53-1005043-01
22 May 2017

Brocade Virtual Traffic Manager: REST API Guide

Supporting 17.2

BROCADE

Brocade Communications Systems, Incorporated

Corporate and Latin American Headquarters
Brocade Communications Systems, Inc.
130 Holger Way
San Jose, CA 95134
Tel: 1-408-333-8000
Fax: 1-408-333-8101
E-mail: info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems China HK, Ltd.
No. 1 Guanghua Road
Chao Yang District
Units 2718 and 2818
Beijing 100020, China
Tel: +8610 6588 8888
Fax: +8610 6588 9999
E-mail: china-info@brocade.com

European Headquarters
Brocade Communications Switzerland Sàrl
Centre Swissair
Tour B - 4ème étage
29, Route de l'Aéroport
Case Postale 105
CH-1215 Genève 15
Switzerland
Tel: +41 22 799 5640
Fax: +41 22 799 5641
E-mail: emea-info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems Co., Ltd. (Shenzhen WFOE)
Citic Plaza
No. 233 Tian He Road North
Unit 1308 – 13th Floor
Guangzhou, China
Tel: +8620 3891 2000
Fax: +8620 3891 2111
E-mail: china-info@brocade.com

Contents

Preface

Read this preface for an overview of the information provided in this guide. This preface includes the following sections:

Document Conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Notes and Warnings

Note, important, and caution statements might be used in this document. They are listed in the order of increasing severity of potential hazards.

Note: A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

Important: An Important statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

Caution: A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

Text Formatting Conventions

Text formatting conventions such as boldface, italic, or Courier font may be used in the flow of the text to highlight specific words or phrases.

Format	Description
bold text	Identifies command names
	Identifies keywords and operands
	Identifies the names of user-manipulated GUI elements
	Identifies text to enter at the GUI
italic text	Identifies emphasis
	Identifies variables
	Identifies document titles
`Courier font`	Identifies CLI output
	Identifies command syntax examples

Command Syntax Conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text. For example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ x | y | z }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options. In Fibre Channel products, square brackets may be used instead for this purpose.
x | y	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, member[member...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Brocade Resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

White papers, data sheets, and the most recent versions of Brocade software and hardware manuals are available at www.brocade.com. Product documentation for all supported releases is available to registered users at MyBrocade. Click the **Support** tab and select **Document Library** to access documentation on MyBrocade or www.brocade.com. You can locate documentation by product or by operating system.

Release notes are bundled with software downloads on MyBrocade. Links to software downloads are available on the MyBrocade landing page and in the Document Library.

Document Feedback

Quality is our first concern at Brocade and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com.
- By sending your feedback to documentation@brocade.com.

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers contact their OEM/Solutions provider.

Brocade Customers

For product support information and the latest information on contacting the Technical Assistance Center, go to www.brocade.com and select **Support**.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone	E-mail
Preferred method of contact for nonurgent issues: • Case management through the MyBrocade portal. • Quick Access links to Knowledge Base, Community, Document Library, Software Downloads and Licensing tools.	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • Toll-free numbers are available in many countries. • For areas unable to access a toll free number: +1-408-333-6061	support@brocade.com Please include: • Problem summary • Serial number • Installation details • Environment description

Brocade OEM Customers

If you have purchased Brocade product support from a Brocade OEM/Solution Provider, contact your OEM/Solution Provider for all of your product support needs.

- OEM/Solution Providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/Solution Provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/Solution Provider.

CHAPTER 1 Introduction to the REST API

This chapter provides an overview of the Brocade Virtual Traffic Manager (Traffic Manager) REST API. This chapter contains the following sections:

About This Guide

The *Brocade Virtual Traffic Manager: REST API Guide* describes the Traffic Manager product REST API.

This guide introduces you to the syntax and constructs used in the REST API, and is intended as a complete reference to all REST resources available in the Traffic Manager.

Introducing REST

REST (REpresentational State Transfer) is a framework for API design. It is based on generic facilities of the standard HTTP protocol, including the six basic HTTP methods (GET, POST, PUT, DELETE, HEAD, INFO) and the full range of HTTP return codes.

A REST interface partitions the API into a series of "resources," each of which can be accessed using one or more HTTP methods. (In the Traffic Manager, only the GET, PUT, and DELETE methods are used; HEAD, POST and INFO are not currently implemented). Each method operates in the Traffic Manager as follows:

- **GET**: Obtain a representation of the resource, without modifying server state (except perhaps for logging purposes).
- **PUT**: Create a new resource or apply some change to a resource. Where the resource exists, only those properties specified in the request are modified; all others remain unchanged. If a resource object does not exist, a new one is created.
- **DELETE**: Delete an existing resource.

Importantly, each resource is uniquely identified with an address, or URI (Uniform Resource Identifier). In other words, if you know the URI you can access the resource (subject to the normal authorization/ authentication processes associated with accessing the administrative systems of the Traffic Manager).

Since all resources have URIs, resources can point to other resources by embedding the URIs of related resources within their representations.

In the Traffic Manager, all resources are represented and stored as JSON (JavaScript Object Notation) structures. Requests and responses that interact with the Traffic Manager through the REST API must adopt the same format.

The full range of HTTP return codes is available in REST, although in practice a useful subset can be identified and applied consistently. So, for example, it should be evident from the response itself whether a request has succeeded or not, without any need for parsing the body of the response. However, the Traffic Manager always attempts to provide extra information regarding a failure into the response body. For more details, see "Errors" on page 19.

Why use a REST API

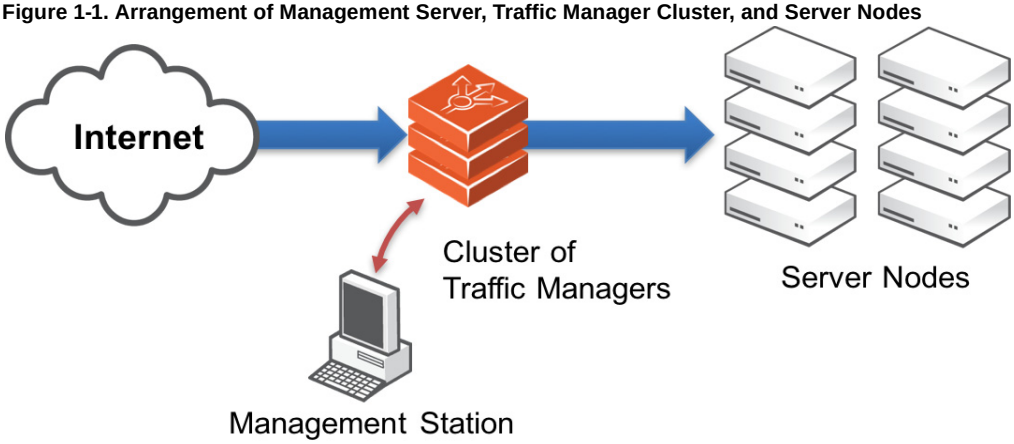
REST interfaces have become popular in public APIs because of their inherent simplicity. An API can focus on available resources, with details regarding updating and deleting of each resource delegated to the appropriate HTTP method in predictable ways.

The purpose of implementing a REST API is not primarily to add functionality but to add structure. Because of the inherent similarity of all REST APIs (by virtue of their underlying HTTP structure), familiarity with any REST API brings familiarity with all of them. In many cases it is just as easy to implement to a REST design as it is to use a more ad hoc API design, while reaping the benefits that come with well-understood REST conventions.

Finally, the availability of return codes is another example of leveraging known semantics when building a useful API. Without a meaningful return code it becomes necessary to parse every response to find out whether it worked or not. In addition, most modern browsers and Web programming frameworks expect that specific HTTP error codes are set in the event of error and respond differently depending on the code. This is especially apparent in the case of AJAX requests, which are often handled differently by many modern Javascript frameworks depending on the status code returned from the server.

A REST-Based Architecture

Figure 1-1. Arrangement of Management Server, Traffic Manager Cluster, and Server Nodes



A cluster of traffic mangers is normally managed using the web-based Administration UI on one of the machines. The Traffic Manager's REST API provides an alternative means to remotely administer and configure a Traffic Manager cluster.

The REST service is enabled by default for new Traffic Manager instances. To toggle the REST service between enabled and disabled, see "Enabling and Disabling the API" on page 23.

Note: For Traffic Managers running version 11.0 or earlier, the REST service is disabled by default. To preserve the integrity of your configuration, the REST service state is unchanged during an upgrade to a Traffic Manager version later than 11.0.

The Traffic Manager REST API can be used by any HTTP client or application environment that supports HTTP services.

The REST API is an interface used to configure, manage, and monitor a cluster of Traffic Managers remotely.

A management application can issue a REST request to any one of the Traffic Managers in a cluster. The application may be running on a stand-alone management server, one of the server nodes, or even on one of the Traffic Managers.

Important: When upgrading your Traffic Manager cluster to the latest software version, do not attempt to use the REST API until all cluster members have been successfully upgraded. Use of the REST API on a cluster in a mixed-version state can result in unexpected behavior.

The application can issue the request to any of your Traffic Managers. All Traffic Managers in the cluster automatically synchronize their configuration, so a configuration change sent to one machine is automatically replicated across the cluster.

Important: Due to the nature of the REST API's ability to access and modify your Traffic Manager configuration, it is strongly recommended that you disallow access to this service from outside of your trusted network.

Scope of This Release

This document describes the features and capabilities of the REST API for the Brocade Virtual Traffic Manager 17.2 release. The REST API version referred to in this document is 4.0.

The API performs basic type checking, although Brocade recommends that your client applications provide suitable validation to ensure the suitability of the configuration data being provided to the Traffic Manager.

All defined users in the system have the ability to authenticate a connection through the Traffic Manager REST API. However, you cannot modify the users configuration file in any way, so it is not possible to add, edit, or delete users through the API.

A full list of specific features, capabilities, and API versions supported by this release can be found in the release notes supplied with your product variant.

CHAPTER 2

Typical Usage in the Traffic Manager

This chapter describes the usage, syntax, and structure of the Traffic Manager REST API. This chapter contains the following sections:

The Resource Model

The Traffic Manager REST API is made up of a hierarchy of resources that are manipulated using standard HTTP calls to a listener service running along side the Traffic Manager. HTTP URIs are used to address the resources in the system.

There are three resource types:

- **Configuration**: to represent Traffic Manager configuration objects.
- **Counters**: for reporting through SNMP counters.
- **Information**: for system information.

Counters and Information resources are read-only, whereas Configuration resources are fully interactive and map directly to the native Traffic Manager configuration system. Each concept, such as pools, virtual servers, TrafficScript rules, or Service Level Monitoring classes, has an associated configuration resource model.

All resources are represented as JSON structures (MIME type `application/json`), and objects of each resource type are captured in this format.

Typically, a configuration resource follows this format:

```
{
   "properties": {
      "sectionname": {
         "key1": "stringvalue1",
         "key2": numericvalue2,
         "key3": booleanvalue3
      }
   }
}
```

A single instance of a resource, for example a virtual server, contains a primary group entitled "properties". This contains all configuration keys attributable to this resource type.

Counter resources contain dynamically generated data to correspond to SNMP counters in the Traffic Manager.

Sections

Sections are designed to contain properties (or "keys") that have a commonality of purpose or perhaps apply in certain circumstances. For example, monitor classes may have keys that apply only to monitors of particular types.

In a configuration resource, the properties group contains several sections, one for each logical set of keys. There is always a section entitled "basic," containing common configuration items, followed by one or more additional sections according to the specification of the resource.

A counter resource contains a single section, "statistics," listing the SNMP counters associated with the resource. Similarly, an information resource contains a single section, "information," listing the system information properties applicable to this Traffic Manager.

Data Types

Each key:value pair is then presented as a comma-separated list within each section, according to the specification shown throughout this guide. Key names are always delimited by quotes, with the values according to the following rules:

Type	Value

Boolean	A value of `true` or `false` (case-sensitive).
	For example:
	`"key1": true,`
	`"key2": false`

Int	A numeric positive or negative value with no decimal point.
	For example:
	`"key1": 1024,`
	`"key2": -10`

Unsigned Int	A numeric positive value with no decimal point.
	For example:
	`"key1": 0,`
	`"key2": 50`

Type	Value

Float	A numeric positive or negative value that can have a decimal point.
	For example:
	`"key1": 1.0,`
	`"key2": -1024.111`

String	A set of alpha-numeric characters that may not include new-lines. Non-alpha characters must use correct character escapes.
	For example:
	`"key1": "Hello world",`
	`"key2": "",`
	`"key3": "Hello y\'all"`

Freeform String	A set of alpha-numeric characters that can contain new-lines. Non-alpha characters must use correct character escapes, and a newline must be represented by a \n. For example: `"key1": "Multi-line\nString",`

Password	A string that cannot be read, only written to. When read, it is displayed as a structure that indicates if the password has been set (is non-empty).
	For example, when reading the key:
	`"key1": { "password_set": false },`
	`"key2": { "password_set": true }`
	When writing to the key, the structure can be unchanged, or a new password can be set:
	`"key2": { "password_set": true },`
	`"key1": "secret123"`

Time	Times are represented as strings in ISO8601 time format, including a time zone designator. For example: `{Year}-{Month}-{Day}T{Hour}:{Minute}:{Second}{Time Zone}`
Set	This is a collection of unique unordered items of a particular type, stored as an array. For consistency, a set is rendered in alpha-numeric order. For example: `"key": [ "Item A", "Item B", "Item D" ]`

Type	Value
List	This is a collection of ordered items of a particular type. It may contain duplicates and is stored as a standard array. For example: `"key": [ "Item A", "Item C", "Item A" ]`

Tables	This is a special type designed to allow nested data within a single config key. In some circumstances, you might wish to specify a list or array of data items, such as a list of pool nodes, where each item has one or more extra pieces of configuration data to be attached to it.
	Each one of these nested list entries expects a value known as the “primary key”, used to identify it. Each sub-key value should then be specified in the same way.
	For example:
	`"key": [`
	`  {`
	`  "prmkey": "Hello World",`
	`  "subkey1": false,`
	`  "subkey2": [ "Item 1", "Item 2" ]`
	`  },`
	`  {`
	`  "prmkey": "Other text",`
	`  "subkey1": true,`
	`  "subkey2": []`
	`  },`
	`]`

Resource URI Patterns

Important: Resource URIs are case-sensitive.

The Traffic Manager provides access to its resources through a common base URI that identifies the root of the resource model:

`https://<host>:<port>/api/tm/<version>`

In this URI path:

- <host>: The hostname of the Traffic Manager whose REST API you are accessing.
- <port>: The port that the REST API is published on (typically "9070").
- <version>: The version number of the REST API you are accessing. Details of supported versions are contained in the release notes supplied with your product variant.

Important: All client applications and scripts that access the Traffic Manager REST API must use the same supported major version of the API at any one time. The Traffic Manager does not support interactions using multiple versions concurrently.

Note: In the previous example, a scheme of HTTPS is used to signify an encrypted connection from a remote client. HTTP is supported only where the connection is to a server on the same host. For further details, see "Authentication" on page 16.

The Traffic Manager presents different resource types at specific child nodes under this root URI.

Configuration Resources

Configuration resources map to objects in the Traffic Manager's configuration system. Use the following URI pattern to access them:

```
https://<host>:<port>/api/tm/<version>/config/active
```

Instances of a particular configuration resource, such as a virtual server, are persistently stored and alter the host Traffic Manager's behavior if changed. Additionally, changes you make here are synchronized automatically to all other Traffic Managers in the cluster.

To view or modify a stored configuration resource record, append the full path to the end of the base URI. For example, to issue a request for a virtual server resource named "Web", use the following URI:

```
https://myhost:9070/api/tm/4.0/config/active/virtual_servers/Web
```

Counter and Information Resources

Counter resources map to SNMP counter objects generated by the Traffic Manager. Use the following URI pattern to access them:

```
https://<host>:<port>/api/tm/<version>/status/<tm>/statistics
```

Information resources provide basic information data about your cluster members. Use the following URI pattern to access them:

```
https://<host>:<port>/api/tm/<version>/status/<tm>/information
```

Conversely to configuration resources, instances of counter and information resources are specific to each Traffic Manager in the cluster. Furthermore, you can access the data for all of your cluster members from the base URI of any one of them by specifying the desired member hostname in the `/<tm>/` child node.

To request the SNMP counter data from cluster member "myhost2" for a pool named "P1", use the following URI:

```
https://myhost:9070/api/tm/4.0/status/myhost2/statistics/pools/P1
```

Traversing the Tree

Resource URIs can be either:

- Resources.

- A directory structure containing child elements denoting sub-directories or resource nodes.

You can test the overall availability of the REST API by querying the following URI in a compatible Web browser or JSON/REST query application:

```
https://<host>:<port>
```

(As mentioned above, `<host>` is the hostname of the Traffic Manager and `<port>` is the port that the REST API is published on).

Note: Some versions of Internet Explorer are unable to directly render the "application/json" MIME type data returned from a Traffic Manager REST API query. Brocade recommends using an alternative browser. However, to resolve the issue for affected browsers, amend the Windows registry as shown here:

1) Open Notepad and enter the following text:

```
Windows Registry Editor Version 5.00;
; Tell IE 7,8,9,10,11 to open JSON documents in the browser on Windows XP and later.
; 25336920-03F9-11cf-8FD0-00AA00686F13 is the CLSID for the "Browse in place" .
;
[HKEY_CLASSES_ROOT\MIME\Database\Content Type\application/json]
"CLSID"="{25336920-03F9-11cf-8FD0-00AA00686F13}"
"Encoding"=hex:08,00,00,00
```

2) Save the document as "IE-Json.reg" and then run it.

3) Restart Internet Explorer and re-enter a REST URI to confirm the JSON results appear within the browser.

For further information, see http://www.codeproject.com/Tips/216175/View-JSON-in-Internet-Explorer.

A GET request for this URI should yield the following result:

```
{
   "children": [{
      "name": "api",
      "href": "/api/"
   }]
},
```

This shows that the REST service at <host>:<port> contains a single child element "/api". As discussed in "Resource URI Patterns" on page 12, the full root URI of the configuration resource model then becomes:

```
https://myhost:9070/api/tm/4.0/config/active
```

Therefore, requesting this URI results in a list of child elements similar to the following example:

```
{
   "children": [{
      "name": "action_programs",
      "href": "/api/tm/4.0/config/active/action_programs/"
   }, {
      "name": "actions",
      "href": "/api/tm/4.0/config/active/actions/"
   }, {
      "name": "aptimizer",
      "href": "/api/tm/4.0/config/active/aptimizer/"
   }, {
      "name": "bandwidth",
      "href": "/api/tm/4.0/config/active/bandwidth/"
   }, {
      "name": "cloud_api_credentials",
      "href": "/api/tm/4.0/config/active/cloud_api_credentials/"
   }, {
   ...
```

```
   (truncated)
   ...
   }, {
      "name": "virtual_servers",
      "href": "/api/tm/4.0/config/active/virtual_servers/"
   }]
}
```

This output identifies all configuration resource types available through the Traffic Manager being queried. Each is identified by a name and href attribute combination.

A query for a specific resource type shows all instances of that resource defined within the Traffic Manager configuration. For example, the following URI lists all virtual servers:

```
https://myhost:9070/api/tm/4.0/config/active/virtual_servers
```

The output shows each stored virtual server, as per the following example:

```
{
   "children": [{
      "name": "vs1",
      "href": "/api/tm/4.0/config/active/virtual_servers/vs1"
   }, {
      "name": "vs2",
      "href": "/api/tm/4.0/config/active/virtual_servers/vs2"
   }]
}
```

SNMP counter and system information resources are unique to each Traffic Manager in the cluster. You can access the data for each cluster member from the API of whichever Traffic Manager you are connected to.

To list the available Traffic Managers in your cluster, perform a request for the following URI:

```
https://myhost1:9070/api/tm/4.0/status
```

The response is a list of child elements similar to the following:

```
{
   "children": [{
      "name": "myhost1.example.com",
      "href": "/api/tm/4.0/status/myhost1.example.com/"
   }, {
      "name": "myhost2.example.com",
      "href": "/api/tm/4.0/status/myhost2.example.com/"
   }, {
      "name": "myhost3.example.com",
      "href": "/api/tm/4.0/status/myhost3.example.com/"
   }, {
      "name": "local_tm",
      "href": "/api/tm/4.0/status/local_tm/"
   }]
}
```

The list also includes a “local_tm” child node that corresponds to the REST API of the Traffic Manager you are currently accessing. This provides a consistent programmatic interface to access resources for the local Traffic Manager only, no matter which host's API you are connected to. For example, the following URI can be used on the API of any Traffic Manager in the cluster, and the response contains results for that Traffic Manager only:

```
/api/tm/4.0/status/local_tm/information
```

To view (or modify, in the case of configuration resources) a stored record for a particular resource type, append the full path to the end of this base URI. For example, a request for a virtual server configuration resource named "Web" looks like this:

```
https://myhost:9070/api/tm/4.0/config/active/virtual_servers/Web
```

Equally, a request for the SNMP counter output for a pool named "P1" looks like this:

```
https://myhost:9070/api/tm/4.0/status/local_tm/statistics/pools/P1
```

The Traffic Manager REST Service

The Traffic Manager REST API is an HTTP service running on the Traffic Manager server. By default, it is available on TCP port 9070, although this can be reconfigured. The REST service supports HTTP versions: 0.9, 1.0, and 1.1; Version 1.1 is recommended.

When connecting to the local machine using a loop-back interface (for example, 127.0.0.1 or "localhost"), plain HTTP must be used. When connecting from a remote machine, connections must be encrypted using SSL (HTTPS).

The service uses the same SSL certificate as the Traffic Manager's admin server, which by default is an automatically generated self-signed certificate. Any HTTP client used to connect to the REST API should have the server's self-signed certificate added to its trusted certificate catalogue. Alternatively the admin server/REST certificate can be replaced with one signed by a trusted certificate authority.

