You can find the most up-to-date technical documentation on the VMware Web site at: 
http://www.vmware.com/support/
The VMware Web site also provides the latest product updates.
If you have comments about this documentation, submit your feedback to:
docfeedback@vmware.com
Contents

1 About vRealize Log Insight Developer’s Guide  5
   Using the vRealize Log Insight Ingestion API  5

Index  9
About vRealize Log Insight Developer's Guide

The VMware vRealize Log Insight Developer's Guide provides information about the vRealize Log Insight Ingestion API.

Intended Audience

This information is intended for anyone who wants to use the vRealize Log Insight Ingestion API. You must be familiar with REST concepts and with the JSON serialization format.

Using the vRealize Log Insight Ingestion API

You can interact with the vRealize Log Insight Ingestion API to send events to the vRealize Log Insight server.

All API request and response bodies are UTF8 encoded JSON strings with Content-Type: application/json header field. On success, all calls return HTTP response code 200.

Enforce SSL Only Connections

You can use the vRealize Log Insight Web UI to configure the Log Insight Agents and the Ingestion API to allow only SSL connections to the server.

NOTE The current version of vRealize Log Insight doesn’t support syslog SSL connections and works only for vRealize Log Insight Ingestion API.

Prerequisites

Verify that you are logged in to the vRealize Log Insight Web user interface as a user with the Edit Admin permission. The URL format is https://log-insight-host, where log-insight-host is the IP address or host name of the vRealize Log Insight virtual appliance.

Procedure

1  Click the configuration drop-down menu icon and select Administration.
2  Under Configuration, click SSL.
3  Under the API Server SSL, select the Require SSL Connection check box.
4  Click Save.

vRealize Log Insight API allows only SSL connections to the server. Non-SSL connections are refused.
Using the messages/ingest Service

You can use the messages/ingest service to send events to a vRealize Log Insight server using HTTP POST requests.

The messages/ingest service uses the following syntax.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>http://loginsight_host:9000/api/v1/messages/ingest/agentId</td>
</tr>
<tr>
<td>HTTPS</td>
<td>https://loginsight_host:9543/api/v1/messages/ingest/agentId</td>
</tr>
</tbody>
</table>

If you enforce SSL from the Web UI you will be able to use only HTTPS. See “Enforce SSL Only Connections,” on page 5.

HTTP Method

POST

Note The vRealize Log Insight Ingestion API has a limit of 100 KB per HTTP POST request. The maximum size of a single text field is 16 KB.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Where to pass</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agentId</td>
<td>String</td>
<td>In URL</td>
<td>The ID of the sending agent should follow the UUID standard. The agent may be an official vRealize Log Insight Windows or Linux agent or any client leveraging the Ingestion API.</td>
</tr>
<tr>
<td>Content-Type:</td>
<td>String</td>
<td>In POST body</td>
<td>The Content-Type parameter specifies the nature of the data in the POST body.</td>
</tr>
<tr>
<td>application/json</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Events array</td>
<td>Array</td>
<td>In POST body</td>
<td>An array of events. Each event must have the following format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>{&quot;messages&quot;:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[{</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;text&quot;: optional, message text as a string,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;timestamp&quot;: optional, timestamp encoded as number of milliseconds since Unix epoch in UTC,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;fields&quot;: optional array of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[{</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;name&quot;: the name of the field,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;content&quot;: optional, the content of the field,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;startPosition&quot;: optional, the start position in the &quot;text&quot;,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;length&quot;: optional, the length of the string in the &quot;text&quot;,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>},....]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>}},....]</td>
</tr>
</tbody>
</table>

Note The vRealize Log Insight server compares the "timestamp" you provide with the local time on the vRealize Log Insight server. If you provide a "timestamp" outside of the default 10 minutes tolerated drift window, the vRealize Log Insight server ignores your "timestamp" and uses its local time. If "timestamp" is not present, vRealize Log Insight uses arrival time.

Note If the "content" of a field is not present, then "startPosition" and "length" must be present and must point to a valid position in the "text" field string.
Return HTTP Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 OK</td>
<td>Integer</td>
<td>Standard HTTP response codes</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>500 Internal Server</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>503 Service Unavailable</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

Example Request

POST http://loginsight:9000/API/v1/messages/ingest/4C4C4544-0037-5910-805A-C4C04F585831

Host: loginsight:9000
Connection: keep-alive
Content-Type: application/json
charset: utf-8
Content-Length: ??

{"messages": [{
    "fields": [
        {"name": "Channel", "content": "Security"},
        {"name": "EventID", "content": "4688"},
        {"name": "EventRecordID", "content": "33311266"},
        {"name": "Keywords", "content": "Audit Success"},
        {"name": "Level", "content": "Information"},
        {"name": "OpCode", "content": "Info"},
        {"name": "ProcessID", "content": "4"},
        {"name": "ProviderName", "content": "Microsoft-Windows-Security-Auditing"},
        {"name": "Task", "content": "Process Creation"},
        {"name": "ThreadID", "content": "64"}
    ],
    "text": "A new process has been created.",
    "timestamp": 1396622879241
}]

Example Response

HTTP/1.1 200 OK

{"status":"ok","message":"messages ingested","ingested":18}
Index

A
api, use  5
API, messages/ingest service  6

E
enforce SSL connection  5

I
intended audience  5