)
- Body: HTTP Response body if present, else nil (<String>)

Configuration

To use the external request validation API, configure a new Pool object with the external server(s) where you wa

```
[admin:controller]: > configure vsdatascriptset vsds-external-request
 [admin:controller]: vsdatascriptset> pool_refs pool-1
 [admin:controller]: vsdatascriptset> save
```

Note: To send HTTPS requests, configure the pool with an SSL profile and ensure that the server port has an SSL

Examples

```
headers = {
                Host= "example.com:80",
                Accept= "*/",
                ["Content-Type"] = "text/html",
                ["Content-Length"] = "11"
            }
 req_args = {headers = headers, body = "example body"}
 response = avi.requests.get('pool-1', 'example.com', '/', req_args)
 --[[
   response = {status, headers, body}
 --]]
```

Owing to the optional arguments, there are multiple ways to call the *avi.requests.xxxx* API:

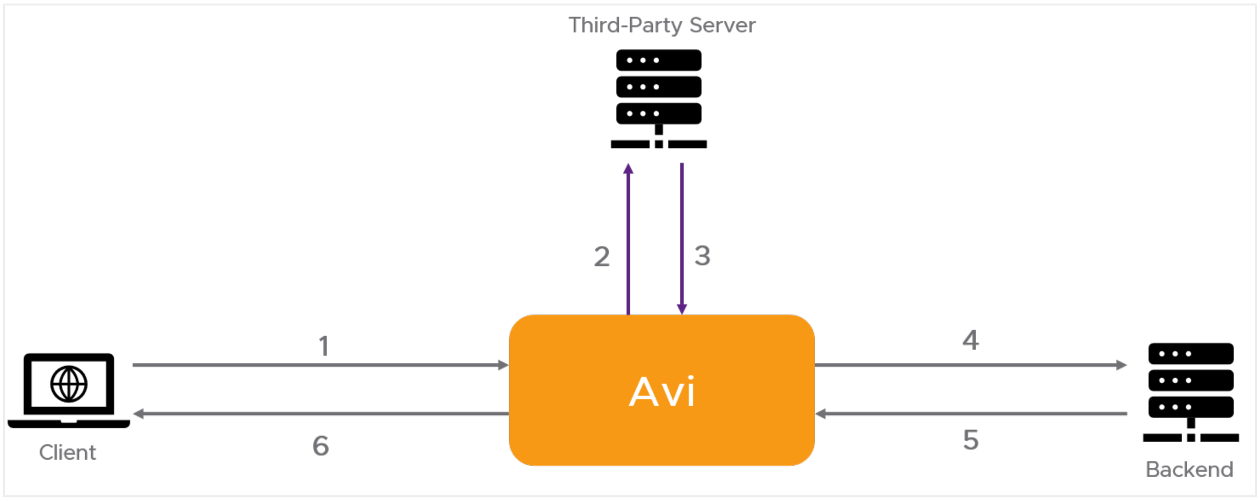
```
avi.requests.get('pool', '/index.html')
 avi.requests.get('pool', 'server', '/index.html')
 avi.requests.get('pool', 'server', 80, '/index.html')
 avi.requests.get('pool', /index_html', req_args)
 avi.requests.get('pool', 'server', '/index_html', req_args)
 avi.requests.get('pool', 'server', 80, '/index.html', req_args)
```

Steps to Configure Out-of-Band Request Processing

1. Configure the Pool. The external application can be defined/referred to in multiple ways. It could be a single server or a pool of servers. To use the external pool in the DataScript, attach the pool to the `VSDataScriptSet` object.
2. Write a DataScript using the function described here.
3. Associate the pool with the DataScript
4. Bind the DataScript to the virtual service for which out-of-band processing is required.

Packet Flow in an Out of Band Request Processing Scenario

Let us consider the packet flow for a client request resume scenario.



1. The client sends an incoming request which triggers the DataScript condition to send an out-of-band request
2. The DataScript creates a HTTP request and sends it to the configured external server
3. The server provides a HTTP response, which will be evaluated by the DataScript
4. After processing the initial client request based on the external server response, the client request is forwarded to the backend
5. Avi receives the response from the backend servers
6. Avi forwards the backend server response to the client

Caveats

- Sending multiple out-of-band requests simultaneously is not supported
- Asynchronous out-of-band requests are not supported
- Sending chunked requests is not supported
- Only HTTP/1.1 requests can be sent

Document Revision History

```
<th>Date</th>
<th>Change Summary</th>
```

December 07, 2020	Published the article for Out-of-band Request Processing(Version 20.1.3)
March 11, 2021	Edited Example in Datascript section