The Brocade Virtual Web Application Firewall REST Interface

The Brocade Virtual Web Application Firewall (vWAF) component maintains a separate REST interface to facilitate control of vWAF-specific resources. You can reach this interface through the standard Traffic Manager REST service, using the following path:

```
https://<host>:<port>/api/af/<version>
```

<version> can be any specific currently published API version, or you can use the string "latest" to access the most current version. Full details of the available resources and actions that can be performed through the vWAF REST API can be found in the vWAF user documentation.

To access the vWAF REST interface, you must first install and activate the vWAF component on your Traffic Manager. You must also enable the Traffic Manager REST service through the Admin UI (for details, see "Enabling and Disabling the API" on page 23).

The Traffic Manager operates as a proxy to the vWAF REST service, and communicates with it through a designated port. To view and modify this port, click **System > Application Firewall > Local Application Firewall Ports** in the Admin UI. Any problems accessing the vWAF REST interface can often be resolved by setting this value to a known free port. Contact your support provider if you require any further information.

Authentication

A REST-based management application communicates with a configuration service running on the Admin Server (the Traffic Manager-based service used to provide the Admin UI), so the same security considerations apply:

- REST requests are authenticated using HTTP Basic Auth.

- REST traffic over HTTPS is automatically encrypted using SSL. Traffic over HTTP is not encrypted, so should only be used inside a secure environment or to/from localhost.
- The Traffic Manager Admin Server authenticates itself with its SSL certificate, which is generally self-signed. You might need to ensure that your REST application accepts self-signed certificates, or install a trusted SSL certificate in your Traffic Manager.
- REST requests are authenticated using the same user credentials as defined in the Administration Server. Individual object access is synonymous with page access in the Admin UI. For example, if a user wishes to view and manipulate pool objects, they must have been granted access to pools on the access permissions page.

Supported HTTP Methods

The REST service supports three primary HTTP methods for accessing and modifying data in the Traffic Manager configuration system:

- GET
- PUT
- DELETE

GET is used when making read-only requests for a resource, whereas PUT is used when updating existing data or adding new configuration objects. DELETE is used when you wish to completely remove configuration objects from the Traffic Manager.

Each of these methods is discussed in more detail in the sections that follow.

Note: The REST service additionally supports the POST method for uploading files to the Traffic Manager, specifically with respect to configuration backup TAR archive files. For more information, see "Configuration Backups" on page 20.

Requesting a Resource

A client interacts with the Traffic Manager REST API by performing operations on its resources. An operation is distinguished by the HTTP method it uses and the path and query components of the associated URI. Some operations, however, are not applicable to every resource.

The GET method is used to retrieve the current representation of the resource it is used on. It does not alter the resource in any way or have any other side effects.

This is achieved by sending a HTTP GET request to the server with no body. The request must accept a response in JSON format only (by specifying an "Accept" header type of `application/json`), and authorization is provided using *HTTP Basic Auth* (for more details, see "Authentication" on page 16). Such a request resembles the following:

```
GET /api/tm/4.0/config/active/bandwidth/BWClass1 HTTP/1.1
Authorization: Basic YWRtaW46c2VjcmV0MTIz
Accept: application/json
```

If successful, the server returns a "200 OK" response code with the full resource in the response body. The above Bandwidth class example might produce the following output:

```
{
   "properties": {
```

```
      "basic": {
         "maximum": 10000,
         "note": "This is my bandwidth class",
         "sharing": "cluster"
      }
   }
}
```

This is a JSON structure representing the configuration keys present in the requested bandwidth class object. In this case, it consists of a single "basic" section containing three key:value pairs. Other resource types might contain different or additional sections and corresponding keys.

Setting Configuration for a Resource

Note: This section does not apply to read-only resources such as SNMP counters or system information.

Changing data items in the Traffic Manager configuration system is achieved through a PUT request to a configuration type resource. This applies to either creating new resource items or updating the properties of an existing resource item.

When creating a new resource item, the request URI must contain the full path to the intended item, with the name being the final element of the path. For example, creating a new bandwidth class called "mynewclass" requires using the following URI:

```
/api/tm/4.0/config/active/bandwidth/mynewclass
```

For both creation and update operations, the request body must contain a representation of the resource properties in JSON format (with the appropriate body "Content-Type" header set). Partial updates to configuration resources can be performed by only including the properties that need to be altered. Other properties are left unchanged.

Note: For PUT requests, ensure that the request body is encoded as UTF-8.

The REST service returns a "200 OK" response for a correctly updated configuration set, or "201 Created" for establishing a new configuration object of a particular resource type. In these cases, the full resource is returned as the response body. The only exception to this rule is when updating a raw file, which instead returns a "204 No Content" empty-body response.

Important: You might want to exercise some care when creating or updating resources. The changes are permanent and no warning is given for existing configuration that is overridden. If you attempt to create a new resource where one of the same name already exists, you overwrite the properties of the existing record. It is recommended that you build such validation into your REST client application.

Removing Resources

Note: This section does not apply to read-only resources such as SNMP counters or system information.

A HTTP DELETE request for the full URI of a configuration item can be sent to the REST server to permanently remove it. On success, a "204 No Content" empty-body response is returned.

Further Aspects of the Resource Model

Enumerated Types

Some configuration keys can accept one or more of a pre-defined set of values. This is known as an enumerated key type, and the list of possible values (with long description) is provided in the reference guide later in this document.

Uploading Files

Resources that represent real files (such as TrafficScript rules) can also be presented in a raw format, where the data returned is the contents of the file. The MIME type of the request payload should be set to `application/octet-stream`.

Custom Configuration Sets

You can store and retrieve arbitrary name:value configuration pairs in the Traffic Manager configuration system using the REST API. This configuration is replicated across your cluster, and is only accessible through the REST API, SOAP API, and ZCLI.

To store a custom configuration, create an instance of the “custom” resource and set your name:value data to the ”string_lists” property. For example, to create a resource called "customdata", use the following URI:

```
https://myhost:9070/api/tm/4.0/config/active/custom/customdata
```

Set the request body to a JSON structure resembling the following:

```
{
   "properties": {
      "basic": {
         "string_lists": [{
            "name": "customname1",
            "value": ["val1", "val2"]
         },{
            "name": "customname2",
            "value": "val3"
         }]
      }
   }
}
```

Using this system, you can organize your custom configuration into logical groups, initially by an instance of the “custom” resource, and within this, by a name:value pair. Each value can itself be a single item or a list of items.

Errors

If the REST server is unable to handle a HTTP request, it returns a HTTP response with an appropriate HTTP error code. The response body is in JSON and contains a data structure describing the error with a unique identifier (separate to the numeric error code) and a description.

The unique identifier is made up of 2 parts:

```
{section}.{error_type}
```

Some errors might provide additional formatted information, specified with an optional "error_info" parameter. For example, the REST API uses this parameter to return per-property errors when a value fails validation. The following structure demonstrates the general form of an error:

```
{
  "error": {
     "error_id":   "{error identifier}",
     "error_text": "{error description}",
     "error_info": {error specific data structure, optional}
  }
}
```

A validation error occurs when one or more of the properties within a configuration resource fail a validation check. The error_info section then contains a sub-error for each property that failed validation. These sub-errors are like normal errors in that they contain an identifier (error_id) and a human readable text description (error_text):

```
{
  "error": {
     "error_id":   "resource.validation_failed",
     "error_text": "Some of the properties in the resource failed validation.",
     "error_info": {
        "basic": {
           "key1": {
              "error_id": "num.range",
              "error_text": "Value must be in range 1000 - 2000."
           }
        }
     }
  }
}
```

Configuration Backups

The REST service supports the ability to create new backup sets of your Traffic Manager configuration.

When a user creates a new backup in the Admin UI, the Traffic Manager copies the current configuration folder ($ZEUSHOME/zxtm/conf) into a new backup directory under $ZEUSHOME/zxtm/confarchive. The name of the backup directory reflects the name chosen in the Admin UI.

Each backup set is saved with a series of metadata files:

Filename	Purpose
DESCRIPTION	Description of the backup
NAME	Optional file containing the name of the backup
VERSION_<version>	File to indicate the software version of the Traffic Manager used to create the backup. This file has no contents.
TIMESTAMP	File indicating when the backup was created.
PARTIAL	Indicates that the backup is a partial backup.

To learn more about configuration backups, see the *Brocade Virtual Traffic Manager: REST API Guide*.

Note: The REST service exposes only the name and description metadata for your backups, although each listed file is still present in a backup archive on the Traffic Manager.

Backups are represented through the REST API as child resources of the System Information resource:

```
/api/tm/4.0/status/<tm>/backups/full
```

As with all System Information resources, you must specify the cluster member you want to interrogate (denoted by `<tm>` in the URI shown). Each child resource of this URI is named in accordance with the corresponding backup name.

Listing Existing Backups

To see the configuration backups currently stored in a Traffic Manager, perform a GET request for the backups parent resource:

```
/api/tm/4.0/status/<tm>/backups/full
```

If successful, the server returns a "200 OK" response code with the backup list in the response body:

```
{
   "children": [{
      "name": "Backup 1",
      "href": "/api/tm/3.9/status/mytm/backups/full/Backup%201/"
   }, {
      "name": "vTMBackup-May16",
      "href": "/api/tm/3.9/status/mytm/backups/full/vTMBackup-May16/"
   }, {
      "name": "another backup",
      "href": "/api/tm/3.9/status/mytm/backups/full/another%20backup/"
   }]
}
```

To observe the metadata for a specific backup, perform a GET request for a child name. For example, a request for `/api/tm/4.0/status/<tm>/backups/full/vTMBackup-May16` might return the following JSON strucure in the response body:

```
{
   "properties": {
      "backup": {
         "description": "Backup for May 2016",
         "time_stamp": 1462445371,
         "version": "17.2"
      }
   }
}
```

Creating a Backup

To create a new backup, use an HTTP PUT request to the backups URI:

```
/api/tm/4.0/status/<tm>/backups/full/<backup_name>
```

Set <backup_name> as the name of the backup you want to create.

The request body must contain a JSON resource representing the backup, with an optional description:

```
{
   "properties": {
      "backup": {
```

```
        "description": "My vTM backup for today",
      }
   }
}
```

Set the Content-Type header to “application/json”.

If your backup request is successful, the Traffic Manager returns a “201 Created” response with the full backup metadata in the response body. If the Traffic Manager cannot create the backup, the response contains an HTTP error and the body contains an explanation of why the backup was unsuccessful.

Caution: Make sure your backup name does not already exist. The Traffic Manager does not override existing backups and instead returns an error in the response.

Restoring a Backup

Caution: Restoring a backup means your current configuration is overwritten permanently and all unsaved changes are lost. Brocade strongly recommends only performing the following procedure if you are sure of the results.

To restore a backup to be your current Traffic Manager configuration, send a PUT request with the name of the selected backup in the URI, with a “restore” argument:

```
/api/tm/4.0/status/<tm>/backups/full/MyBackup?restore
```

Set the request body to an empty properties structure, such as the following:

```
{
   "properties":{
   }
}
```

If your restore request is successful, the Traffic Manager returns a “200 OK” response with the backup metadata in the response body. If the Traffic Manager is unable to restore the backup, the response contains an HTTP error and the body contains an explanation of why the operation was unsuccessful.

Downloading Backups

To download a backup as a TAR archive file, peform a GET request with the name of the required backup appended to the base backups URI. For example:

```
/api/tm/4.0/status/<tm>/backups/full/MyBackup
```

Set the “Accept” header to “application/x-tar”.

When the request is made, a TAR file will be generated, which will be streamed to the client.

Some backup clients are able to make the request, but cannot properly handle an application/x-tar response. In this case, Brocade recommends a command line tool such as CURL. The following command is an example of how to download a backup with CURL:

```
curl -H "accept:application/x-tar"
     -u <username>:<password>
     http://<host>:<port>/api/tm/4.0/status/<tm>/backups/full/MyBackup > MyBackup.tar
```

Uploading Backups

To upload a configuration backup file to your Traffic Manager /confarchive directory, use an HTTP POST request to the base backups URI:

```
/api/tm/4.0/status/<tm>/backups/full
```

The name of the backup is not required in the URI; instead, the Traffic Manager uses the contents of the "NAME" file in the backup TAR archive file. Make sure the backup name does not already exist in the Traffic Manager backups list.

Before you send the request, attach your backup archive file and set the Content-Type header to "application/x-tar". The actual process of uploading or attaching the archive file depends on your REST client.

The REST service returns a "201 Created" response for a correctly uploaded configuration backup archive file. If the Traffic Manager is unable to upload the backup file, the response contains an HTTP error and the body contains an explanation of why the operation was unsuccessful.

Deleting Backups

To delete a configuration backup, perform a DELETE request with the full URI of the backup you want to remove. For example, to remove a backup called "MyBackup", use the following URI:

```
/api/tm/4.0/status/<tm>/backups/full/MyBackup
```

If the request is successful, a "204 No Content" empty-body response is returned.

Traffic Manager UI Features

Enabling and Disabling the API

To enable or disable the REST service, click **System > Security > REST API** in the Traffic Manager Admin UI. You can additionally use this page to set the TCP port that the service listens on. The default port is 9070, however any unreserved port can be used here provided it does not conflict with other services already running on the Traffic Manager system. Click **Update** to apply any changes.

Important: The REST API is currently not available in conjunction with the Traffic Manager's Multi-Site Manager (MSM) feature. Attempts to enable the REST service whilst MSM is operational are denied. Equally, attempting to enable MSM whilst the REST service is running triggers an error. The current state of the Traffic Manager remains unchanged in either of these situations.

To manually restart the REST API service, click **System > Traffic Managers** and then, in the Software Restart section, click **Restart REST API...**. The Traffic Manager asks you to confirm the restart on the next screen. Note that any existing connections are lost while the service restarts.

Controlling Timeout Events

The **System > Security > REST API** page provides a number of settings to control how the Traffic Manager responds to certain timeout events that occur through use of the REST API. These are:

Setting	Description
rest!auth_timeout	The timeout period, in seconds, for the REST Authentication cache. As REST does not include the concept of a "session", each request must include user and password credentials. These credentials are validated each time; however, to save requesting repeated external authentications for the same user (from the same IP address), a cache of recent authentications is maintained. This timeout is the maximum amount of time a given user's credentials can stay in the cache. A setting of 0 (zero) disables the cache, forcing every REST request to be authenticated as it is received. However, this affects the performance of the API. (Default: 120 seconds)
rest!replulltime	This is the *lull time* for configuration replication using REST. This is the time, in seconds, of inactivity through the REST API before configuration replication starts. Increasing this value delays configuration replication among a cluster of Traffic Managers. (Default: 5 seconds)
rest!repabstime	This is the absolute timeout prior to configuration replication using REST. This is the longest time, in seconds, before configuration replication using REST starts, regardless of activity through the API. (Default: 20 seconds)
rest!reptimeout	The configuration replication duration timeout using REST. This is the time, in seconds, allowed for the process of configuration replication to complete. On a system with slow cluster communications or a very large configuration, increasing this value improves replication reliability. (Default: 10 seconds)

Configuring the IP Addresses That the REST API Listens On

System > Security > REST API contains a setting, rest!bindips, that can be used to control the IP address(es) that the REST API listens on for connections. This can be a space-separated or comma-separated list of IPv4 or IPv6 addresses. Alternatively, it can contain an entry of "*", in which case the REST API listens on all IP addresses.

If you configure the Traffic Manager to use a management IP address during initial configuration, rest!bindips defaults to this same IP address. You can, however, override this later with a separate IP address or list of IP addresses specific to the REST API.

Important: rest!bindips is a machine-specific setting and is not included in the cluster configuration that is replicated out to joining Traffic Manager instances. If you intend to join an unrestricted Traffic Manager into a cluster that has been previously configured to use a management IP address, you must ensure rest!bindips is set on the new instance before you join the cluster to avoid exposing unintended access.

The addresses that are bound to are listed in the error log. Addresses to which the REST API cannot be bound are also logged. If no addresses can be bound, the REST API shuts down.

Restricting Access to Trusted Users

In addition to username/password access, click **System > Security > Restricting Access** to further restrict access to the administrative capabilities of your Traffic Manager system to a set of trusted IP addresses, CIDR subnets, or DNS wildcards. Access to the REST API is also affected by this capability.

Log Messages in the Traffic Manager

The Event Log

A number of specific API-related messages might be found in the Traffic Manager event log under certain conditions:

- "REST API started: https://<URI>"

  Raised when the REST Daemon starts.

- "REST API is shutting down"

  Raised when the REST Daemon closes down.

- "On IPv6 host but cannot set unspecified ip address to ::"

  Raised when the REST Daemon can't set itself up to listen on the IPv6 wildcard address.

- "Could not open Traffic Manager PID file for read: <error>"

  Raised when REST Daemon can't identify the Traffic Manager PID, and so can't signal it to reload its config after a change has been made via the REST API.

- "Could not open Traffic Manager PID file: <error>"

  Raised when REST Daemon can't identify the Traffic Manager PID, and so can't signal it to reload its config after a change has been made via the REST API.

- "Failed to write to audit log: <error>"

  Raised when the REST Daemon can't add lines to the audit log.

The Audit Log

The audit log records login attempts, configuration changes, and user logouts. It also records changes made using the Traffic Manager Control API, and through the Traffic Manager Command-Line Interface (CLI). Configuration changes made through the REST API follow the same behavior.

In addition to the typical configuration messages entered into the audit log, the Traffic Manager also provides the ability to track user activity in the REST API. It does this by grouping REST request/response exchanges made in close succession from a given user into a "session".

The Traffic Manager logs the first request in a group of one or more requests from a particular user/ip address combination in the audit log as a "session start". Requests received after the initial request are deemed to be part of the same user session. Then, after a specified timeout interval since the most recent request was received from the same user, a "session end" is logged.

CHAPTER 3 Examples and Use-Cases

This chapter provides examples and use-cases for the REST API. This chapter contains the following sections:

Typical Usage

Important: The following examples are developed to work with version 4.0 of the Traffic Manager REST API. Brocade makes no warranty as to their suitability for older versions.

The following code samples demonstrate how to interact with the REST API for a variety of purposes. The examples are based on Perl using the "REST::Client" module to handle the connections to the Traffic Manager REST daemon.

For further information on REST::Client, see the CPAN Web site: www.cpan.org.

A typical Perl client connection might resemble the following:

```
#!/usr/bin/perl

use REST::Client;
use strict;

# Set up the connection
my $client = REST::Client->new();
$client->setHost( 'https://myhost:9070' );
$client->addHeader( 'Authorization', 'Basic YWRtaW46am9iYmll' );
$client->addHeader( 'Content-Type', 'application/json' );

# Perform a HTTP GET on this URI
$client->GET( '/api/tm/4.0/config/active' );

# Print out the response body
print $client->responseContent();
```

In the above example, a new connection is established to the REST service on the Traffic Manager "myhost" on port 9070.

The setHost() function allows us to set up a definitive hostname and port to which all requests are made. This is an optional feature, and the full hostname can be supplied when making the actual request if multiple hosts are required.

Two HTTP headers can be added here, one to provide Basic Auth authentication and the other to provide a declaration of the Content Type when making PUT requests. In the majority of cases, the content type is "application/json", apart from transactions involving raw files where it is necessary to use "application/octet-stream".

A GET request is sent to the REST service with a target of the resource URI as the supplied argument. Typically, the above script outputs a JSON structure showing the Traffic Manager resource tree at the top level:

```
{
    "children": [{
        "name": "rules",
        "href": "/api/tm/4.0/config/active/rules/"
    }, {
        "name": "actions",
        "href": "/api/tm/4.0/config/active/actions/"
    },
    ...
    (truncated)
    ...
   {
        "name": "auth",
        "href": "/api/tm/4.0/config/active/auth/"
    }]
}
```

Note: Each of the following examples make use of a further Perl module "JSON" in order to encode and decode between the JSON string used by REST::Client and a native Perl structure. This is done to simplify the parsing algorithm within the script. Further information regarding the JSON Perl module can be found at the CPAN Web site: www.cpan.org.

Listing Running Virtual Servers

In this example, we collect data on stored virtual servers by querying the “vservers” resource and identifying which ones are enabled (running).

The code structure is as follows:

- Instantiate a new REST Client object.
- Specify the hostname and port of the REST service to which all requests are to be directed.
- Add required HTTP headers for authentication and content type.
- Send a GET request for the “vservers” resource in order to return a list of all Virtual Servers on the system.
- Check the response body, and decode from JSON into a Perl structure. This value is a hash ref.
- Identify the “children” hash key, and iterate through the array to which it points.
- Each array item contains a hash of “name” and “href” associative values.
- Using the “name” value, perform a new GET request to return the full configuration for this named virtual server resource.

- Again, using the decoded JSON response body, identify the Boolean value of the "enabled" key in the "basic" configuration section. If it is "true", this virtual server is running so print it's name to STDOUT.

Important: This script does not contain any error checking in order to best demonstrate the basic functionality. It is strongly recommended you incorporate return value checking and other validation mechanisms as appropriate.

```
#!/usr/bin/perl

use REST::Client;
use JSON;
use strict;

# Set up the connection
my $client = REST::Client->new();
$client->setHost( 'https://myhost:9070' );
$client->addHeader( 'Authorization', 'Basic YWRtaW46am9iYmll' );
$client->addHeader( 'Content-Type', 'application/json' );

# Request a list of all virtual servers
client->GET( '/api/tm/4.0/config/active/vservers' );

# Decode response into a perl structure for easy parsing
my $response = decode_json( $client->responseContent() );

# Obtain a reference to the children array
my $vsArrayRef = $response->{children};

# For each VS, make a request for its configuration and
# check the Boolean value of the 'enabled' key
foreach my $vs ( @$vsArrayRef ) {
   my $vsName = $vs->{name};
   $client->GET( "/api/tm/4.0/config/active/vservers/$vsName" );
   my $vsConfig = decode_json( $client->responseContent() );
   if( $vsConfig->{properties}->{basic}->{enabled} eq "true" ) {
      # Print the name of this matched VS
      print "$vsn\n";
   }
}
```

The expected output of a script such as this would be:

```
$ ./listVS.pl
Main Website
Intranet
Support Site
```

Adding a Node to a Pool

Provisioning systems can dynamically deploy applications across servers, perhaps in reaction to increased server load. This example demonstrates an application that modifies the nodes that a pool balances traffic to.

The code structure is as follows:

- Instantiate a new REST Client object.
- Specify the hostname and port of the REST service to which all requests are to be directed.

- Add required HTTP headers for authentication and content type.
- Send a GET request for the pool that the new node is added to. Check the response body, and decode from JSON into a Perl structure. This value is a hash ref.
- The new node must be added to the table of existing nodes before writing the data back to the pool resource. Failing to do this results in the existing table being overwritten with a single row containing the new node. Each of the subkeys associated with the node have default values and do not need to be specified.
- Re-encode the Perl structure into JSON and pass as an argument to the PUT request (using the pool name URI as the target).
- In this example, the script performs a check on the response code to ensure any problems are reported back (where the response code is not 200 OK).
- There is an optional portion of code at the end to iterate through the stored node table to ensure that the new node name appears.

```
#!/usr/bin/perl -w

use REST::Client;
use JSON;
use strict;

# Set up the connection
my $client = REST::Client->new();
$client->setHost( 'http://localhost:9070' );
$client->addHeader( 'Authorization', 'Basic YWRtaW46am9iYmll' );
$client->addHeader( 'Content-Type', 'application/json' );

# Our pool and new node details
my $poolName = "WebPool";
my $newNode = { "node" => "www3.brocade.com:80" };

# Get the config for the pool in question
$client->GET( "/api/tm/4.0/config/active/pools/$poolName" );
my $poolConfig = decode_json( $client->responseContent() );

# Find the existing nodes table (a hashref), and add our new node
my $nodesRef = $poolConfig->{properties}->{basic}->{nodes_table};
push @$nodesRef, $newNode;

# Re-encode as a JSON string
my $poolStr = encode_json( $poolConfig );

# Now send a PUT request to the REST service
$client->PUT( "/api/tm/4.0/config/active/pools/$poolName",
              $poolStr );

# Print out the response code if we were NOT successful
if( $client->responseCode() ne '200' ) {
   die "FAILED with HTTP code: " . $client->responseCode();
}

# We're done! Verify that the node has been added
$client->GET( "/api/tm/4.0/config/active/pools/$poolName" );
$poolConfig = decode_json( $client->responseContent() );
print "Stored nodes for pool '$poolName':\n";
foreach my $tablerow ( @{$poolConfig->{properties}->{basic}->{nodes_table}} ) {
   print "$tablerow->{node}\n";
}
```

The expected output of a script such as this would be:

```
$ ./addNode.pl
Stored nodes for pool 'WebPool':
www1.brocade.com:80
www2.brocade.com:80
www3.brocade.com:80
```

CHAPTER 4

Resource Model Reference

This chapter provides a complete reference listing for the Traffic Manager REST API resource model. This chapter contains the following sections:

About the Resource Model Reference

This chapter lists all the configuration, counter, and information resources available through the REST API model.

Each section relates to a specific resource and lists its name, description, and unique URI path, and provides a table of properties.

For each property, you can find the description and data type. Additional information is provided where applicable, such as default value, permitted values (for enumerated types), and SNMP counter name. For Table-type properties, a list of the Primary and Sub keys is provided.

The path to use in your URIs is listed for each resource. For example, the URI path for a virtual server configuration resource is "virtual_servers", so to address a stored virtual server named "foo", you would use:

```
/api/tm/4.0/config/active/virtual_servers/foo
```

Configuration Resources

Action Program

URI Path: action_programs

This is a program or script that can be referenced and used by actions of type 'Program'

Property	Description
There are no properties to display for this resource.	

Alerting Action

URI Path: actions

A response to an event occurring in your traffic manager. An example of an action might be sending an email or writing a line to a log file.

Property	Description
note	A description of the action. • Type: FreeformString • Default value: <none>
syslog_msg_len_limit	Maximum length in bytes of a message sent to the remote syslog. Messages longer than this will be truncated before they are sent. • Type: UInt • Default value: "1024"
timeout	How long the action can run for before it is stopped automatically (set to 0 to disable timeouts). • Type: UInt • Default value: "60"
type	The action type. • Type: Enum(String) • Default value: <none> • Permitted values: "email": E-Mail "log": Log to File "program": Program "soap": SOAP Callback "syslog": Log to Syslog "trap": SNMP Notify or Trap
verbose	Enable or disable verbose logging for this action. • Type: Boolean • Default value: false
Properties for the "email" section:	
server	The SMTP server to which messages should be sent. This must be a valid IPv4 address or resolvable hostname (with optional port). • Type: String • Default value: <none>

Property	Description
to	A set of e-mail addresses to which messages will be sent. • Type: Set(String) • Default value: <none>
Properties for the "log" section:	
file	The full path of the file to log to. The text %zeushome% will be replaced with the location where the software is installed. • Type: String • Default value: <none>
from	The e-mail address from which messages will appear to originate. • Type: String • Default value: "vTM@%hostname%"
Properties for the "program" section:	
arguments	A table containing arguments and argument values to be passed to the event handling program. • Primary key: – name (String): The name of the argument to be passed to the event handling program. • Sub keys: – value (String): The value of the argument to be passed to the event handling program. – description (String): A description for the argument provided to the program.
program	The program to run. • Type: String • Default value: <none>
Properties for the "soap" section:	
additional_data	Additional information to send with the SOAP call. • Type: String • Default value: <none>
password	The password for HTTP basic authentication. • Type: Password • Default value: <none>
proxy	The address of the server implementing the SOAP interface (For example, https://example.com). • Type: String • Default value: <none>
username	Username for HTTP basic authentication. Leave blank if you do not wish to use authentication. • Type: String • Default value: <none>
Properties for the "syslog" section:	

Property	Description
sysloghost	The host and optional port to send syslog messages to (if empty, messages will be sent to localhost). • Type: String • Default value: <none>
Properties for the "trap" section:	
auth_password	The authentication password for sending a Notify over SNMPv3. Blank to send unauthenticated traps. • Type: Password • Default value: <none>
community	The community string to use when sending a Trap over SNMPv1 or a Notify over SNMPv2c. • Type: String • Default value: <none>
hash_algorithm	The hash algorithm for SNMPv3 authentication. • Type: Enum(String) • Default value: "md5" • Permitted values: "md5": MD5 "sha1": SHA-1
priv_password	The encryption password to encrypt a Notify message for SNMPv3. Requires that authentication also be configured. Blank to send unencrypted traps. • Type: Password • Default value: <none>
traphost	The hostname or IPv4 address and optional port number that should receive traps. • Type: String • Default value: <none>
username	The SNMP username to use to send the Notify over SNMPv3. • Type: String • Default value: <none>
version	The SNMP version to use to send the Trap/Notify. • Type: Enum(String) • Default value: "snmpv1" • Permitted values: "snmpv1": SNMPv1 "snmpv2c": SNMPv2c "snmpv3": SNMPv3

Aptimizer Application Scope

URI Path: aptimizer/scopes

Application scopes define criteria that match URLs to specific logical web applications hosted by a virtual server.

Property	Description
canonical_hostname	If the hostnames for this scope are aliases of each other, the canonical hostname will be used for requests to the server. • Type: String • Default value: <none>
hostnames	The hostnames to limit acceleration to. • Type: Set(String) • Default value: <none>
root	The root path of the application defined by this application scope. • Type: String • Default value: "/"

BGP Neighbor

URI Path: bgpneighbors

The conf/bgpneighbors directory contains configuration files for BGP neighbors. The name of a file is the name of the neighbor configuration that it defines. BGP neighbors can be managed under the System > Fault Tolerance > BGP Neighbors section of the Admin UI, or by using functions under the BGPNeighbors section of the SOAP API and CLI.

Property	Description
address	The IP address of the BGP neighbor • Type: String • Default value: <none>
advertisement_interval	The minimum interval between the sending of BGP routing updates to neighbors. Note that as a result of jitter, as defined for BGP, the interval during which no advertisements are sent will be between 75% and 100% of this value. • Type: UInt • Default value: "5"
as_number	The AS number for the BGP neighbor • Type: UInt • Default value: "65534"
authentication_password	The password to be used for authentication of sessions with neighbors • Type: String • Default value: <none>
holdtime	The period after which the BGP session with the neighbor is deemed to have become idle - and requires re-establishment - if the neighbor falls silent. • Type: UInt • Default value: "90"

Property	Description
keepalive	The interval at which messages are sent to the BGP neighbor to keep the mutual BGP session established. • Type: UInt • Default value: "30"
machines	The traffic managers that are to use this neighbor • Type: Set(String) • Default value: <none>

Bandwidth Class

URI Path: bandwidth

A Bandwidth class, which can be assigned to a virtual server or pool in order to limit the number of bytes per second used by inbound or outbound traffic.

Property	Description
maximum	The maximum bandwidth to allocate to connections that are associated with this bandwidth class (in kbits/second). • Type: UInt • Default value: "10000"
note	A description of this bandwidth class. • Type: FreeformString • Default value: <none>
sharing	The scope of the bandwidth class. • Type: Enum(String) • Default value: "cluster" • Permitted values: "cluster": Bandwidth is shared across all traffic managers "connection": Each connection can use the maximum rate "machine": Bandwidth is shared per traffic manager

Brocade Virtual Web Application Firewall

URI Path: application_firewall

The conf/zeusafm.conf file contains configuration files for the application firewall. Some keys present in the zeusafm.conf are not documented here. Refer to the Brocade Virtual Web Application Firewall documentation for further details. The configuration can be edited under the System > Application Firewall section of the Administration Server or by using functions under the AFM section of the SOAP API and CLI.

Property	Description
There are no properties to display for this resource.	

Cloud Credentials

URI Path: cloud_api_credentials

Cloud credentials used in cloud API calls

Property	Description
api_server	The vCenter server hostname or IP address. • Type: String • Default value: <none>
cloud_api_timeout	The traffic manager creates and destroys nodes via API calls. This setting specifies (in seconds) how long to wait for such calls to complete. • Type: UInt • Default value: "200"
cred1	The first part of the credentials for the cloud user. Typically this is some variation on the username concept. • Type: String • Default value: <none>
cred2	The second part of the credentials for the cloud user. Typically this is some variation on the password concept. • Type: Password • Default value: <none>
cred3	The third part of the credentials for the cloud user. Typically this is some variation on the authentication token concept. • Type: Password • Default value: <none>
script	The script to call for communication with the cloud API. • Type: String • Default value: <none>
update_interval	The traffic manager will periodically check the status of the cloud through an API call. This setting specifies the interval between such updates. • Type: UInt • Default value: "30"

Custom configuration set

URI Path: custom

Custom configuration sets store arbitrary named values. These values can be read by SOAP or REST clients.

Property	Description
string_lists	This table contains named lists of strings • Primary key: – name (String): Name of list • Sub keys: – value (List(String)): Named list of user-specified strings.

DNS Zone

URI Path: dns_server/zones

The conf/dnsserver/zones/ file contains zone metadata

Property	Description
origin	The domain origin of this Zone. • Type: String • Default value: <none>
zonefile	The Zone File encapsulated by this Zone. • Type: String • Default value: <none>

DNS Zone File

URI Path: dns_server/zone_files

The conf/dnsserver/zonefiles/ directory contains files that define DNS zones.

Property	Description
There are no properties to display for this resource.	

Event Type

URI Path: event_types

Configuration that ties actions to a set of events that trigger them.

Property	Description
actions	The actions triggered by events matching this event type, as a list of action references. • Type: List(Reference(config-event-action)) • Default value: <none>
built_in	If set to Yes this indicates that this configuration is built-in (provided as part of the software) and must not be deleted or edited. • Type: Boolean • Default value: false
note	A description of this event type. • Type: FreeformString • Default value: <none>
Properties for the "cloudcredentials" section:	
event_tags	Cloud credentials event tags • Type: List(String) • Default value: <none>
objects	Cloud credentials object names • Type: List(String) • Default value: <none>
Properties for the "config" section:	
event_tags	Configuration file event tags • Type: List(String) • Default value: <none>
Properties for the "faulttolerance" section:	
event_tags	Fault tolerance event tags • Type: List(String) • Default value: <none>
Properties for the "general" section:	
event_tags	General event tags • Type: List(String) • Default value: <none>
Properties for the "glb" section:	
event_tags	GLB service event tags • Type: List(String) • Default value: <none>
objects	GLB service object names • Type: List(String) • Default value: <none>
Properties for the "java" section:	

Property	Description
event_tags	Java event tags • Type: List(String) • Default value: <none>
Properties for the "licensekeys" section:	
event_tags	License key event tags • Type: List(String) • Default value: <none>
objects	License key object names • Type: List(String) • Default value: <none>
Properties for the "locations" section:	
event_tags	Location event tags • Type: List(String) • Default value: <none>
objects	Location object names • Type: List(String) • Default value: <none>
Properties for the "monitors" section:	
event_tags	Monitor event tags • Type: List(String) • Default value: <none>
objects	Monitors object names • Type: List(String) • Default value: <none>
Properties for the "pools" section:	
event_tags	Pool key event tags • Type: List(String) • Default value: <none>
objects	Pool object names • Type: List(String) • Default value: <none>
Properties for the "protection" section:	
event_tags	Service protection class event tags • Type: List(String) • Default value: <none>
objects	Service protection class object names • Type: List(String) • Default value: <none>

Property	Description
Properties for the "rules" section:	
event_tags	Rule event tags • Type: List(String) • Default value: <none>
objects	Rule object names • Type: List(String) • Default value: <none>
Properties for the "slm" section:	
event_tags	SLM class event tags • Type: List(String) • Default value: <none>
objects	SLM class object names • Type: List(String) • Default value: <none>
Properties for the "ssl" section:	
event_tags	SSL event tags • Type: List(String) • Default value: <none>
Properties for the "sslhw" section:	
event_tags	SSL hardware event tags • Type: List(String) • Default value: <none>
Properties for the "trafficscript" section:	
event_tags	TrafficScript event tags • Type: List(String) • Default value: <none>
Properties for the "vservers" section:	
event_tags	Virtual server event tags • Type: List(String) • Default value: <none>
objects	Virtual server object names • Type: List(String) • Default value: <none>
Properties for the "zxtms" section:	

Property	Description
event_tags	Traffic manager event tags • Type: List(String) • Default value: <none>
objects	Traffic manager object names • Type: List(String) • Default value: <none>

Extra File

URI Path: extra_files

A user-uploaded file. Such files can be used in TrafficScript code using the resource.get function.

Property	Description
There are no properties to display for this resource.	

GLB Service

URI Path: glb_services

A global load balancing service is used by a virtual server to modify DNS requests in order load balance data across different GLB locations.

Property	Description
algorithm	Defines the global load balancing algorithm to be used. • Type: Enum(String) • Default value: "hybrid" • Permitted values: "chained": Sends traffic to one location at a time, until that location fails where the next one in the chain is used. "geo": Distributes traffic based solely on the geographic location of each client. "hybrid": Distribute traffic based on both the load and geographic location. "load": Distributes traffic based on the current load to each location. "round_robin": Distributes traffic by assigning each request to a new location in turn. Over a period of time, all locations will receive the same number of requests. "weighted_random": Distributes traffic in a random way, but according to a weighted policy defined by individual location weights
all_monitors_needed	Are all the monitors required to be working in a location to mark this service as alive? • Type: Boolean • Default value: false

Property	Description
autorecovery	The last location to fail will be available as soon as it recovers. • Type: Boolean • Default value: false
chained_auto_failback	Enable/Disable automatic failback mode. • Type: Boolean • Default value: false
chained_location_order	The locations this service operates for and defines the order in which locations fail. • Type: List(String) • Default value: <none>
disable_on_failure	Locations recovering from a failure will become disabled. • Type: Boolean • Default value: false
dnssec_keys	A table mapping domains to the private keys that authenticate them • Primary key: – domain (String): A domain authenticated by the associated private keys. • Sub keys: – ssl_key (Set(String)): Private keys that authenticate the associated domain.
domains	The domains shown here should be a list of Fully Qualified Domain Names that you would like to balance globally. Responses from the back end DNS servers for queries that do not match this list will be forwarded to the client unmodified. Note: "*" may be used as a wild card. • Type: Set(String) • Default value: <none>
enabled	Enable/Disable our response manipulation of DNS. • Type: Boolean • Default value: false
geo_effect	How much should the locality of visitors affect the choice of location used? This value is a percentage, 0% means that no locality information will be used, and 100% means that locality will always control which location is used. Values between the two extremes will act accordingly. • Type: UInt • Default value: "50"
last_resort_response	The response to be sent in case there are no locations available. • Type: Set(String) • Default value: <none>
location_draining	This is the list of locations for which this service is draining. A location that is draining will never serve any of its service IP addresses for this domain. This can be used to take a location off-line. • Type: Set(String) • Default value: <none>

Property	Description
location_settings	Table containing location specific settings. • Primary key: – location (String): Location to which the associated settings apply. • Sub keys: – weight (UInt): Weight for this location, for use by the weighted random algorithm. – ips (Set(String)): The IP addresses that are present in a location. If the Global Load Balancer decides to direct a DNS query to this location, then it will filter out all IPs that are not in this list. – monitors (Set(String)): The monitors that are present in a location.
return_ips_on_fail	Return all or none of the IPs under complete failure. • Type: Boolean • Default value: false
rules	Response rules to be applied in the context of the service, in order, comma separated. • Type: List(Reference(config-trafficscript)) • Default value: <none>
ttl	The TTL for the DNS resource records handled by the GLB service. • Type: Int • Default value: "-1"
Properties for the "log" section:	
enabled	Log connections to this GLB service? • Type: Boolean • Default value: false
filename	The filename the verbose query information should be logged to. Appliances will ignore this. • Type: String • Default value: "%zeushome%/zxtm/log/services/%g.log"
format	The format of the log lines. • Type: String • Default value: "%t, %s, %l, %q, %g, %n, %d, %a"

Global Settings

URI Path: global_settings

General settings that apply to every machine in the cluster.

Property	Description
accepting_delay	How often, in milliseconds, each traffic manager child process (that isn't listening for new connections) checks to see whether it should start listening for new connections. • Type: UInt • Default value: "50"
afm_enabled	Is the application firewall enabled. • Type: Boolean • Default value: false
chunk_size	The default chunk size for reading/writing requests. • Type: UInt • Default value: "16384"
client_first_opt	Whether or not your traffic manager should make use of TCP optimisations to defer the processing of new client-first connections until the client has sent some data. • Type: Boolean • Default value: false
cluster_identifier	Cluster identifier. Generally supplied by Services Director. • Type: String • Default value: <none>
data_plane_acceleration_cores	The number of CPU cores assigned to assist with data plane acceleration. These cores are dedicated to reading and writing packets to the network interface cards and distributing packets between the traffic manager processes. • Type: Enum(UInt) • Default value: "one" • Permitted values: "four": 4 "one": 1 "two": 2
data_plane_acceleration_mode	Whether Data Plane Acceleration Mode is enabled. • Type: Boolean • Default value: false
license_servers	A list of license servers for FLA licensing. A license server should be specified as a <ip/host>:<port> pair. • Type: Set(String) • Default value: <none>
max_fds	The maximum number of file descriptors that your traffic manager will allocate. • Type: UInt • Default value: "1048576"

Property	Description
monitor_memory_size	The maximum number of each of nodes, pools or locations that can be monitored. The memory used to store information about nodes, pools and locations is allocated at start-up, so the traffic manager must be restarted after changing this setting. • Type: UInt • Default value: "4096"
rate_class_limit	The maximum number of Rate classes that can be created. Approximately 100 bytes will be pre-allocated per Rate class. • Type: UInt • Default value: "25000"
shared_pool_size	The size of the shared memory pool used for shared storage across worker processes (e.g. bandwidth shared data).This is specified as either a percentage of system RAM, 5% for example, or an absolute size such as 10MB. • Type: String • Default value: "10MB"
slm_class_limit	The maximum number of SLM classes that can be created. Approximately 100 bytes will be pre-allocated per SLM class. • Type: UInt • Default value: "1024"
so_rbuff_size	The size of the operating system's read buffer. A value of 0 (zero) means to use the OS default; in normal circumstances this is what should be used. • Type: UInt • Default value: <none>
so_wbuff_size	The size of the operating system's write buffer. A value of 0 (zero) means to use the OS default; in normal circumstances this is what should be used. • Type: UInt • Default value: <none>
socket_optimizations	Whether or not the traffic manager should use potential network socket optimisations. If set to auto, a decision will be made based on the host platform. • Type: Enum(String) • Default value: "auto" • Permitted values: "auto": Decide based on local platform "no": Disable socket optimizations "yes": Enable socket optimizations
tip_class_limit	The maximum number of Traffic IP Groups that can be created. • Type: UInt • Default value: "10000"
Properties for the "admin" section:	

Property	Description
honor_fallback_scsv	Whether or not the admin server, the internal control port and the config daemon honor the Fallback SCSV to protect connections against downgrade attacks. • Type: Boolean • Default value: false
ssl3_allow_rehandshake	Whether or not SSL3/TLS re-handshakes should be supported for admin server and internal connections. • Type: Enum(String) • Default value: "rfc5746" • Permitted values: "always": Always allow "never": Never allow "rfc5746": Only if client uses RFC 5746 (Secure Renegotiation Extension) "safe": Allow safe re-handshakes
ssl3_ciphers	The SSL ciphers to use for admin server and internal connections. For information on supported ciphers see the online help. • Type: String • Default value: "SSL_RSA_WITH_AES_128_GCM_SHA256,SSL_RSA_WITH_AES_128_CBC_SHA256,SSL_RSA_WITH_AES_128_CBC_SHA,SSL_RSA_WITH_AES_256_GCM_SHA384,SSL_RSA_WITH_AES_256_CBC_SHA256,SSL_RSA_WITH_AES_256_CBC_SHA,SSL_RSA_WITH_3DES_EDE_CBC_SHA,SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,SSL_DHE_DSS_WITH_AES_128_CBC_SHA,SSL_DHE_DSS_WITH_AES_256_CBC_SHA,SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA"
ssl3_diffie_hellman_key_length	The length in bits of the Diffie-Hellman key for ciphers that use Diffie-Hellman key agreement for admin server and internal connections. • Type: Enum(UInt) • Default value: "dh_2048" • Permitted values: "dh_1024": Use 1024 bit keys for Diffie-Hellman ciphers. "dh_2048": Use 2048 bit keys for Diffie-Hellman ciphers. "dh_3072": Use 3072 bit keys for Diffie-Hellman ciphers. "dh_4096": Use 4096 bit keys for Diffie-Hellman ciphers.
ssl3_min_rehandshake_interval	If SSL3/TLS re-handshakes are supported on the admin server, this defines the minimum time interval (in milliseconds) between handshakes on a single SSL3/TLS connection that is permitted. To disable the minimum interval for handshakes the key should be set to the value 0. • Type: UInt • Default value: "1000"
ssl_elliptic_curves	The SSL elliptic curve preference list for admin and internal connections. The named curves P256, P384 and P521 may be configured. • Type: List(String) • Default value: <none>

Property	Description
ssl_insert_extra_fragment	Whether or not SSL3 and TLS1 use one-byte fragments as a BEAST countermeasure for admin server and internal connections. • Type: Boolean • Default value: false
ssl_max_handshake_message_size	The maximum size (in bytes) of SSL handshake messages that the admin server and internal connections will accept. To accept any size of handshake message the key should be set to the value 0. • Type: UInt • Default value: "10240"
ssl_prevent_timing_side_channels	Take performance degrading steps to prevent exposing timing side-channels with SSL3 and TLS used by the admin server and internal connections. • Type: Boolean • Default value: false
ssl_signature_algorithms	The SSL signature algorithms preference list for admin and internal connections using TLS version 1.2 or higher. For information on supported algorithms see the online help. • Type: String • Default value: <none>
support_ssl2	No longer supported. Formerly controlled whether SSLv2 could be used for connections to the Administration Server. • Type: Boolean • Default value: false
support_ssl3	Whether or not SSL3 support is enabled for admin server and internal connections. • Type: Boolean • Default value: false
support_tls1	Whether or not TLS1.0 support is enabled for admin server and internal connections. • Type: Boolean • Default value: false
support_tls11	Whether or not TLS1.1 support is enabled for admin server and internal connections. • Type: Boolean • Default value: false
support_tls12	Whether or not TLS1.2 support is enabled for admin server and internal connections. • Type: Boolean • Default value: false
Properties for the "appliance" section:	
bootloader_password	The password used to protect the bootloader. An empty string means there will be no protection. • Type: Password • Default value: <none>

Property	Description
manage_ncipher	Whether or not we should manage the nCipher Support Software automatically. • Type: Boolean • Default value: false
nethsm_esn	The ESN (electronic serial number) for the NetHSM. • Type: String • Default value: <none>
nethsm_hash	The key hash for the NetHSM. • Type: String • Default value: <none>
nethsm_ip	The IP address of the nCipher NetHSM to use. • Type: String • Default value: <none>
nethsm_ncipher_rfs	The IP address of the nCipher Remote File System to use. • Type: String • Default value: <none>
return_path_routing_enabled	Whether or not the traffic manager will attempt to route response packets back to clients via the same route on which the corresponding request arrived. Note that this applies only to the last hop of the route - the behaviour of upstream routers cannot be altered by the traffic manager. • Type: Boolean • Default value: false
Properties for the "aptimizer" section:	
max_dependent_fetch_size	The maximum size of a dependent resource that can undergo Web Accelerator optimization. Any content larger than this size will not be optimized. Units of KB and MB can be used, no postfix denotes bytes. A value of 0 disables the limit. • Type: String • Default value: "2MB"
max_original_content_buffer_size	The maximum size of unoptimized content buffered in the traffic manager for a single backend response that is undergoing Web Accelerator optimization. Responses larger than this will not be optimized. Note that if the backend response is compressed then this setting pertains to the compressed size, before Web Accelerator decompresses it. Units of KB and MB can be used, no postfix denotes bytes. Value range is 1 - 128MB. • Type: String • Default value: "2MB"
watchdog_interval	The period of time (in seconds) after which a previous failure will no longer count towards the watchdog limit. • Type: UInt • Default value: "300"

Property	Description
watchdog_limit	The maximum number of times the Web Accelerator sub-process will be started or restarted within the interval defined by the aptimizer!watchdog_interval setting. If the process fails this many times, it must be restarted manually from the Diagnose page. Zero means no limit. • Type: UInt • Default value: "3"
Properties for the "auditlog" section:	
via_eventd	Whether to mirror the audit log to EventD. • Type: Boolean • Default value: false
via_syslog	Whether to output audit log message to the syslog. • Type: Boolean • Default value: false
Properties for the "autoscaler" section:	
verbose	Whether or not detailed messages about the autoscaler's activity are written to the error log. • Type: Boolean • Default value: false
Properties for the "bandwidth" section:	
Properties for the "bgp" section:	
as_number	The number of the BGP AS in which the traffic manager will operate. Must be entered in decimal. • Type: UInt • Default value: "65534"
enabled	Whether BGP Route Health Injection is enabled • Type: Boolean • Default value: false
Properties for the "cluster_comms" section:	
allow_update_default	The default value of allow_update for new cluster members. If you have cluster members joining from less trusted locations (such as cloud instances) this can be set to false in order to make them effectively "read-only" cluster members. • Type: Boolean • Default value: false
allowed_update_hosts	The hosts that can contact the internal administration port on each traffic manager. This should be a list containing IP addresses, CIDR IP subnets, and localhost; or it can be set to all to allow any host to connect. • Type: List(String) • Default value: "all"

Property	Description
state_sync_interval	How often to propagate the session persistence and bandwidth information to other traffic managers in the same cluster. Set this to 0 (zero) to disable propagation. Note that a cluster using "unicast" heartbeat messages cannot turn off these messages. • Type: UInt • Default value: "3"
state_sync_timeout	The maximum amount of time to wait when propagating session persistence and bandwidth information to other traffic managers in the same cluster. Once this timeout is hit the transfer is aborted and a new connection created. • Type: UInt • Default value: "6"
Properties for the "connection" section:	
idle_connections_max	The maximum number of unused HTTP keepalive connections with back-end nodes that the traffic manager should maintain for re-use. Setting this to 0 (zero) will cause the traffic manager to auto-size this parameter based on the available number of file-descriptors. • Type: UInt • Default value: <none>
idle_timeout	How long an unused HTTP keepalive connection should be kept before it is discarded. • Type: UInt • Default value: "10"
listen_queue_size	The listen queue size for managing incoming connections. It may be necessary to increase the system's listen queue size if this value is altered. If the value is set to 0 then the default system setting will be used. • Type: UInt • Default value: <none>
max_accepting	Number of processes that should accept new connections. Only this many traffic manager child processes will listen for new connections at any one time. Setting this to 0 (zero) will cause your traffic manager to select an appropriate default value based on the architecture and number of CPUs. • Type: UInt • Default value: <none>
multiple_accept	Whether or not the traffic manager should try to read multiple new connections each time a new client connects. This can improve performance under some very specific conditions. However, in general it is recommended that this be set to 'false'. • Type: Boolean • Default value: false
Properties for the "data_plane_acceleration" section:	

Property	Description
tcp_delay_ack	The time, in milliseconds, to delay sending a TCP ACK response, providing an opportunity for additional data to be incorporated into the response and potentially improving network performance. The setting affects TCP connections handled by layer 7 services running in Data Plane Acceleration mode. • Type: UInt • Default value: "200"
tcp_win_scale	The TCP window scale option, which configures the size of the receive window for TCP connections handled by layer 7 services when running in Data Plane Acceleration mode. • Type: UInt • Default value: "7"
Properties for the "dns" section:	
max_ttl	Maximum Time To Live (expiry time) for entries in the DNS cache. • Type: UInt • Default value: "86400"
min_ttl	Minimum Time To Live (expiry time) for entries in the DNS cache. • Type: UInt • Default value: "86400"
negative_expiry	Expiry time for failed lookups in the DNS cache. • Type: UInt • Default value: "60"
size	Maximum number of entries in the DNS cache. • Type: UInt • Default value: "10867"
timeout	Timeout for receiving a response from a DNS server. • Type: UInt • Default value: "12"
Properties for the "dns_autoscale" section:	
Properties for the "ec2" section:	
access_key_id	Amazon EC2 Access Key ID. • Type: String • Default value: <none>
awstool_timeout	The maximum amount of time requests to the AWS Query API can take before timing out. • Type: UInt • Default value: "10"
secret_access_key	Amazon EC2 Secret Access Key. • Type: Password • Default value: <none>

Property	Description
verify_query_server_cert	Whether to verify Amazon EC2 endpoint's certificate using CA(s) present in SSL Certificate Authorities Catalog. • Type: Boolean • Default value: false
Properties for the "eventing" section:	
mail_interval	The minimum length of time that must elapse between alert emails being sent. Where multiple alerts occur inside this timeframe, they will be retained and sent within a single email rather than separately. • Type: UInt • Default value: "30"
max_attempts	The number of times to attempt to send an alert email before giving up. • Type: UInt • Default value: "10"
Properties for the "fault_tolerance" section:	
arp_count	The number of ARP packets a traffic manager should send when an IP address is raised. • Type: UInt • Default value: "10"
auto_failback	Whether or not traffic IPs automatically move back to machines that have recovered from a failure and have dropped their traffic IPs. • Type: Boolean • Default value: false
autofailback_delay	Configure the delay of automatic failback after a previous failover event. This setting has no effect if autofailback is disabled. • Type: UInt • Default value: "10"
child_timeout	How long the traffic manager should wait for status updates from any of the traffic manager's child processes before assuming one of them is no longer servicing traffic. • Type: UInt • Default value: "5"
frontend_check_ips	The IP addresses used to check front-end connectivity. The text %gateway% will be replaced with the default gateway on each system. Set this to an empty string if the traffic manager is on an Intranet with no external connectivity. • Type: Set(String) • Default value: "%gateway%"
heartbeat_method	The method traffic managers should use to exchange cluster heartbeat messages. • Type: Enum(String) • Default value: "unicast" • Permitted values: "multicast": multicast "unicast": unicast

Property	Description
igmp_interval	The interval between unsolicited periodic IGMP Membership Report messages for Multi-Hosted Traffic IP Groups. • Type: UInt • Default value: "30"
l4accel_child_timeout	When running in Data Plane Acceleration Mode, how long the traffic manager should wait for a status update from child processes handling L4Accel services before assuming it is no longer servicing traffic. • Type: UInt • Default value: "2"
l4accel_sync_port	The port on which cluster members will transfer state information for L4Accel services when running in Data Plane Acceleration Mode. • Type: UInt • Default value: "10240"
monitor_interval	The frequency, in milliseconds, that each traffic manager machine should check and announce its connectivity. • Type: UInt • Default value: "500"
monitor_timeout	How long, in seconds, each traffic manager should wait for a response from its connectivity tests or from other traffic manager machines before registering a failure. • Type: UInt • Default value: "5"
multicast_address	The multicast address and port to use to exchange cluster heartbeat messages. • Type: String • Default value: "239.100.1.1:9090"
unicast_port	The unicast UDP port to use to exchange cluster heartbeat messages. • Type: UInt • Default value: "9090"
use_bind_ip	Whether or not cluster heartbeat messages should only be sent and received over the management network. • Type: Boolean • Default value: false
verbose	Whether or not a traffic manager should log all connectivity tests. This is very verbose, and should only be used for diagnostic purposes. • Type: Boolean • Default value: false
Properties for the "fips" section:	
enabled	Enable FIPS Mode (requires software restart). • Type: Boolean • Default value: false
Properties for the "ftp" section:	

Property	Description
data_bind_low	Whether or not the traffic manager should permit use of FTP data connection source ports lower than 1024. If No the traffic manager can completely drop root privileges, if Yes some or all privileges may be retained in order to bind to low ports. • Type: Boolean • Default value: false
Properties for the "glb" section:	
verbose	Write a message to the logs for every DNS query that is load balanced, showing the source IP address and the chosen datacenter. • Type: Boolean • Default value: false
Properties for the "historical_activity" section:	
keep_days	Number of days to store historical traffic information, if set to 0 the data will be kept indefinitely. • Type: UInt • Default value: "90"
Properties for the "http" section:	
Properties for the "ip" section:	
appliance_returnpath	A table of MAC to IP address mappings for each router where return path routing is required. • Primary key: – mac (String): The MAC address of a router the software is connected to. • Sub keys: – ipv4 (String): The MAC address to IPv4 address mapping of a router the software is connected to. The * (asterisk) in the key name is the MAC address, the value is the IP address. – ipv6 (String): The MAC address to IPv6 address mapping of a router the software is connected to. The * (asterisk) in the key name is the MAC address, the value is the IP address.
Properties for the "java" section:	
classpath	CLASSPATH to use when starting the Java runner. • Type: String • Default value: <none>
command	Java command to use when starting the Java runner, including any additional options. • Type: String • Default value: "java -server"
enabled	Whether or not Java support should be enabled. If this is set to No, then your traffic manager will not start any Java processes. Java support is only required if you are using the TrafficScript java.run() function. • Type: Boolean • Default value: false

Property	Description
lib	Java library directory for additional jar files. The Java runner will load classes from any .jar files stored in this directory, as well as the * jar files and classes stored in traffic manager's catalog. • Type: String • Default value: <none>
max_connections	Maximum number of simultaneous Java requests. If there are more than this many requests, then further requests will be queued until the earlier requests are completed. This setting is per-CPU, so if your traffic manager is running on a machine with 4 CPU cores, then each core can make this many requests at one time. • Type: UInt • Default value: "256"
session_age	Default time to keep a Java session. • Type: UInt • Default value: "86400"
Properties for the "kerberos" section:	
verbose	Whether or not a traffic manager should log all Kerberos related activity. This is very verbose, and should only be used for diagnostic purposes. • Type: Boolean • Default value: false
Properties for the "l4accel" section:	
max_concurrent_connections	The maximum number of concurrent connections, in millions, that can be handled by each L4Accel child process. An appropriate amount of memory to store this many connections will be allocated when the traffic manager starts. • Type: UInt • Default value: "1"
Properties for the "log" section:	
error_level	The minimum severity of events/alerts that should be logged to disk. INFO will log all events; a higher severity setting will log fewer events. More fine-grained control can be achieved using events and actions. • Type: Enum(UInt) • Default value: "info" • Permitted values: "fatal": Only fatal errors are logged "info": All events are logged to disk "serious": Only serious errors or worse "warn": Only warnings and errors are logged
flush_time	How long to wait before flushing the request log files for each virtual server. • Type: UInt • Default value: "5"
log_file	The file to log event messages to. • Type: String • Default value: "%zeushome%/zxtm/log/errors"

Property	Description
rate	The maximum number of connection errors logged per second when connection error reporting is enabled. • Type: UInt • Default value: "50"
reopen	How long to wait before re-opening request log files, this ensures that log files will be recreated in the case of log rotation. • Type: UInt • Default value: "30"
time	The minimum time between log messages for log intensive features such as SLM. • Type: UInt • Default value: "60"
Properties for the "log_export" section:	
auth_hec_token	The HTTP Event Collector token to use for HTTP authentication with a Splunk server. • Type: String • Default value: <none>
auth_http	The HTTP authentication method to use when exporting log entries. • Type: Enum(String) • Default value: "none" • Permitted values: "basic": Basic (Username and Password) "none": None "splunk": Splunk (HEC token)
auth_password	The password to use for HTTP basic authentication. • Type: Password • Default value: <none>
auth_username	The username to use for HTTP basic authentication. • Type: String • Default value: <none>
enabled	Monitor log files and export entries to the configured endpoint. • Type: Boolean • Default value: false
endpoint	The URL to which log entries should be sent. Entries are sent using HTTP(S) POST requests. • Type: String • Default value: <none>

Property	Description
request_timeout	The number of seconds after which HTTP requests sent to the configured endpoint will be considered to have failed if no response is received. A value of 0 means that HTTP requests will not time out. • Type: UInt • Default value: "30"
tls_verify	Whether the server certificate should be verified when connecting to the endpoint. If enabled, server certificates that do not match the server name, are self-signed, have expired, have been revoked, or that are signed by an unknown CA will be rejected. • Type: Boolean • Default value: false
Properties for the "ospfv2" section:	
area	The OSPF area in which the traffic manager will operate. May be entered in decimal or IPv4 address format. • Type: String • Default value: "0.0.0.1"
area_type	The type of OSPF area in which the traffic manager will operate. This must be the same for all routers in the area, as required by OSPF. • Type: Enum(String) • Default value: "normal" • Permitted values: "normal": Normal area "nssa": Not So Stubby Area (RFC3101) "stub": Stub area
authentication_key_id_a	OSPFv2 authentication key ID. If set to 0, which is the default value, the key is disabled. • Type: UInt • Default value: <none>
authentication_key_id_b	OSPFv2 authentication key ID. If set to 0, which is the default value, the key is disabled. • Type: UInt • Default value: <none>
authentication_shared_secret_a	OSPFv2 authentication shared secret (MD5). If set to blank, which is the default value, the key is disabled. • Type: String • Default value: <none>
authentication_shared_secret_b	OSPFv2 authentication shared secret (MD5). If set to blank, which is the default value, the key is disabled. • Type: String • Default value: <none>
enabled	Whether OSPFv2 Route Health Injection is enabled • Type: Boolean • Default value: false

Property	Description
hello_interval	The interval at which OSPF "hello" packets are sent to the network. • Type: UInt • Default value: "10"
router_dead_interval	The number of seconds before declaring a silent router down. • Type: UInt • Default value: "40"
Properties for the "periodic_log" section:	
Properties for the "protection" section:	
conncount_size	The amount of shared memory reserved for an inter-process table of combined connection counts, used by all Service Protection classes that have per_process_connection_count set to No. The amount is specified as an absolute size, eg 20MB. • Type: String • Default value: "20MB"
Properties for the "recent_connections" section:	
max_per_process	How many recently closed connections each traffic manager process should save. These saved connections will be shown alongside currently active connections when viewing the Connections page. You should set this value to 0 in a benchmarking or performance-critical environment. • Type: UInt • Default value: "500"
retain_time	The amount of time for which snapshots will be retained on the Connections page. • Type: UInt • Default value: "60"
snapshot_size	The maximum number of connections each traffic manager process should show when viewing a snapshot on the Connections page. This value includes both currently active connections and saved connections. If set to 0 all active and saved connection will be displayed on the Connections page. • Type: UInt • Default value: "500"
Properties for the "remote_licensing" section:	
owner	The Owner of a Services Director instance, used for self-registration. • Type: String • Default value: <none>
owner_secret	The secret associated with the Owner. • Type: String • Default value: <none>
policy_id	The auto-accept Policy ID that this instance should attempt to use. • Type: String • Default value: <none>

Property	Description
registration_server	A Services Director address for self-registration. A registration server should be specified as a <ip/host>:<port> pair. • Type: String • Default value: <none>
server_certificate	The certificate of a Services Director instance, used for self-registration. • Type: String • Default value: <none>
Properties for the "rest_api" section:	
auth_timeout	The length of time after a successful request that the authentication of a given username and password will be cached for an IP address. A setting of 0 disables the cache forcing every REST request to be authenticated which will adversely affect performance. • Type: UInt • Default value: "120"
enabled	Whether or not the REST service is enabled. • Type: Boolean • Default value: false
http_max_header_length	The maximum allowed length in bytes of a HTTP request's headers. • Type: UInt • Default value: "4096"
replicate_absolute	Configuration changes will be replicated across the cluster after this period of time, regardless of whether additional API requests are being made. • Type: UInt • Default value: "20"
replicate_lull	Configuration changes made via the REST API will be propagated across the cluster when no further API requests have been made for this period of time. • Type: UInt • Default value: "5"
replicate_timeout	The period of time after which configuration replication across the cluster will be cancelled if it has not completed. • Type: UInt • Default value: "10"
Properties for the "security" section:	
login_banner	Banner text displayed on the Admin Server login page and before logging in to appliance SSH servers. • Type: FreeformString • Default value: <none>
login_banner_accept	Whether or not users must explicitly agree to the displayed login_banner text before logging in to the Admin Server. • Type: Boolean • Default value: false

Property	Description
login_delay	The number of seconds before another login attempt can be made after a failed attempt. • Type: UInt • Default value: "4"
max_login_attempts	The number of sequential failed login attempts that will cause a user account to be suspended. Setting this to 0 disables this feature. To apply this to users who have never successfully logged in, track_unknown_users must also be enabled. • Type: UInt • Default value: <none>
max_login_external	Whether or not usernames blocked due to the max_login_attempts limit should also be blocked from authentication against external services (such as LDAP and RADIUS). • Type: Boolean • Default value: false
max_login_suspension_time	The number of minutes to suspend users who have exceeded the max_login_attempts limit. • Type: UInt • Default value: "15"
password_allow_consecutive_chars	Whether or not to allow the same character to appear consecutively in passwords. • Type: Boolean • Default value: false
password_changes_per_day	The maximum number of times a password can be changed in a 24-hour period. Set to 0 to disable this restriction. • Type: UInt • Default value: <none>
password_min_alpha_chars	Minimum number of alphabetic characters a password must contain. Set to 0 to disable this restriction. • Type: UInt • Default value: <none>
password_min_length	Minimum number of characters a password must contain. Set to 0 to disable this restriction. • Type: UInt • Default value: <none>
password_min_numeric_chars	Minimum number of numeric characters a password must contain. Set to 0 to disable this restriction. • Type: UInt • Default value: <none>
password_min_special_chars	Minimum number of special (non-alphanumeric) characters a password must contain. Set to 0 to disable this restriction. • Type: UInt • Default value: <none>

Property	Description
password_min_uppercase_chars	Minimum number of uppercase characters a password must contain. Set to 0 to disable this restriction. • Type: UInt • Default value: <none>
password_reuse_after	The number of times a password must have been changed before it can be reused. Set to 0 to disable this restriction. • Type: UInt • Default value: <none>
post_login_banner	Banner text to be displayed on the appliance console after login. • Type: String • Default value: <none>
track_unknown_users	Whether to remember past login attempts from usernames that are not known to exist (should be set to false for an Admin Server accessible from the public Internet). This does not affect the audit log. • Type: Boolean • Default value: false
ui_page_banner	Banner text to be displayed on all Admin Server pages. • Type: String • Default value: <none>
Properties for the "session" section:	
asp_cache_size	The maximum number of entries in the ASP session cache. This is used for storing session mappings for ASP session persistence. Approximately 100 bytes will be pre-allocated per entry. • Type: UInt • Default value: "32768"
ip_cache_size	The maximum number of entries in the IP session cache. This is used to provide session persistence based on the source IP address. Approximately 100 bytes will be pre-allocated per entry. • Type: UInt • Default value: "32768"
j2ee_cache_size	The maximum number of entries in the J2EE session cache. This is used for storing session mappings for J2EE session persistence. Approximately 100 bytes will be pre-allocated per entry. • Type: UInt • Default value: "32768"
ssl_cache_size	The maximum number of entries in the SSL session persistence cache. This is used to provide session persistence based on the SSL session ID. Approximately 200 bytes will be pre-allocated per entry. • Type: UInt • Default value: "32768"

Property	Description
universal_cache_size	The maximum number of entries in the global universal session cache. This is used for storing session mappings for universal session persistence. Approximately 100 bytes will be pre-allocated per entry. • Type: UInt • Default value: "32768"
Properties for the "snmp" section:	
user_counters	The number of user defined SNMP counters. Approximately 100 bytes will be pre-allocated at start-up per user defined SNMP counter. • Type: UInt • Default value: "10"
Properties for the "soap" section:	
idle_minutes	The number of minutes that the SOAP server should remain idle before exiting. The SOAP server has a short startup delay the first time a SOAP request is made, subsequent SOAP requests don't have this delay. • Type: UInt • Default value: "10"
Properties for the "source_nat" section:	
ip_limit	The maximum number of Source NAT IP addresses that can be used across all Traffic IP Groups. • Type: UInt • Default value: "16"
ip_local_port_range_high	The upper boundary of the port range reserved for use by the kernel. Ports above this range will be used by the traffic manager for establishing outgoing connections. • Type: UInt • Default value: "10240"
shared_pool_size	The size of the Source NAT shared memory pool used for shared storage across child processes. This value is specified as an absolute size such as 10MB. • Type: UInt • Default value: "10"
Properties for the "ssl" section:	
cache_expiry	How long the SSL session IDs for SSL decryption should be stored for. • Type: UInt • Default value: "1800"
cache_per_virtualserver	Whether an SSL session created by a given virtual server can only be resumed by a connection to the same virtual server. • Type: Boolean • Default value: false

Property	Description
cache_size	How many entries the SSL session ID cache should hold. This cache is used to cache SSL sessions to help speed up SSL handshakes when performing SSL decryption. To turn off SSL session resumption, set this key to the value 0. Each entry will allocate approximately 1.5kB of metadata. • Type: UInt • Default value: "6151"
crl_mem_size	How much shared memory to allocate for loading Certificate Revocation Lists. This should be at least 3 times the total size of all CRLs on disk. This is specified as either a percentage of system RAM, 1% for example, or an absolute size such as 10MB. • Type: String • Default value: "5MB"
elliptic_curves	The SSL elliptic curve preference list for SSL connections using TLS version 1.0 or higher, unless overridden by virtual server or pool settings. The named curves P256, P384 and P521 may be configured. • Type: List(String) • Default value: <none>
honor_fallback_scsv	Whether or not ssl-decrypting Virtual Servers honor the Fallback SCSV to protect connections against downgrade attacks. • Type: Boolean • Default value: false
insert_extra_fragment	Whether or not SSL3 and TLS1 use one-byte fragments as a BEAST countermeasure. • Type: Boolean • Default value: false
max_handshake_message_size	The maximum size (in bytes) of SSL handshake messages that SSL connections will accept. To accept any size of handshake message the key should be set to the value 0. • Type: UInt • Default value: "10240"
ocsp_cache_size	The maximum number of cached client certificate OCSP results stored. This cache is used to speed up OCSP checks against client certificates by caching results. Approximately 1040 bytes are pre-allocated per entry. • Type: UInt • Default value: "2048"
ocsp_stapling_default_refresh_interval	How long to wait before refreshing requests on behalf of the store of certificate status responses used by OCSP stapling, if we don't have an up-to-date OCSP response. • Type: UInt • Default value: "60"
ocsp_stapling_maximum_refresh_interval	Maximum time to wait before refreshing requests on behalf of the store of certificate status responses used by OCSP stapling. (0 means no maximum.) • Type: UInt • Default value: "864000"

Property	Description
ocsp_stapling_mem_size	How much shared memory to allocate for the store of certificate status responses for OCSP stapling. This should be at least 2kB times the number of certificates configured to use OCSP stapling. This is specified as either a percentage of system RAM, 1% for example, or an absolute size such as 10MB. • Type: String • Default value: "1MB"
ocsp_stapling_time_tolerance	How many seconds to allow the current time to be outside the validity time of an OCSP response before considering it invalid. • Type: UInt • Default value: "30"
ocsp_stapling_verify_response	Whether the OCSP response signature should be verified before the OCSP response is cached. • Type: Boolean • Default value: false
prevent_timing_side_channels	Take performance degrading steps to prevent exposing timing side-channels with SSL3 and TLS. • Type: Boolean • Default value: false
signature_algorithms	The SSL signature algorithms preference list for SSL connections unless overridden by virtual server or pool settings. For information on supported algorithms see the online help. • Type: String • Default value: <none>
ssl3_allow_rehandshake	Whether or not SSL3/TLS re-handshakes should be supported. Enabling support for re-handshakes can expose services to Man-in-the-Middle attacks. It is recommended that only "safe" handshakes be permitted, or none at all. • Type: Enum(String) • Default value: "safe" • Permitted values: "always": Always allow "never": Never allow "rfc5746": Only if client uses RFC 5746 (Secure Renegotiation Extension) "safe": Allow safe re-handshakes
ssl3_ciphers	The SSL ciphers to use. For information on supported ciphers see the online help. • Type: String • Default value: <none>

Property	Description
ssl3_diffie_hellman_key_length	The length in bits of the Diffie-Hellman key for ciphers that use Diffie-Hellman key agreement. • Type: Enum(UInt) • Default value: "dh_2048" • Permitted values: "dh_1024": 1024 "dh_2048": 2048 "dh_3072": 3072 "dh_4096": 4096
ssl3_min_rehandshake_interval	If SSL3/TLS re-handshakes are supported, this defines the minimum time interval (in milliseconds) between handshakes on a single SSL3/TLS connection that is permitted. To disable the minimum interval for handshakes the key should be set to the value 0. • Type: UInt • Default value: "1000"
support_ssl2	No longer supported. Formerly controlled whether SSL2 could be used by default. • Type: Boolean • Default value: false
support_ssl3	Whether or not SSL3 support is enabled. • Type: Boolean • Default value: false
support_tls1	Whether or not TLS1.0 support is enabled. • Type: Boolean • Default value: false
support_tls1_1	Whether or not TLS1.1 support is enabled. • Type: Boolean • Default value: false
support_tls1_2	Whether or not TLS1.2 support is enabled. • Type: Boolean • Default value: false
Properties for the "ssl_hardware" section:	
accel	Whether or not the SSL hardware is an "accelerator" (faster than software). By default the traffic manager will only use the SSL hardware if a key requires it (i.e. the key is stored on secure hardware and the traffic manager only has a placeholder/identifier key). With this option enabled, your traffic manager will instead try to use hardware for all SSL decrypts. • Type: Boolean • Default value: false
azure_client_id	The client identifier used when accessing the Microsoft Azure Key Vault. • Type: String • Default value: <none>

Property	Description
azure_client_secret	The client secret used when accessing the Microsoft Azure Key Vault. • Type: Password • Default value: <none>
azure_vault_url	The URL for the REST API of the Microsoft Azure Key Vault. • Type: String • Default value: <none>
azure_verify_rest_api_cert	Whether or not the Azure Key Vault REST API certificate should be verified. • Type: Boolean • Default value: false
driver_pkcs11_debug	Print verbose information about the PKCS11 hardware security module to the event log. • Type: Boolean • Default value: false
driver_pkcs11_lib	The location of the PKCS#11 library for your SSL hardware if it is not in a standard location. The traffic manager will search the standard locations by default. • Type: String • Default value: <none>
driver_pkcs11_slot_desc	The label of the SSL Hardware slot to use. Only required if you have multiple HW accelerator slots. • Type: String • Default value: <none>
driver_pkcs11_slot_type	The type of SSL hardware slot to use. • Type: Enum(String) • Default value: "operator" • Permitted values: "module": Module Protected "operator": Operator Card Set "softcard": Soft Card
driver_pkcs11_user_pin	The User PIN for the PKCS token (PKCS#11 devices only). • Type: Password • Default value: <none>
failure_count	The number of consecutive failures from the SSL hardware that will be tolerated before the traffic manager assumes its session with the device is invalid and tries to log in again. This is necessary when the device reboots following a power failure. • Type: UInt • Default value: "5"

Property	Description
library	The type of SSL hardware to use. The drivers for the SSL hardware should be installed and accessible to the traffic manager software. • Type: Enum(String) • Default value: "none" • Permitted values: "azure": Microsoft Azure Key Vault "none": None "pkcs11": PKCS#11 (e.g. nCipher NetHSM)
Properties for the "trafficscript" section:	
array_elements	The amount of storage that will be allocated to array elements in TrafficScript. If more elements are required then the necessary memory will be allocated during the execution of the rule. • Type: UInt • Default value: "100000"
data_local_size	The maximum amount of memory available to store TrafficScript data.local.set() information. This can be specified as a percentage of system RAM, 5% for example; or an absolute size such as 200MB. • Type: String • Default value: "5%"
data_size	The maximum amount of memory available to store TrafficScript data.set() information. This can be specified as a percentage of system RAM, 5% for example; or an absolute size such as 200MB. • Type: String • Default value: "5%"
execution_time_warning	Raise an event if a TrafficScript rule runs for more than this number of milliseconds in a single invocation. If you get such events repeatedly, you may want to consider re-working some of your TrafficScript rules. A value of 0 means no warnings will be issued. • Type: UInt • Default value: "500"
max_instr	The maximum number of instructions a TrafficScript rule will run. A rule will be aborted if it runs more than this number of instructions without yielding, preventing infinite loops. • Type: UInt • Default value: "100000"
memory_warning	Raise an event if a TrafficScript rule requires more than this amount of buffered network data. If you get such events repeatedly, you may want to consider re-working some of your TrafficScript rules to use less memory or to stream the data that they process rather than storing it all in memory. This setting also limits the amount of data that can be returned by request.GetLine(). • Type: UInt • Default value: "1048576"

Property	Description
regex_cache_size	The maximum number of regular expressions to cache in TrafficScript. Regular expressions will be compiled in order to speed up their use in the future. • Type: UInt • Default value: "57"
regex_match_limit	The maximum number of ways TrafficScript will attempt to match a regular expression at each position in the subject string, before it aborts the rule and reports a TrafficScript error. • Type: UInt • Default value: "10000000"
regex_match_warn_percentage	The percentage of regex_match_limit at which TrafficScript reports a performance warning. • Type: UInt • Default value: "5"
variable_pool_use	Allow the pool.use and pool.select TrafficScript functions to accept variables instead of requiring literal strings. Enabling this feature has the following effects1. Your traffic manager may no longer be able to know whether a pool is in use.2. Errors for pools that aren't in use will not be hidden.3. Some settings displayed for a Pool may not be appropriate for the type of traffic being managed.4. Pool usage information on the pool edit pages and config summary may not be accurate.5. Monitors will run for all pools (with this option disabled monitors will only run for Pools that are used). • Type: Boolean • Default value: false
Properties for the "transaction_export" section:	
enabled	Export metadata about transactions processed by the traffic manager to an external location. • Type: Boolean • Default value: false
endpoint	The endpoint to which transaction metadata should be exported. The endpoint is specified as a hostname or IP address with a port. • Type: String • Default value: <none>
tls	Whether the connection to the specified endpoint should be encrypted. • Type: Boolean • Default value: false
tls_verify	Whether the server certificate presented by the endpoint should be verified, preventing a connection from being established if the certificate does not match the server name, is self-signed, is expired, is revoked, or has an unknown CA. • Type: Boolean • Default value: false
Properties for the "web_cache" section:	

Property	Description
avg_path_length	The estimated average length of the path (including query string) for resources being cached. An amount of memory equal to this figure multiplied by max_file_num will be allocated for storing the paths for cache entries. This setting can be increased if your web site makes extensive use of long URLs. • Type: UInt • Default value: "512"
disk	Whether or not to use a disk-backed (typically SSD) cache. If set to Yes cached web pages will be stored in a file on disk. This enables the traffic manager to use a cache that is larger than available RAM. The size setting should also be adjusted to select a suitable maximum size based on your disk space. Note that the disk caching is optimized for use with SSD storage. • Type: Boolean • Default value: false
disk_dir	If disk caching is enabled, this sets the directory where the disk cache file will be stored. The traffic manager will create a file called webcache.data in this location. Note that the disk caching is optimized for use with SSD storage. • Type: String • Default value: "%zeushome%/zxtm/internal"
max_file_num	Maximum number of entries in the cache. Approximately 0.9 KB will be pre-allocated per entry for metadata, this is in addition to the memory reserved for the content cache and for storing the paths of the cached resources. • Type: UInt • Default value: "10000"
max_file_size	Largest size of a cacheable object in the cache. This is specified as either a percentage of the total cache size, 2% for example, or an absolute size such as 20MB. • Type: String • Default value: "2%"
max_path_length	The maximum length of the path (including query string) for the resource being cached. If the path exceeds this length then it will not be added to the cache. • Type: UInt • Default value: "2048"
normalize_query	Enable normalization (lexical ordering of the parameter-assignments) of the query string. • Type: Boolean • Default value: false
size	The maximum size of the HTTP web page cache. This is specified as either a percentage of system RAM, 20% for example, or an absolute size such as 200MB. • Type: String • Default value: "20%"
verbose	Add an X-Cache-Info header to every HTTP response, showing whether the request and/or the response was cacheable. • Type: Boolean • Default value: false

Kerberos Configuration File

URI Path: kerberos/krb5confs

A Kerberos krb5.conf file that provides the raw configuration for a Kerberos principal.

Property	Description
There are no properties to display for this resource.	

Kerberos Keytab

URI Path: kerberos/keytabs

A Kerberos keytab file contains credentials to authenticate as (a number of) Kerberos principals.

Property	Description
There are no properties to display for this resource.	

Kerberos Principal

URI Path: kerberos/principals

A Kerberos principal can be used by the traffic manager to participate in a Kerberos realm.

Property	Description
kdcs	A list of <hostname/ip>:<port> pairs for Kerberos key distribution center (KDC) services to be explicitly used for the realm of the principal. If no KDCs are explicitly configured, DNS will be used to discover the KDC(s) to use. • Type: List(String) • Default value: <none>
keytab	The name of the Kerberos keytab file containing suitable credentials to authenticate as the specified Kerberos principal. • Type: String • Default value: <none>
krb5conf	The name of an optional Kerberos configuration file (krb5.conf). • Type: String • Default value: <none>

Property	Description
realm	The Kerberos realm where the principal belongs. • Type: String • Default value: <none>
service	The service name part of the Kerberos principal name the traffic manager should use to authenticate itself. • Type: String • Default value: <none>

License

URI Path: license_keys

A license key is a encoded text file that controls what functionality is available from each traffic manager in the cluster. Every production traffic manager must have a valid licence key in order to function; a traffic manager without a license will operate in developer mode, allowing developers to trial a wide range of functionality, but placing restrictions on bandwidth.

Property	Description
There are no properties to display for this resource.	

Location

URI Path: locations

These are geographic locations as used by Global Load Balancing services. Such a location may not necessarily contain a traffic manager; instead it could refer to the location of a remote datacenter.

Property	Description
id	The identifier of this location. • Type: UInt • Default value: <none>
latitude	The latitude of this location. • Type: Float • Default value: "0.0"
longitude	The longitude of this location. • Type: Float • Default value: "0.0"

Property	Description
note	A note, used to describe this location. • Type: FreeformString • Default value: <none>
type	Does this location contain traffic managers and configuration or is it a recipient of GLB requests? • Type: Enum(String) • Default value: "config" • Permitted values: "config": Configuration "glb": GLB

Log Export

URI Path: log_export

Definitions of log files which should be exported to the analytics engine

Property	Description
appliance_only	Whether entries from the specified log files should be exported only from appliances. • Type: Boolean • Default value: false
enabled	Export entries from the log files included in this category. • Type: Boolean • Default value: false
files	The set of files to export as part of this category, specified as a list of glob patterns. • Type: Set(String) • Default value: <none>
history	How much historic log activity should be exported. • Type: Enum(String) • Default value: "none" • Permitted values: "all": Export all historic entries "none": Do not export any historic entries "recent": Export recent historic entries, according to the 'history_period' setting
history_period	The number of days of historic log entries that should be exported. • Type: UInt • Default value: "10"

Property	Description
metadata	This is table 'metadata' • Primary key: – name (String): The name of a metadata item which should be sent to the analytics engine along with entries from these log files. • Sub keys: – value (String): Additional metadata to include with the log entries when exporting them to the configured endpoint. Metadata can be used by the system that is receiving the exported data to categorise and parse the log entries.
note	A description of this category of log files. • Type: String • Default value: <none>

Monitor

URI Path: monitors

Monitors check important remote services are running, by periodically sending them traffic and checking the response is correct. They are used by virtual servers to detect the failure of backend nodes.

Property	Description
back_off	Should the monitor slowly increase the delay after it has failed? • Type: Boolean • Default value: false
delay	The minimum time between calls to a monitor. • Type: UInt • Default value: "3"
failures	The number of times in a row that a node must fail execution of the monitor before it is classed as unavailable. • Type: UInt • Default value: "3"
health_only	Should this monitor only report health (ignore load)? • Type: Boolean • Default value: false
machine	The machine to monitor, where relevant this should be in the form <hostname>:<port>, for "ping" monitors the :<port> part must not be specified. • Type: String • Default value: <none>
note	A description of the montitor. • Type: FreeformString • Default value: <none>

Property	Description
scope	A monitor can either monitor each node in the pool separately and disable an individual node if it fails, or it can monitor a specific machine and disable the entire pool if that machine fails. GLB location monitors must monitor a specific machine. • Type: Enum(String) • Default value: "pernode" • Permitted values: "pernode": Node: Monitor each node in the pool separately "poolwide": Pool/GLB: Monitor a specified machine
timeout	The maximum runtime for an individual instance of the monitor. • Type: UInt • Default value: "3"
type	The internal monitor implementation of this monitor. • Type: Enum(String) • Default value: "ping" • Permitted values: "connect": TCP Connect monitor "http": HTTP monitor "ping": Ping monitor "program": External program monitor "rtsp": RTSP monitor "sip": SIP monitor "tcp_transaction": TCP transaction monitor
use_ssl	Whether or not the monitor should connect using SSL. • Type: Boolean • Default value: false
verbose	Whether or not the monitor should emit verbose logging. This is useful for diagnosing problems. • Type: Boolean • Default value: false
Properties for the "http" section:	
authentication	The HTTP basic-auth <user>:<password> to use for the test HTTP request. • Type: String • Default value: <none>
body_regex	A regular expression that the HTTP response body must match. If the response body content doesn't matter then set this to .* (match anything). • Type: String • Default value: <none>
host_header	The host header to use in the test HTTP request. • Type: String • Default value: <none>

Property	Description
path	The path to use in the test HTTP request. This must be a string beginning with a / (forward slash). • Type: String • Default value: "/"
status_regex	A regular expression that the HTTP status code must match. If the status code doesn't matter then set this to .* (match anything). • Type: String • Default value: "^[234][0-9][0-9]$"
Properties for the "rtsp" section:	
body_regex	The regular expression that the RTSP response body must match. • Type: String • Default value: <none>
path	The path to use in the RTSP request (some servers will return 500 Internal Server Error unless this is a valid media file). • Type: String • Default value: "/"
status_regex	The regular expression that the RTSP response status code must match. • Type: String • Default value: "^[234][0-9][0-9]$"
Properties for the "script" section:	
arguments	A table containing arguments and argument values to be passed to the monitor program. • Primary key: – name (String): The name of the argument to be passed to the monitor program. • Sub keys: – value (String): The value of the argument to be passed to the monitor program. – description (String): A description for the argument provided to the program.
program	The program to run. This must be an executable file, either within the monitor scripts directory or specified as an absolute path to some other location on the filesystem. • Type: String • Default value: <none>
Properties for the "sip" section:	
body_regex	The regular expression that the SIP response body must match. • Type: String • Default value: <none>
status_regex	The regular expression that the SIP response status code must match. • Type: String • Default value: "^[234][0-9][0-9]$"

Property	Description
transport	Which transport protocol the SIP monitor will use to query the server. • Type: Enum(String) • Default value: "udp" • Permitted values: "tcp": TCP "udp": UDP
Properties for the "tcp" section:	
close_string	An optional string to write to the server before closing the connection. • Type: String • Default value: <none>
max_response_len	The maximum amount of data to read back from a server, use 0 for unlimited. Applies to TCP and HTTP monitors. • Type: UInt • Default value: "2048"
response_regex	A regular expression to match against the response from the server. Applies to TCP monitors only. • Type: String • Default value: ".+"
write_string	The string to write down the TCP connection. • Type: String • Default value: <none>
Properties for the "udp" section:	
accept_all	If this monitor uses UDP, should it accept responses from any IP and port? • Type: Boolean • Default value: false

Monitor Program

URI Path: monitor_scripts

An executable program that can be used to by external program monitors to report the health of backend services.

Property	Description
There are no properties to display for this resource.	

NAT Configuration

URI Path: appliance/nat

The NAT configuration file stores rules controlling NAT on an appliance.

Property	Description
many_to_one_all_ports	This is table 'many_to_one_all_ports' • Primary key: – rule_number (String): A unique rule identifier • Sub keys: – pool (String): Pool of a "many to one overload" type NAT rule. – tip (String): TIP Group of a "many to one overload" type NAT rule.
many_to_one_port_locked	This is table 'many_to_one_port_locked' • Primary key: – rule_number (String): A unique rule identifier • Sub keys: – pool (String): Pool of a "many to one port locked" type NAT rule. – port (UInt): Port number of a "many to one port locked" type NAT rule. – protocol (Enum(String)): Protocol of a "many to one port locked" type NAT rule. Permitted values: "icmp": ICMP "sctp": SCTP "tcp": TCP "udp": UDP "udplite": UDPLITE – tip (String): TIP Group of a "many to one port locked" type NAT rule.
one_to_one	This is table 'one_to_one' • Primary key: – rule_number (String): A unique rule identifier • Sub keys: – enable_inbound (Boolean): Enabling the inbound part of a "one to one" type NAT rule. – ip (String): IP Address of a "one to one" type NAT rule. – tip (String): TIP group of a "one to one" type NAT rule.
port_mapping	This is table 'port_mapping' • Primary key: – rule_number (String): A unique rule identifier • Sub keys: – dport_first (UInt): First port of the dest. port range of a "port mapping" rule. – dport_last (UInt): Last port of the dest. port range of a "port mapping" rule. – virtual_server (String): Target Virtual Server of a "port mapping" rule.

Pool

URI Path: pools

The conf/pools directory contains configuration files for backend node pools. The name of a file is the name of the pool it defines. Pools can be configured under the Services > Pools section of the Admin Server UI or by using functions under the Pool section of the SOAP API and CLI.

Property	Description
bandwidth_class	The Bandwidth Management Class this pool uses, if any. • Type: Reference(config-bandwidth) • Default value: <none>
failure_pool	If all of the nodes in this pool have failed, then requests can be diverted to another pool. • Type: Reference(config-pool) • Default value: <none>
max_connection_attempts	The maximum number of nodes to which the traffic manager will attempt to send a request before returning an error to the client. Requests that are non-retryable will be attempted against only one node. Zero signifies no limit. • Type: UInt • Default value: <none>
max_idle_connections_pernode	The maximum number of unused HTTP keepalive connections that should be maintained to an individual node. Zero signifies no limit. • Type: UInt • Default value: "50"
max_timed_out_connection_attempts	The maximum number of connection attempts the traffic manager will make where the server fails to respond within the time limit defined by the max_reply_time setting. Zero signifies no limit. • Type: UInt • Default value: "2"
monitors	The monitors assigned to this pool, used to detect failures in the back end nodes. • Type: Set(Reference(config-monitor)) • Default value: <none>
node_close_with_rst	Whether or not connections to the back-end nodes should be closed with a RST packet, rather than a FIN packet. This avoids the TIME_WAIT state, which on rare occasions allows wandering duplicate packets to be safely ignored. • Type: Boolean • Default value: false
node_connection_attempts	The number of times the software will attempt to connect to the same back-end node before marking it as failed. This is only used when passive_monitoring is enabled. • Type: UInt • Default value: "3"

Property	Description
node_delete_behavior	Specify the deletion behavior for nodes in this pool. • Type: Enum(String) • Default value: "immediate" • Permitted values: "drain": Allow existing connections to the node to finish before deletion. "immediate": All connections to the node are closed immediately.
node_drain_to_delete_timeout	The maximum time that a node will be allowed to remain in a draining state after it has been deleted. A value of 0 means no maximum time. • Type: UInt • Default value: <none>
nodes_table	A table of all nodes in this pool. A node should be specified as a <ip>:<port> pair, and has a state, weight and priority. • Primary key: – node (String): A node is a combination of an ip address and port • Sub keys: – priority (UInt): The priority of the node, higher values signify higher priority. If a priority is not specified for a node it is assumed to be 1. – state (Enum(String)): The state of the pool, which can either be Active, Draining or Disabled Permitted values: "active": The node is is active. "disabled": The node is disabled. "draining": The node is draining. – weight (Int): Weight for the node. The actual value in isolation does not matter: As long as it is a valid integer 1-100, the per-node weightings are calculated on the relative values between the nodes. – source_ip (String): The source address the Traffic Manager uses to connect to this node.
note	A description of the pool. • Type: String • Default value: <none>
passive_monitoring	Whether or not the software should check that 'real' requests (i.e. not those from monitors) to this pool appear to be working. This should normally be enabled, so that when a node is refusing connections, responding too slowly, or sending back invalid data, it can mark that node as failed, and stop sending requests to it. If this is disabled, you should ensure that suitable health monitors are configured to check your servers instead, otherwise failed requests will not be detected and subsequently retried. • Type: Boolean • Default value: false
persistence_class	The default Session Persistence class this pool uses, if any. • Type: Reference(config-persistence) • Default value: <none>

Property	Description
transparent	Whether or not connections to the back-ends appear to originate from the source client IP address. • Type: Boolean • Default value: false
Properties for the "auto_scaling" section:	
addnode_delaytime	The time in seconds from the creation of the node which the traffic manager should wait before adding the node to the autoscaled pool. Set this to allow applications on the newly created node time to intialize before being sent traffic. • Type: UInt • Default value: <none>
cloud_credentials	The Cloud Credentials object containing authentication credentials to use in cloud API calls. • Type: Reference(cloud-api) • Default value: <none>
cluster	The ESX host or ESX cluster name to put the new virtual machine instances on. • Type: String • Default value: <none>
data_center	The name of the logical datacenter on the vCenter server. Virtual machines will be scaled up and down under the datacenter root folder. • Type: String • Default value: <none>
data_store	The name of the datastore to be used by the newly created virtual machine. • Type: String • Default value: <none>
enabled	Are the nodes of this pool subject to autoscaling? If yes, nodes will be automatically added and removed from the pool by the chosen autoscaling mechanism. • Type: Boolean • Default value: false
external	Whether or not autoscaling is being handled by an external system. Set this value to Yes if all aspects of autoscaling are handled by an external system, such as RightScale. If set to No, the traffic manager will determine when to scale the pool and will communicate with the cloud provider to create and destroy nodes as necessary. • Type: Boolean • Default value: false
hysteresis	The time period in seconds for which a change condition must persist before the change is actually instigated. • Type: UInt • Default value: "20"

Property	Description
imageid	The identifier for the image of the instances to create. • Type: String • Default value: <none>
ips_to_use	Which type of IP addresses on the node to use. Choose private IPs if the traffic manager is in the same cloud as the nodes, otherwise choose public IPs. • Type: Enum(String) • Default value: "publicips" • Permitted values: "private_ips": Private IP addresses "publicips": Public IP addresses
last_node_idle_time	The time in seconds for which the last node in an autoscaled pool must have been idle before it is destroyed. This is only relevant if min_nodes is 0. • Type: UInt • Default value: "3600"
max_nodes	The maximum number of nodes in this autoscaled pool. • Type: UInt • Default value: "4"
min_nodes	The minimum number of nodes in this autoscaled pool. • Type: UInt • Default value: "1"
name	The beginning of the name of nodes in the cloud that are part of this autoscaled pool. • Type: String • Default value: <none>
port	The port number to use for each node in this autoscaled pool. • Type: UInt • Default value: "80"
refractory	The time period in seconds after the instigation of a re-size during which no further changes will be made to the pool size. • Type: UInt • Default value: "180"
response_time	The expected response time of the nodes in ms. This time is used as a reference when deciding whether a node's response time is conforming. All responses from all the nodes will be compared to this reference and the percentage of conforming responses is the base for decisions about scaling the pool up or down. • Type: UInt • Default value: "1000"
scale_down_level	The fraction, in percent, of conforming requests above which the pool size is decreased. If the percentage of conforming requests exceeds this value, the pool is scaled down. • Type: UInt • Default value: "95"

Property	Description
scale_up_level	The fraction, in percent, of conforming requests below which the pool size is increased. If the percentage of conforming requests drops below this value, the pool is scaled up. • Type: UInt • Default value: "40"
securitygroupids	List of security group IDs to associate to the new EC2 instance. • Type: Set(String) • Default value: <none>
size_id	The identifier for the size of the instances to create. • Type: String • Default value: <none>
subnetids	List of subnet IDs where the new EC2-VPC instance(s) will be launched. Instances will be evenly distributed among the subnets. If the list is empty, instances will be launched inside EC2-Classic. • Type: Set(String) • Default value: <none>
Properties for the "connection" section:	
max_connect_time	How long the pool should wait for a connection to a node to be established before giving up and trying another node. • Type: UInt • Default value: "4"
max_connections_per_node	The maximum number of concurrent connections allowed to each back-end node in this pool per machine. A value of 0 means unlimited connections. • Type: UInt • Default value: <none>
max_queue_size	The maximum number of connections that can be queued due to connections limits. A value of 0 means unlimited queue size. • Type: UInt • Default value: <none>
max_reply_time	How long the pool should wait for a response from the node before either discarding the request or trying another node (retryable requests only). • Type: UInt • Default value: "30"
queue_timeout	The maximum time to keep a connection queued in seconds. • Type: UInt • Default value: "10"
Properties for the "dns_autoscale" section:	
enabled	When enabled, the Traffic Manager will periodically resolve the hostnames in the "hostnames" list using a DNS query, and use the results to automatically add, remove or update the IP addresses of the nodes in the pool. • Type: Boolean • Default value: false

Property	Description
hostnames	A list of hostnames which will be used for DNS-derived autoscaling • Type: Set(String) • Default value: <none>
port	The port number to use for each node when using DNS-derived autoscaling • Type: UInt • Default value: "80"
Properties for the "ftp" section:	
support_rfc_2428	Whether or not the backend IPv4 nodes understand the EPRT and EPSV command from RFC 2428. It is always assumed that IPv6 nodes support these commands. • Type: Boolean • Default value: false
Properties for the "http" section:	
keepalive	Whether or not the pool should maintain HTTP keepalive connections to the nodes. • Type: Boolean • Default value: false
keepalive_non_idempotent	Whether or not the pool should maintain HTTP keepalive connections to the nodes for non-idempotent requests. • Type: Boolean • Default value: false
Properties for the "kerberos_protocol_transition" section:	
principal	The Kerberos principal the traffic manager should use when performing Kerberos Protocol Transition. • Type: String • Default value: <none>
target	The Kerberos principal name of the service this pool targets. • Type: String • Default value: <none>
Properties for the "l4accel" section:	
snat	Whether connections to the back-end nodes should appear to originate from an IP address raised on the traffic manager, rather than the IP address from which they were received by the traffic manager. • Type: Boolean • Default value: false
Properties for the "load_balancing" section:	

Property	Description
algorithm	The load balancing algorithm that this pool uses to distribute load across its nodes. • Type: Enum(String) • Default value: "round_robin" • Permitted values: "fastest_response_time": The Response Time algorithm monitors the response times for recent requests to each node. It sends each new request to the node that has recently been responding the most quickly. "least_connections": This algorithm sends each new request to the node with the fewest currently active connections. "perceptive": The Perceptive algorithm uses a combination of response time data and connection counts to predict which node is likely to have the fastest response time for each request. "random": This algorithm chooses a random node for each request. "round_robin": This algorithm distributes traffic by assigning each request to a new node in turn. "weighted_least_connections": This algorithm works in a similar way to the Least Connections algorithm, but assigns more requests to nodes with a greater 'weight'. "weighted_round_robin": Weighted Round Robin works in a similar way to Round Robin, but assigns more requests to nodes with a greater 'weight'.
priority_enabled	Enable priority lists. • Type: Boolean • Default value: false
priority_nodes	Minimum number of highest-priority active nodes. • Type: UInt • Default value: "1"
Properties for the "node" section:	
close_on_death	Close all connections to a node once we detect that it has failed. • Type: Boolean • Default value: false
retry_fail_time	The amount of time, in seconds, that a traffic manager will wait before re-trying a node that has been marked as failed by passive monitoring. • Type: UInt • Default value: "60"
Properties for the "smtp" section:	
send_starttls	If we are encrypting traffic for an SMTP connection, should we upgrade to SSL using STARTTLS. • Type: Boolean • Default value: false
Properties for the "ssl" section:	

Property	Description
client_auth	Whether or not a suitable certificate and private key from the SSL Client Certificates catalog be used if the back-end server requests client authentication. • Type: Boolean • Default value: false
common_name_match	A list of names against which the 'common name' of the certificate is matched; these names are used in addition to the node's hostname or IP address as specified in the config file or added by the autoscaler process. • Type: Set(String) • Default value: <none>
elliptic_curves	The SSL elliptic curve preference list for SSL connections from this pool using TLS version 1.0 or higher. Leaving this empty will make the pool use the globally configured preference list. The named curves P256, P384 and P521 may be configured. • Type: List(String) • Default value: <none>
enable	Whether or not the pool should encrypt data before sending it to a back-end node. • Type: Boolean • Default value: false
enhance	SSL protocol enhancements allow your traffic manager to prefix each new SSL connection with information about the client. This enables Brocade vTM virtual servers referenced by this pool to discover the original client's IP address. Only enable this if you are using nodes for this pool which are Brocade Virtual Traffic Managers, whose virtual servers have the ssl_trust_magic setting enabled. • Type: Boolean • Default value: false
send_close_alerts	Whether or not to send an SSL/TLS "close alert" when initiating a socket disconnection. • Type: Boolean • Default value: false
server_name	Whether or not the software should use the TLS 1.0 server_name extension, which may help the back-end node provide the correct certificate. Enabling this setting will force the use of at least TLS 1.0. • Type: Boolean • Default value: false
signature_algorithms	The SSL signature algorithms preference list for SSL connections from this pool using TLS version 1.2 or higher. Leaving this empty will make the pool use the globally configured preference list, signature_algorithms in the ssl section of the global_settings resource. See there and in the online help for how to specify SSL signature algorithms. • Type: String • Default value: <none>

Property	Description
ssl_ciphers	The SSL/TLS ciphers to allow for connections to a back-end node. Leaving this empty will make the pool use the globally configured ciphers, see configuration key ssl!ssl3_ciphers in the Global Settings section of the System tab. See there for how to specify SSL/TLS ciphers. • Type: String • Default value: <none>
ssl_support_ssl2	No longer supported. Formerly controlled whether SSLv2 could be used for SSL connections to pool nodes. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable SSLv2 "enabled": Enable SSLv2 (not recommended) "use_default": Use the global setting for SSLv2
ssl_support_ssl3	Whether or not SSLv3 is enabled for this pool. Choosing the global setting means the value of the configuration key ssl!support_ssl3 from the Global Settings section of the System tab will be enforced. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable SSLv3 "enabled": Enable SSLv3 "use_default": Use the global setting for SSLv3
ssl_support_tls1	Whether or not TLSv1.0 is enabled for this pool. Choosing the global setting means the value of the configuration key ssl!support_tls1 from the Global Settings section of the System tab will be enforced. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable TLSv1.0 "enabled": Enable TLSv1.0 "use_default": Use the global setting for TLSv1.0
ssl_support_tls1_1	Whether or not TLSv1.1 is enabled for this pool. Choosing the global setting means the value of the configuration key ssl!support_tls1.1 from the Global Settings section of the System tab will be enforced. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable TLSv1.1 "enabled": Enable TLSv1.1 "use_default": Use the global setting for TLSv1.1

Property	Description
ssl_support_tls1_2	Whether or not TLSv1.2 is enabled for this pool. Choosing the global setting means the value of the configuration key ssl!support_tls1.2 from the Global Settings section of the System tab will be enforced. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable TLSv1.2 "enabled": Enable TLSv1.2 "use_default": Use the global setting for TLSv1.2
strict_verify	Whether or not strict certificate verification should be performed. This will turn on checks to disallow server certificates that don't match the server name or a name in the ssl_common_name_match list, are self-signed, expired, revoked, or have an unknown CA. • Type: Boolean • Default value: false
Properties for the "tcp" section:	
nagle	Whether or not Nagle's algorithm should be used for TCP connections to the back-end nodes. • Type: Boolean • Default value: false
Properties for the "udp" section:	
accept_from	The IP addresses and ports from which responses to UDP requests should be accepted. If set to accept responses from a specific set of IP addresses, you will need to enter a CIDR Mask (such as 10.100.0.0/16). • Type: Enum(String) • Default value: "dest_only" • Permitted values: "all": Any IP address and any port. "dest_ip_only": Only the IP address to which the request was sent, but from any port. "dest_only": Only the IP address and port to which the request was sent. "ip_mask": Only a specific set of IP addresses, but from any port.
accept_from_mask	The CIDR mask that matches IPs we want to receive responses from. • Type: String • Default value: <none>
response_timeout	The maximum length of time that a node is permitted to take after receiving a UDP request packet before sending a reply packet. Zero indicates that there is no maximum, preventing a node that does not send replies from being presumed to have failed. • Type: UInt • Default value: <none>

Protection Class

URI Path: protection

A protection class specifies the level of protection against network attacks for a virtual server.

Property	Description
debug	Whether or not to output verbose logging. • Type: Boolean • Default value: false
enabled	Enable or disable this service protection class. • Type: Boolean • Default value: false
log_time	Log service protection messages at these intervals. If set to 0 no messages will be logged and no alerts will be sent. • Type: UInt • Default value: "60"
note	A description of the service protection class. • Type: String • Default value: <none>
per_process_connection_count	Whether simultaneous connection counting and limits are per-process. (Each Traffic Manager typically has several processes: one process per available CPU core.) If Yes, a connecting IP address may make that many connections to each process within a Traffic Manager. If No, a connecting IP address may make that many connections to each Traffic Manager as a whole. • Type: Boolean • Default value: false
rule	A TrafficScript rule that will be run on the connection after the service protection criteria have been evaluated. This rule will be executed prior to normal rules configured for the virtual server. • Type: Reference(config-trafficscript) • Default value: <none>
testing	Place the service protection class into testing mode. (Log when this class would have dropped a connection, but allow all connections through). • Type: Boolean • Default value: false
Properties for the "access_restriction" section:	
allowed	Always allow access to these IP addresses. This overrides the connection limits for these machines, but does not stop other restrictions such as HTTP validity checks. • Type: Set(String) • Default value: <none>
banned	Disallow access to these IP addresses. • Type: Set(String) • Default value: <none>

Property	Description
Properties for the "connection_limiting" section:	
max_10_connections	Additional limit on maximum simultaneous connections from the top 10 busiest connecting IP addresses combined. The value should be between 1 and 10 times the max_1_connections limit. (This limit is disabled if per_process_connection_count is No, or max_1_connections is 0, or min_connections is 0.) • Type: UInt • Default value: "200"
max_1_connections	Maximum simultaneous connections each connecting IP address is allowed. Set to 0 to disable this limit. • Type: UInt • Default value: "30"
max_connection_rate	Maximum number of new connections each connecting IP address is allowed to make in the rate_timer interval. Set to 0 to disable this limit. If applied to an HTTP Virtual Server each request sent on a connection that is kept alive counts as a new connection. The rate limit is per process: each process within a Traffic Manager accepts new connections from the connecting IP address at this rate. (Each Traffic Manager typically has several processes: one process per available CPU core). • Type: UInt • Default value: <none>
min_connections	Entry threshold for the max_10_connections limit: the max_10_connections limit is not applied to connecting IP addresses with this many or fewer simultaneous connections. Setting to 0 disables both the max_1_connections and max_10_connections limits, if per_process_connection_count is Yes. (If per_process_connection_count is No, this setting is ignored.) • Type: UInt • Default value: "4"
rate_timer	How frequently the max_connection_rate is assessed. For example, a value of 1 (second) will impose a limit of max_connection_rate connections per second; a value of 60 will impose a limit of max_connection_rate connections per minute. The valid range is 1-99999 seconds. • Type: UInt • Default value: "60"
Properties for the "http" section:	
check_rfc2396	Whether or not requests with poorly-formed URLs be should be rejected. This tests URL compliance as defined in RFC2396. Note that enabling this may block some older, non-conforming web browsers. • Type: Boolean • Default value: false
max_body_length	Maximum permitted length of HTTP request body data, set to 0 to disable the limit. • Type: UInt • Default value: <none>

Property	Description
max_header_length	Maximum permitted length of a single HTTP request header (key and value), set to 0 to disable the limit. • Type: UInt • Default value: <none>
max_request_length	Maximum permitted size of all the HTTP request headers, set to 0 to disable the limit. • Type: UInt • Default value: <none>
max_url_length	Maximum permitted URL length, set to 0 to disable the limit. • Type: UInt • Default value: <none>
reject_binary	Whether or not URLs and HTTP request headers that contain binary data (after decoding) should be rejected. • Type: Boolean • Default value: false
send_error_page	This setting tells the traffic manager to send an HTTP error message if a connection fails the service protection tests, instead of just dropping it. Details of which HTTP response will be sent when particular tests fail can be found in the Help section for this page. • Type: Boolean • Default value: false

Rate Shaping Class

URI Path: rate

A rate shaping class restricts the number of connections being processed by a virtual server at once.

Property	Description
max_rate_per_minute	Requests that are associated with this rate class will be rate-shaped to this many requests per minute, set to 0 to disable the limit. • Type: UInt • Default value: <none>
max_rate_per_second	Although requests will be rate-shaped to the max_rate_per_minute, the traffic manager will also rate limit per-second. This smooths traffic so that a full minute's traffic will not be serviced in the first second of the minute, set this to 0 to disable the per-second limit. • Type: UInt • Default value: <none>
note	A description of the rate class. • Type: FreeformString • Default value: <none>

Rule

URI Path: rules

TrafficScript rules allow traffic inspection and modification.

Property	Description
There are no properties to display for this resource.	

SLM Class

URI Path: service_level_monitors

Service level monitoring is used to produce alerts when an application's performance is degraded. This is done by monitoring the response time of connections to a virtual server.

Property	Description
note	A description for the SLM class. • Type: FreeformString • Default value: <none>
response_time	Responses that arrive within this time limit, expressed in milliseconds, are treated as conforming. • Type: UInt • Default value: "1000"
serious_threshold	When the percentage of conforming responses drops below this level, a serious error level message will be emitted. • Type: UInt • Default value: <none>
warning_threshold	When the percentage of conforming responses drops below this level, a warning message will be emitted. • Type: UInt • Default value: "50"

SSL Client Key Pair

URI Path: ssl/client_keys

SSL Client Certificates are used when connecting to backend nodes that require client certificate authentication.

Property	Description
note	Notes for this certificate • Type: FreeformString • Default value: <none>
private	Private key for certificate • Type: FreeformString • Default value: <none>
public	Public certificate • Type: FreeformString • Default value: <none>
request	Certificate Signing Request for certificate • Type: FreeformString • Default value: <none>

SSL Key Pair

URI Path: ssl/server_keys

SSL Server Certificates are presented to clients by virtual servers when SSL decryption is enabled.

Property	Description
note	Notes for this certificate • Type: FreeformString • Default value: <none>
private	Private key for certificate • Type: FreeformString • Default value: <none>
public	Public certificate • Type: FreeformString • Default value: <none>
request	Certificate Signing Request for certificate • Type: FreeformString • Default value: <none>

SSL Trusted Certificate

URI Path: ssl/cas

SSL certificate authority certificates (CAs) and certificate revocation lists (CRLs) can be used when validating server and client certificates.

Property	Description
There are no properties to display for this resource.	

Security Settings

URI Path: security

Security settings that restrict remote administration for the cluster. Additional security options can be found in Global Settings.

Property	Description
access	Access to the admin server and REST API is restricted by usernames and passwords. You can further restrict access to just trusted IP addresses, CIDR IP subnets or DNS wildcards. These access restrictions are also used when another traffic manager initially joins the cluster, after joining the cluster these restrictions are no longer used. Care must be taken when changing this setting, as it can cause the administration server to become inaccessible.Access to the admin UI will not be affected until it is restarted. • Type: Set(String) • Default value: <none>
Properties for the "ssh_intrusion" section:	
bantime	The amount of time in seconds to ban an offending host for. • Type: UInt • Default value: "600"
blacklist	The list of hosts to permanently ban, identified by IP address or DNS hostname in a space-separated list. • Type: Set(String) • Default value: <none>
enabled	Whether or not the SSH Intrusion Prevention tool is enabled. • Type: Boolean • Default value: false
findtime	The window of time in seconds the maximum number of connection attempts applies to. More than (maxretry) failed attempts in this time span will trigger a ban. • Type: UInt • Default value: "600"

Property	Description
maxretry	The number of failed connection attempts a host can make before being banned. • Type: UInt • Default value: "6"
whitelist	The list of hosts to never ban, identified by IP address, DNS hostname or subnet mask, in a space-separated list. • Type: Set(String) • Default value: <none>

Session Persistence Class

URI Path: persistence

A session persistence class is used to identify the session a new connection belongs too and deliver it to the same backend node.

Property	Description
cookie	The cookie name to use for tracking session persistence. • Type: String • Default value: <none>
delete	Whether or not the session should be deleted when a session failure occurs. (Note, setting a failure mode of 'choose a new node' implicitly deletes the session.) • Type: Boolean • Default value: false
failure_mode	The action the pool should take if the session data is invalid or it cannot contact the node specified by the session. • Type: Enum(String) • Default value: "new_node" • Permitted values: "close": Close the connection (using error_file on Virtual Servers > Edit > Protocol Settings) "new_node": Choose a new node to use "url": Redirect the user to a given URL
note	A description of the session persistence class. • Type: FreeformString • Default value: <none>
subnet_prefix_length_v4	When using IP-based session persistence, ensure all requests from this IPv4 subnet, specified as a prefix length, are sent to the same node. If set to 0, requests from different IPv4 addresses will be load-balanced individually. • Type: Int • Default value: <none>

Property	Description
subnet_prefix_length_v6	When using IP-based session persistence, ensure all requests from this IPv6 subnet, specified as a prefix length, are sent to the same node. If set to 0, requests from different IPv6 addresses will be load-balanced individually. • Type: Int • Default value: <none>
type	The type of session persistence to use. • Type: Enum(String) • Default value: "ip" • Permitted values: "asp": ASP and ASP.NET session persistence "cookie": Monitor application cookies "ip": IP-based persistence "j2ee": J2EE session persistence "named": Named Node session persistence "ssl": SSL Session ID persistence "transparent": Transparent session affinity "universal": Universal session persistence "x_zeus": X-Zeus-Backend cookies
url	The redirect URL to send clients to if the session persistence is configured to redirect users when a node dies. • Type: String • Default value: <none>

Traffic IP Group

URI Path: traffic_ip_groups

Traffic IP groups are sets of IP addresses that are distributed across a cluster for fault tolerance.

Property	Description
backend_traffic_ips	IP addresses associated with the Traffic IP group that can be used for communication with back-end servers. • Type: Set(String) • Default value: <none>
enabled	If set to No, the traffic IP group will be disabled and none of the traffic IP addresses will be raised. • Type: Boolean • Default value: false
hash_source_port	Whether or not the source port should be taken into account when deciding which traffic manager should handle a request. • Type: Boolean • Default value: false

Property	Description
ip_assignment_mode	Configure how traffic IPs are assigned to traffic managers in Single-Hosted mode • Type: Enum(String) • Default value: "balanced" • Permitted values: "alphabetic": Alphabetical order of traffic manager hostnames "balanced": Approximately balanced between traffic managers
ip_mapping	A table assigning traffic IP addresses to machines that should host them. Traffic IP addresses not specified in this table will automatically be assigned to a machine. • Primary key: – ip (String): A traffic IP address (from the ipaddresses property). • Sub keys: – traffic_manager (String): The name of the traffic manager that should host the IP address.
ipaddresses	The IP addresses that belong to the Traffic IP group. • Type: Set(String) • Default value: <none>
keeptogether	If set to Yes then all the traffic IPs will be raised on a single traffic manager. By default they're distributed across all active traffic managers in the traffic IP group. • Type: Boolean • Default value: false
location	The location in which the Traffic IP group is based. • Type: Int • Default value: <none>
machines	The traffic managers that can host the traffic IP group's IP addresses. • Type: Set(Reference(config-tm)) • Default value: <none>
mode	The method used to distribute traffic IPs across machines in the cluster. If "multihosted" is used then multicast must be set to an appropriate multicast IP address. • Type: Enum(String) • Default value: "singlehosted" • Permitted values: "ec2elastic": Use an EC2-Classic Elastic IP address. "ec2vpcelastic": Use an EC2-VPC Elastic IP address. "ec2vpcprivate": Use an EC2-VPC Private IP address. "multihosted": Raise each address on every machine in the group (Multi-Hosted mode) - IPv4 only "rhi": Use route health injection to route traffic to the active machine - IPv4 only "singlehosted": Raise each address on a single machine (Single-Hosted mode)

Property	Description
multicast	The multicast IP address used to duplicate traffic to all traffic managers in the group. • Type: String • Default value: <none>
note	A note, used to describe this Traffic IP Group • Type: String • Default value: <none>
rhi_bgp_metric_base	The base BGP routing metric for this Traffic IP group. This is the advertised routing cost for the active traffic manager in the cluster. It can be used to set up inter-cluster failover. • Type: UInt • Default value: "10"
rhi_bgp_passive_metric_offset	The BGP routing metric offset for this Traffic IP group. This is the difference between the advertised routing cost for the active and passive traffic manager in the cluster. • Type: UInt • Default value: "10"
rhi_ospfv2_metric_base	The base OSPFv2 routing metric for this Traffic IP group. This is the advertised routing cost for the active traffic manager in the cluster. It can be used to set up inter-cluster failover. • Type: UInt • Default value: "10"
rhi_ospfv2_passive_metric_offset	The OSPFv2 routing metric offset for this Traffic IP group. This is the difference between the advertised routing cost for the active and passive traffic manager in the cluster. • Type: UInt • Default value: "10"
rhi_protocols	A list of protocols to be used for RHI. Currently must be 'ospf' or 'bgp' or both. The default, if empty, is 'ospf', which means that it is not possible to specify no protocol. • Type: String • Default value: "ospf"
slaves	A list of traffic managers that are in 'passive' mode. This means that in a fully working environment, they will not have any traffic IP addresses assigned to them. • Type: Set(Reference(config-tm)) • Default value: <none>

Traffic Manager

URI Path: traffic_managers

The conf/zxtms directory contains a configuration file for each traffic manager in your cluster. The name of each file is the hostname of the traffic manager it represents. These files contain host-specific configuration data and on each installation of the software, the conf/../global.cfg file is sym-linked to the host's own configuration in the conf/zxtms directory. The files may contain a variety of configuration options that are configured in various locations under the System section of the Admin Server UI and the System section of the SOAP API and CLI.

Property	Description
adminMasterXMLIP	The Application Firewall master XML IP. • Type: String • Default value: "0.0.0.0"
adminSlaveXMLIP	The Application Firewall slave XML IP. • Type: String • Default value: "0.0.0.0"
appliance_card	The table of network cards of a hardware appliance • Primary key: – name (String): Network card PCI ID • Sub keys: – interfaces (List(String)): The order of the interfaces of a network card – label (String): The labels of the installed network cards
appliance_sysctl	Custom kernel parameters applied by the user with sysctl interface • Primary key: – sysctl (String): The name of the kernel parameter, e.g. net.ipv4.forward • Sub keys: – description (String): Associated optional description for the sysctl – value (String): The value of the kernel parameter
authenticationServerIP	The Application Firewall Authentication Server IP. • Type: String • Default value: "0.0.0.0"
cloud_platform	Cloud platform where the traffic manager is running. • Type: String • Default value: <none>
location	This is the location of the local traffic manager is in. • Type: String • Default value: <none>
nameip	Replace Traffic Manager name with an IP address. • Type: String • Default value: <none>

Property	Description
num_aptimizer_threads	How many worker threads the Web Accelerator process should create to optimise content. By default, one thread will be created for each CPU on the system. • Type: UInt • Default value: <none>
num_children	The number of worker processes the software will run. By default, one child process will be created for each CPU on the system. You may wish to reduce this to effectively "reserve" CPU(s) for other processes running on the host system. • Type: UInt • Default value: <none>
numberOfCPUs	The number of Application Firewall decider process to run. • Type: UInt • Default value: <none>
restServerPort	The Application Firewall REST Internal API port, this port should not be accessed directly • Type: UInt • Default value: <none>
trafficip	A table mapping interfaces to networks, used by the traffic manager to select which interface to raise a Traffic IP on. • Primary key: – name (String): A network interface. • Sub keys: – networks (Set(String)): A set of IP/masks to which the network interface maps.
updaterIP	The Application Firewall Updater IP. • Type: String • Default value: "0.0.0.0"
Properties for the "appliance" section:	
gateway_ipv4	The default gateway. • Type: String • Default value: <none>
gateway_ipv6	The default IPv6 gateway. • Type: String • Default value: <none>
hostname	Name (hostname.domainname) of the appliance. • Type: String • Default value: <none>

Property	Description
hosts	A table of hostname to static ip address mappings, to be placed in the /etc/hosts file. • Primary key: – name (String): The name of a host. • Sub keys: – ip_address (String): The static IP address of the host.
if	A table of network interface specific settings. • Primary key: – name (String): A network interface name. • Sub keys: – autoneg (Boolean): Whether auto-negotiation should be enabled for the interface. – bmode (Enum(String)): The trunking mode used for the interface (only 802.3ad is currently supported). Permitted values: "802_3ad": IEEE 802.3ad "balance_alb": Adaptive Load Balancing – bond (String): The trunk of which the interface should be a member. – duplex (Boolean): Whether full-duplex should be enabled for the interface. – mode (Enum(String)): Set the configuriation mode of an interface, the interface name is used in place of the * (asterisk). Permitted values: "dhcp": DHCP "static": Static – mtu (UInt): The maximum transmission unit (MTU) of the interface. – speed (Enum(String)): The speed of the interface. Permitted values: "10": 10Mbs "100": 100Mbs "1000": 1Gbs
ip	A table of network interfaces and their network settings. • Primary key: – name (String): A network interface name. • Sub keys: – addr (String): The IP address for the interface. – isexternal (Boolean): Whether the interface is externally facing. – mask (String): The IP mask (netmask) for the interface.
ipmi_lan_access	Whether IPMI LAN access should be enabled or not. • Type: Boolean • Default value: false

Property	Description
ipmi_lan_addr	The IP address of the appliance IPMI LAN channel. • Type: String • Default value: <none>
ipmi_lan_gateway	The default gateway of the IPMI LAN channel. • Type: String • Default value: <none>
ipmi_lan_ipsrc	The addressing mode the IPMI LAN channel operates. • Type: Enum(String) • Default value: "static" • Permitted values: "dhcp": Address obtained by DHCP "static": Static IP Address
ipmi_lan_mask	Set the IP netmask for the IPMI LAN channel. • Type: String • Default value: <none>
ipv4_forwarding	Whether or not IPv4 forwarding is enabled. • Type: Boolean • Default value: false
ipv6_forwarding	Whether or not IPv6 forwarding is enabled. • Type: Boolean • Default value: false
licence_agreed	Whether or not the license agreement has been accepted. This determines whether or not the Initial Configuration wizard is displayed. • Type: Boolean • Default value: false
manageazureroutes	Whether or not the software manages the Azure policy routing. • Type: Boolean • Default value: false
managedpa	Whether or not the software manages system configuration based on Data Plane Acceleration mode • Type: Boolean • Default value: false
manageec2conf	Whether or not the software manages the EC2 config. • Type: Boolean • Default value: false
manageiptrans	Whether or not the software manages the IP transparency • Type: Boolean • Default value: false

Property	Description
managereturnpath	Whether or not the software manages return path routing. If disabled, the appliance won't modify iptables / rules / routes for this feature. • Type: Boolean • Default value: false
managevpcconf	Whether or not the software manages the EC2-VPC secondary IPs. • Type: Boolean • Default value: false
name_servers	The IP addresses of the nameservers the appliance should use and place in /etc/resolv.conf. • Type: Set(String) • Default value: <none>
ntpservers	The NTP servers the appliance should use to synchronize its clock. • Type: List(String) • Default value: "0.vyatta.pool.ntp.org 1.vyatta.pool.ntp.org 2.vyatta.pool.ntp.org 3.vyatta.pool.ntp.org"
routes	A table of destination IP addresses and routing details to reach them. • Primary key: – name (String): A destination IP address. • Sub keys: – gw (String): The gateway IP to configure for the route. – if (String): The network interface to configure for the route. – mask (String): The netmask to apply to the IP address.
search_domains	The search domains the appliance should use and place in /etc/resolv.conf. • Type: Set(String) • Default value: <none>
shim_client_id	The client ID provided by the portal for this server. • Type: String • Default value: <none>
shim_client_key	The client key provided by the portal for this server. • Type: String • Default value: <none>
shim_enabled	Enable the Riverbed Cloud SteelHead discovery agent on this appliance. • Type: Boolean • Default value: false
shim_ips	The IP addresses of the Riverbed Cloud SteelHeads to use, as a space or comma separated list. If using priority load balancing this should be in ascending order of priority (highest priority last). • Type: String • Default value: <none>

Property	Description
shim_load_balance	The load balancing method for selecting a Riverbed Cloud SteelHead appliance. • Type: Enum(String) • Default value: "round_robin" • Permitted values: "priority": Priority "round_robin": Round Robin
shim_log_level	The minimum severity that the discovery agent will record to its log. • Type: Enum(UInt) • Default value: "notice" • Permitted values: "critical": Log critical errors "debug": Log debug or more severe errors (all errors) "info": Log info or more severe errors "notice": Log notice or more severe errors "serious": Log serious or more severe errors "warning": Log warning or more severe errors
shim_mode	The mode used to discover Riverbed Cloud SteelHeads in the local cloud or data center. • Type: Enum(String) • Default value: "portal" • Permitted values: "local": Local Portal "manual": Manual "portal": Riverbed Portal
shim_portal_url	The hostname or IP address of the local portal to use. • Type: String • Default value: <none>
shim_proxy_host	The IP or hostname of the proxy server to use to connect to the portal. Leave blank to not use a proxy server. • Type: String • Default value: <none>
shim_proxy_port	The port of the proxy server, must be set if a proxy server has been configured. • Type: String • Default value: <none>
ssh_enabled	Whether or not the SSH server is enabled on the appliance. • Type: Boolean • Default value: false
ssh_password_allowed	Whether or not the SSH server allows password based login. • Type: Boolean • Default value: false

Property	Description
ssh_port	The port that the SSH server should listen on. • Type: UInt • Default value: "22"
timezone	The timezone the appliance should use. This must be a path to a timezone file that exists under /usr/share/zoneinfo/. • Type: String • Default value: "US/Pacific"
vlans	The VLANs the software should raise. A VLAN should be configured using the format <dev>.<vlanid>, where <dev> is the name of a network device that exists in the host system, eth0.100 for example. • Type: Set(String) • Default value: <none>
Properties for the "autodiscover" section:	
Properties for the "cluster_comms" section:	
allow_update	Whether or not this instance of the software can send configuration updates to other members of the cluster. When not clustered this key is ignored. When clustered the value can only be changed by another machine in the cluster that has allow_update set to true. If set to false then it will not be possible to log into the admin server for this instance. • Type: Boolean • Default value: false
bind_ip	The IP address that the software should bind to for internal administration communications. See also port. If the software is not part of a cluster the default is to use 127.0.0.1 and there should be no reason to touch this setting. If the software is part of a cluster then the default is to listen on all raised IPs, in this case an alternative configuration is to listen on a single IP address. This may be useful if you have a separate management network and wish to restrict control messages to it. It is important to ensure that the allowed_update_hosts (in the Global Settings resource) is compatible with the IP configured here. • Type: String • Default value: "*"
external_ip	This is the optional external ip of the traffic manager, which is used to circumvent natting when traffic managers in a cluster span different networks. • Type: String • Default value: <none>
port	The port that the software should listen on for internal administration communications. See also bind_ip. • Type: UInt • Default value: "9080"
Properties for the "ec2" section:	
trafficips_public_enis	List of MAC addresses of interfaces which the traffic manager can use to associate the EC2 elastic IPs (Traffic IPs) to the instance. • Type: Set(String) • Default value: <none>
Properties for the "fault_tolerance" section:	

Property	Description
bgp_router_id	The BGP router id If set to empty, then the IPv4 address used to communicate with the default IPv4 gateway is used instead. Specifying 0.0.0.0 will stop the traffic manager routing software from running the BGP protocol. • Type: String • Default value: <none>
lss_dedicated_ips	IP addresses associated with the links dedicated by the user for receiving L4 state sync messages from other peers in a cluster. • Type: List(String) • Default value: <none>
ospfv2_ip	The traffic manager's permanent IPv4 address which the routing software will use for peering and transit traffic, and as its OSPF router ID. If set to empty, then the address used to communicate with the default IPv4 gateway is used instead. Specifying 0.0.0.0 will stop the traffic manager routing software from running the OSPF protocol. • Type: String • Default value: <none>
ospfv2_neighbor_addrs	The IP addresses of routers which are expected to be found as OSPFv2 neighbors of the traffic manager. A warning will be reported if some of the expected routers are not peered, and an error will be reported if none of the expected routers are peered. An empty list disables monitoring. The special value %gateway% is a placeholder for the default gateway. • Type: Set(String) • Default value: "%gateway%"
Properties for the "iop" section:	
Properties for the "iptables" section:	
config_enabled	Whether the Traffic Manager should configure the iptables built-in chains to call Traffic Manager defined rules (e.g. the IP transparency chain). This should only be disabled in case of conflict with other software that manages iptables, e.g. firewalls. When disabled, you will need to add rules manually to use these features - see the user manual for details. • Type: Boolean • Default value: false
Properties for the "iptrans" section:	
fwmark	The netfilter forwarding mark to use for IP transparency rules • Type: UInt • Default value: "320"
iptables_enabled	Whether IP transparency may be used via netfilter/iptables. This requires Linux 2.6.24 and the iptables socket extension. For older Linux versions, the "ztrans" kernel module may be used instead. • Type: Boolean • Default value: false
routing_table	The special routing table ID to use for IP transparency rules • Type: UInt • Default value: "320"
Properties for the "java" section:	

Property	Description
port	The port the Java Extension handler process should listen on. This port will be bound for localhost communications only. • Type: UInt • Default value: "9060"
Properties for the "kerberos" section:	
Properties for the "remote_licensing" section:	
email_address	The e-mail address sent as part of a remote licensing request. • Type: String • Default value: <none>
message	A free-text field sent as part of a remote licensing request. • Type: String • Default value: <none>
Properties for the "rest_api" section:	
bind_ips	A list of IP Addresses which the REST API will listen on for connections. The list should contain IP addresses (IPv4 or IPv6) or a single entry containing an asterisk (*). This indicates that the REST API should listen on all IP Addresses. • Type: Set(String) • Default value: "*"
port	The port on which the REST API should listen for requests. • Type: UInt • Default value: "9070"
Properties for the "snmp" section:	
allow	Restrict which IP addresses can access the SNMP command responder service. The value can be all, localhost, or a list of IP CIDR subnet masks. For example 10.100.0.0/16 would allow connections from any IP address beginning with 10.100. • Type: Set(String) • Default value: "all"
auth_password	The authentication password. Required (minimum length 8 characters) if security_level includes authentication. • Type: Password • Default value: <none>
bind_ip	The IP address the SNMP service should bind its listen port to. The value * (asterisk) means SNMP will listen on all IP addresses. • Type: String • Default value: "*"
community	The community string required for SNMPv1 and SNMPv2c commands. (If empty, all SNMPv1 and SNMPv2c commands will be rejected). • Type: String • Default value: "public"

Property	Description
enabled	Whether or not the SNMP command responder service should be enabled on this traffic manager. • Type: Boolean • Default value: false
hash_algorithm	The hash algorithm for authenticated SNMPv3 communications. • Type: Enum(String) • Default value: "md5" • Permitted values: "md5": MD5 "sha1": SHA-1
port	The port the SNMP command responder service should listen on. The value default denotes port 161 if the software is running with root privileges, and 1161 otherwise. • Type: String • Default value: "default"
priv_password	The privacy password. Required (minimum length 8 characters) if security_level includes privacy (message encryption). • Type: Password • Default value: <none>
security_level	The security level for SNMPv3 communications. • Type: Enum(String) • Default value: "noauthnopriv" • Permitted values: "authnopriv": Authentication only "authpriv": Authentication and Privacy "noauthnopriv": No Authentication, No Privacy
username	The username required for SNMPv3 commands. (If empty, all SNMPv3 commands will be rejected). • Type: String • Default value: <none>

TrafficScript Authenticator

URI Path: rule_authenticators

TrafficScript authenticators define remote authentication services that can be queried via a TrafficScript rule.

Property	Description
host	The hostname or IP address of the remote authenticator. • Type: String • Default value: <none>
note	A description of the authenticator. • Type: FreeformString • Default value: <none>
port	The port on which the remote authenticator should be contacted. • Type: UInt • Default value: "389"
Properties for the "ldap" section:	
attributes	A list of attributes to return from the search. If blank, no attributes will be returned. If set to '*' then all user attributes will be returned. • Type: Set(String) • Default value: <none>
bind_dn	The distinguished name (DN) of the 'bind' user. The traffic manager will connect to the LDAP server as this user when searching for user records. • Type: String • Default value: <none>
bind_password	The password for the bind user. • Type: Password • Default value: <none>
filter	The filter used to locate the LDAP record for the user being authenticated. Any occurrences of '%u' in the filter will be replaced by the name of the user being authenticated. • Type: String • Default value: <none>
filter_base_dn	The base distinguished name (DN) under which user records are located on the server. • Type: String • Default value: <none>
ssl_cert	The SSL certificate that the traffic manager should use to validate the remote server. If no certificate is specified then no signature validation will be performed. • Type: Reference(config-ssl-cacrl) • Default value: <none>

Property	Description
ssl_enabled	Whether or not to enable SSL encryption to the LDAP server. • Type: Boolean • Default value: false
ssl_type	The type of LDAP SSL encryption to use. • Type: Enum(String) • Default value: "ldaps" • Permitted values: "ldaps": LDAPS "starttls": Start TLS

User Authenticator

URI Path: user_authenticators

A user authenticator is used to allow access to the UI and REST API by querying a remote authentication service.

Property	Description
description	A description of the authenticator. • Type: String • Default value: <none>
enabled	Whether or not this authenticator is enabled. • Type: Boolean • Default value: false
type	The type and protocol used by this authentication service. • Type: Enum(String) • Default value: <none> • Permitted values: "ldap": LDAP "radius": RADIUS "tacacs_plus": TACACS+
Properties for the "ldap" section:	
base_dn	The base DN (Distinguished Name) under which directory searches will be applied. The entries for your users should all appear under this DN. An example of a typical base DN is: OU=users, DC=mycompany, DC=local • Type: String • Default value: <none>

Property	Description
bind_dn	Template to construct the bind DN (Distinguished Name) from the username. The string %u will be replaced by the username. Examples: %u@mycompany.local for Active Directory or cn=%u, dc=mycompany, dc=local for both LDAP and Active Directory. • Type: String • Default value: <none>
dn_method	The bind DN (Distinguished Name) for a user can either be searched for in the directory using the base distinguished name and filter values, or it can be constructed from the username. • Type: Enum(String) • Default value: "none" • Permitted values: "construct": Construct "none": No setting configured "search": Search
fallback_group	If the group attribute is not defined, or returns no results for the user logging in, the group named here will be used. If not specified, users will be denied access to the traffic manager if no groups matching a Permission Group can be found for them in the directory. • Type: String • Default value: <none>
filter	A filter that can be used to extract a unique user record located under the base DN (Distinguished Name). The string %u will be replaced by the username. This filter is used to find a user's bind DN when dn_method is set to "Search", and to extract group information if the group filter is not specified. Examples: sAMAccountName=%u for Active Directory, or uid=%u for some Unix LDAP schemas. • Type: String • Default value: <none>
group_attribute	The LDAP attribute that gives a user's group. If there are multiple entries for the attribute all will be extracted and they'll be lexicographically sorted, then the first one to match a Permission Group name will be used. • Type: String • Default value: <none>
group_field	The sub-field of the group attribute that gives a user's group. For example, if group_attribute is memberOf and this retrieves values of the form CN=mygroup, OU=groups, OU=users, DC=mycompany, DC=local you would set group_field to CN. If there are multiple matching fields only the first matching field will be used. • Type: String • Default value: <none>

Property	Description
group_filter	If the user record returned by filter does not contain the required group information you may specify an alternative group search filter here. This will usually be required if you have Unix/POSIX-style user records. If multiple records are returned the list of group names will be extracted from all of them. The string %u will be replaced by the username. Example: (&(memberUid=%u)(objectClass=posixGroup)) • Type: String • Default value: <none>
port	The port to connect to the LDAP server on. • Type: UInt • Default value: "389"
search_dn	The bind DN (Distinguished Name) to use when searching the directory for a user's bind DN. You can leave this blank if it is possible to perform the bind DN search using an anonymous bind. • Type: String • Default value: <none>
search_password	If binding to the LDAP server using search_dn requires a password, enter it here. • Type: Password • Default value: <none>
server	The IP or hostname of the LDAP server. • Type: String • Default value: <none>
timeout	Connection timeout in seconds. • Type: UInt • Default value: "30"
Properties for the "radius" section:	
fallback_group	If no group is found using the vendor and group identifiers, or the group found is not valid, the group specified here will be used. • Type: String • Default value: <none>
group_attribute	The RADIUS identifier for the attribute that specifies an account's group. May be left blank if fallback group is specified. • Type: UInt • Default value: "1"
group_vendor	The RADIUS identifier for the vendor of the RADIUS attribute that specifies an account's group. Leave blank if using a standard attribute (i.e. for Filter-Id set group_attribute to 11). • Type: UInt • Default value: "7146"
nas_identifier	This value is sent to the RADIUS server. • Type: String • Default value: <none>

Property	Description
nas_ip_address	This value is sent to the RADIUS server, if left blank the address of the interfaced used to connect to the server will be used. • Type: String • Default value: <none>
port	The port to connect to the RADIUS server on. • Type: UInt • Default value: "1812"
secret	Secret key shared with the RADIUS server. • Type: Password • Default value: <none>
server	The IP or hostname of the RADIUS server. • Type: String • Default value: <none>
timeout	Connection timeout in seconds. • Type: UInt • Default value: "30"
Properties for the "tacacs_plus" section:	
auth_type	Authentication type to use. • Type: Enum(String) • Default value: "pap" • Permitted values: "ascii": ASCII "pap": PAP
fallback_group	If group_service is not used, or no group value is provided for the user by the TACACS+ server, the group specified here will be used. If this is not specified, users with no TACACS+ defined group will be denied access. • Type: String • Default value: <none>
group_field	The TACACS+ "service" field that provides each user's group. • Type: String • Default value: "permission-group"
group_service	The TACACS+ "service" that provides each user's group field. • Type: String • Default value: "zeus"
port	The port to connect to the TACACS+ server on. • Type: UInt • Default value: "49"
secret	Secret key shared with the TACACS+ server. • Type: Password • Default value: <none>

Property	Description
server	The IP or hostname of the TACACS+ server. • Type: String • Default value: <none>
timeout	Connection timeout in seconds. • Type: UInt • Default value: "30"

User Group

URI Path: user_groups

Permission groups specify permissions for groups of users. These groups can be given read-write or read-only access to different parts of the configuration hierarchy. Each group will contain a table of permissions. Each table entry has a name that corresponds to a part of the configuration hierarchy, and a corresponding access level. The access level may have values of either none, ro (read only, this is the default), or full. Some permissions have sub-permissions, these are denoted by following the parent permission name with a colon (:) followed by the sub-permission name. The built-in admin group has a special permission key of all with the value full, this must not be altered for the admin group but can be used in other group configuration files to change the default permission level for the group.

Property	Description
description	A description for the group. • Type: String • Default value: <none>
password_expire_time	Members of this group must renew their passwords after this number of days. To disable password expiry for the group set this to 0 (zero). Note that this setting applies only to local users. • Type: UInt • Default value: <none>
permissions	A table defining which level of permission this group has for specific configuration elements. • Primary key: – name (String): Configuration element to which this group has a level of permission. • Sub keys: – access_level (String): Permission level for the configuration element (none, ro or full)
timeout	Inactive UI sessions will timeout after this number of seconds. To disable inactivity timeouts for the group set this to 0 (zero). • Type: UInt • Default value: "30"

Virtual Server

URI Path: virtual_servers

The conf/vservers directory contains configuration files that define virtual servers. The name of a file is the name of the virtual server it defines. Virtual servers can be configured under the Services > Virtual Servers section of the Admin Server UI or by using functions under the VirtualServer section of the SOAP API and CLI.

Property	Description
add_cluster_ip	Whether or not the virtual server should add an "X-Cluster-Client-Ip" header to the request that contains the remote client's IP address. • Type: Boolean • Default value: false
add_x_forwarded_for	Whether or not the virtual server should append the remote client's IP address to the X-Forwarded-For header. If the header does not exist, it will be added. • Type: Boolean • Default value: false
add_x_forwarded_proto	Whether or not the virtual server should add an "X-Forwarded-Proto" header to the request that contains the original protocol used by the client to connect to the traffic manager. • Type: Boolean • Default value: false
autodetect_upgrade_headers	Whether the traffic manager should check for HTTP responses that confirm an HTTP connection is transitioning to the WebSockets protocol. If that such a response is detected, the traffic manager will cease any protocol-specific processing on the connection and just pass incoming data to the client/server as appropriate. • Type: Boolean • Default value: false
bandwidth_class	The bandwidth management class that this server should use, if any. • Type: Reference(config-bandwidth) • Default value: <none>
bypass_data_plane_acceleration	Whether this service should, where possible, bypass data plane acceleration mechanisms. • Type: Boolean • Default value: false
close_with_rst	Whether or not connections from clients should be closed with a RST packet, rather than a FIN packet. This avoids the TIME_WAIT state, which on rare occasions allows wandering duplicate packets to be safely ignored. • Type: Boolean • Default value: false
completionrules	Rules that are run at the end of a transaction, in order, comma separated. • Type: List(String) • Default value: <none>

Property	Description
connect_timeout	The time, in seconds, to wait for data from a new connection. If no data is received within this time, the connection will be closed. A value of 0 (zero) will disable the timeout. • Type: UInt • Default value: "10"
enabled	Whether the virtual server is enabled. • Type: Boolean • Default value: false
ftp_force_server_secure	Whether or not the virtual server should require that incoming FTP data connections from the nodes originate from the same IP address as the node. • Type: Boolean • Default value: false
glb_services	The associated GLB services for this DNS virtual server. • Type: Set(String) • Default value: <none>
listen_on_any	Whether to listen on all IP addresses • Type: Boolean • Default value: false
listen_on_hosts	Hostnames and IP addresses to listen on • Type: Set(String) • Default value: <none>
listen_on_traffic_ips	Traffic IP Groups to listen on • Type: Set(String) • Default value: <none>
max_concurrent_connections	The maximum number of concurrent TCP connections that will be handled by this virtual server. If set to a non-zero value, the traffic manager will limit the number of concurrent TCP connections that this virtual server will accept to the value specified. When the limit is reached, new connections to this virtual server will not be accepted. If set to 0 the number of concurrent TCP connections will not be limited. • Type: UInt • Default value: <none>
note	A description for the virtual server. • Type: FreeformString • Default value: <none>
pool	The default pool to use for traffic. • Type: Reference(config-pool) • Default value: <none>
port	The port on which to listen for incoming connections. • Type: UInt • Default value: <none>

Property	Description
protection_class	The service protection class that should be used to protect this server, if any. • Type: String • Default value: <none>
protocol	The protocol that the virtual server is using. • Type: Enum(String) • Default value: "http" • Permitted values: "client_first": Generic client first "dns": DNS (UDP) "dns_tcp": DNS (TCP) "ftp": FTP "http": HTTP "https": SSL (HTTPS) "imaps": SSL (IMAPS) "imapv2": IMAPv2 "imapv3": IMAPv3 "imapv4": IMAPv4 "l4accel_dns": L4Accel DNS "l4accel_generic": L4Accel Generic "l4accel_stateless": L4Accel Stateless "l4accel_tcp": L4Accel TCP "l4accel_udp": L4Accel UDP "ldap": LDAP "ldaps": SSL (LDAPS) "pop3": POP3 "pop3s": SSL (POP3S) "rtsp": RTSP "server_first": Generic server first "siptcp": SIP (TCP) "sipudp": SIP (UDP) "smtp": SMTP "ssl": SSL "stream": Generic streaming "telnet": Telnet "udp": UDP "udpstreaming": UDP - Streaming
proxy_protocol	Expect connections to the traffic manager to be prefixed with a PROXY protocol header. If enabled, the information contained in the PROXY header will be available in TrafficScript. Connections that are not prefixed with a valid PROXY protocol header will be discarded. • Type: Boolean • Default value: false

Property	Description
request_rules	Rules to be applied to incoming requests, in order, comma separated. • Type: List(String) • Default value: <none>
response_rules	Rules to be applied to responses, in order, comma separated. • Type: List(Reference(config-trafficscript)) • Default value: <none>
slm_class	The service level monitoring class that this server should use, if any. • Type: Reference(config-slm) • Default value: <none>
so_nagle	Whether or not Nagle's algorithm should be used for TCP connections. • Type: Boolean • Default value: false
ssl_client_cert_headers	What HTTP headers the virtual server should add to each request to show the data in the client certificate. • Type: Enum(String) • Default value: "none" • Permitted values: "all": Certificate fields and certificate text "none": No data "simple": Certificate fields
ssl_decrypt	Whether or not the virtual server should decrypt incoming SSL traffic. • Type: Boolean • Default value: false
ssl_honor_fallback_scsv	Whether or not the Fallback SCSV sent by TLS clients is honored by this virtual server. Choosing the global setting means the value of configuration key ssl!honor_fallback_scsv from the Global Settings section of the System tab will be enforced. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable Fallback SCSV "enabled": Enable Fallback SCSV "use_default": Use the global setting for Fallback SCSV
strip_x_forwarded_proto	Whether or not the virtual server should strip the 'X-Forwarded-Proto' header from incoming requests. • Type: Boolean • Default value: false
transparent	Whether or not bound sockets should be configured for transparent proxying. • Type: Boolean • Default value: false

Property	Description
udp_end_transaction	When the traffic manager should consider a UDP transaction to have ended. • Type: Enum(String) • Default value: "one_response" • Permitted values: "match_requests": When the number of responses matches the number of requests "one_response": After one response "timeout": When they time out
Properties for the "aptimizer" section:	
enabled	Whether the virtual server should optimize web content. • Type: Boolean • Default value: false
profile	A table of Aptimizer profiles and the application scopes that apply to them. • Primary key: – name (String): The name of an Aptimizer acceleration profile. • Sub keys: – urls (Set(String)): The application scopes which apply to the acceleration profile.
Properties for the "connection" section:	
keepalive	Whether or not the virtual server should use keepalive connections with the remote clients. • Type: Boolean • Default value: false
keepalive_timeout	The length of time that the virtual server should keep an idle keepalive connection before discarding it. A value of 0 (zero) will mean that the keepalives are never closed by the traffic manager. • Type: UInt • Default value: "10"
max_client_buffer	The amount of memory, in bytes, that the virtual server should use to store data sent by the client. Larger values will use more memory, but will minimise the number of read() and write() system calls that the traffic manager must perform. • Type: UInt • Default value: "65536"
max_server_buffer	The amount of memory, in bytes, that the virtual server should use to store data returned by the server. Larger values will use more memory, but will minimise the number of read() and write() system calls that the traffic manager must perform. • Type: UInt • Default value: "65536"

Property	Description
max_transaction_duration	The total amount of time a transaction can take, counted from the first byte being received until the transaction is complete. For HTTP, this can mean all data has been written in both directions, or the connection has been closed; in most other cases it is the same as the connection being closed. The default value of 0 means there is no maximum duration, i.e., transactions can take arbitrarily long if none of the other timeouts occur. • Type: UInt • Default value: <none>
server_first_banner	If specified, the traffic manager will use the value as the banner to send for server-first protocols such as FTP, POP, SMTP and IMAP. This allows rules to use the first part of the client data (such as the username) to select a pool. The banner should be in the correct format for the protocol, e.g. for FTP it should start with "220 " • Type: String • Default value: <none>
timeout	A connection should be closed if no additional data has been received for this period of time. A value of 0 (zero) will disable this timeout. • Type: UInt • Default value: "300"
Properties for the "connection_errors" section:	
error_file	The error message to be sent to the client when the traffic manager detects an internal or backend error for the virtual server. • Type: Reference(config-extra-file) • Default value: "Default"
Properties for the "cookie" section:	
domain	The way in which the traffic manager should rewrite the domain portion of any cookies set by a back-end web server. • Type: Enum(UInt) • Default value: "no_rewrite" • Permitted values: "no_rewrite": Do not rewrite the domain "set_to_named": Rewrite the domain to the named domain value "set_to_request": Rewrite the domain to the host header of the request
new_domain	The domain to use when rewriting a cookie's domain to a named value. • Type: String • Default value: <none>
path_regex	If you wish to rewrite the path portion of any cookies set by a back-end web server, provide a regular expression to match the path: • Type: String • Default value: <none>

Property	Description
path_replace	If cookie path regular expression matches, it will be replaced by this substitution. Parameters $1-$9 can be used to represent bracketed parts of the regular expression. • Type: String • Default value: <none>
secure	Whether or not the traffic manager should modify the "secure" tag of any cookies set by a back-end web server. • Type: Enum(UInt) • Default value: "no_modify" • Permitted values: "no_modify": Do not modify the 'secure' tag "set_secure": Set the 'secure' tag "unset_secure": Unset the 'secure' tag
Properties for the "dns" section:	
edns_client_subnet	Enable/Disable use of EDNS client subnet option • Type: Boolean • Default value: false
edns_udpsize	EDNS UDP size advertised in responses. • Type: UInt • Default value: "4096"
max_udpsize	Maximum UDP answer size. • Type: UInt • Default value: "4096"
rrset_order	Response record ordering. • Type: Enum(String) • Default value: "fixed" • Permitted values: "cyclic": Cyclic "fixed": Fixed
verbose	Whether or not the DNS Server should emit verbose logging. This is useful for diagnosing problems. • Type: Boolean • Default value: false
zones	The DNS zones • Type: Set(String) • Default value: <none>
Properties for the "ftp" section:	

Property	Description
data_source_port	The source port to be used for active-mode FTP data connections. If 0, a random high port will be used, otherwise the specified port will be used. If a port below 1024 is required you must first explicitly permit use of low ports with the data_bind_low global setting. • Type: UInt • Default value: <none>
force_client_secure	Whether or not the virtual server should require that incoming FTP data connections from the client originate from the same IP address as the corresponding client control connection. • Type: Boolean • Default value: false
port_range_high	If non-zero, then this controls the upper bound of the port range to use for FTP data connections. • Type: UInt • Default value: <none>
port_range_low	If non-zero, then this controls the lower bound of the port range to use for FTP data connections. • Type: UInt • Default value: <none>
ssl_data	Use SSL on the data connection as well as the control connection (if not enabled it is left to the client and server to negotiate this). • Type: Boolean • Default value: false
Properties for the "gzip" section:	
compress_level	Compression level (1-9, 1=low, 9=high). • Type: UInt • Default value: "1"
enabled	Compress web pages sent back by the server. • Type: Boolean • Default value: false
etag_rewrite	How the ETag header should be manipulated when compressing content. • Type: Enum(String) • Default value: "wrap" • Permitted values: "delete": Delete the ETag header "ignore": Leave the ETag unchanged "weaken": Change the ETag header to specify a weak match "wrap": Wrap the ETag, and attempt to unwrap safe conditional requests
include_mime	MIME types to compress. Complete MIME types can be used, or a type can end in a '*' to match multiple types. • Type: Set(String) • Default value: "text/html text/plain"

Property	Description
max_size	Maximum document size to compress (0 means unlimited). • Type: UInt • Default value: "10000000"
min_size	Minimum document size to compress. • Type: UInt • Default value: "1000"
no_size	Compress documents with no given size. • Type: Boolean • Default value: false
Properties for the "http" section:	
chunk_overhead_forwarding	Handling of HTTP chunk overhead. When vTM receives data from a server or client that consists purely of protocol overhead (contains no payload), forwarding of such segments is delayed until useful payload data arrives (setting "lazy"). Changing this key to "eager" will make vTM incur the overhead of immediately passing such data on; it should only be used with HTTP peers whose chunk handling requires it. • Type: Enum(String) • Default value: "lazy" • Permitted values: "eager": Forward all data, even when no new payload information is available. "lazy": Only forward segments when useful payload data is available.
location_regex	If the 'Location' header matches this regular expression, rewrite the header using the 'location_replace' pattern. • Type: String • Default value: <none>
location_replace	If the 'Location' header matches the 'location_regex' regular expression, rewrite the header with this pattern (parameters such as $1-$9 can be used to match parts of the regular expression): • Type: String • Default value: <none>
location_rewrite	The action the virtual server should take if the "Location" header does not match the location_regex regular expression. • Type: Enum(UInt) • Default value: "if_host_matches" • Permitted values: "always": Rewrite the hostname to the request's "Host" header, and rewrite the protocol and port if necessary; "if_host_matches": Do not rewrite the hostname. Rewrite the protocol and port if the hostname matches the request's "Host" header. "never": Nothing;

Property	Description
mime_default	Auto-correct MIME types if the server sends the "default" MIME type for files. • Type: String • Default value: "text/plain"
mime_detect	Auto-detect MIME types if the server does not provide them. • Type: Boolean • Default value: false
Properties for the "http2" section:	
connect_timeout	The time, in seconds, to wait for a request on a new HTTP/2 connection. If no request is received within this time, the connection will be closed. This setting overrides the connect_timeout setting. If set to 0 (zero), the value of connect_timeout will be used instead. • Type: UInt • Default value: <none>
data_frame_size	This setting controls the preferred frame size used when sending body data to the client. If the client specifies a smaller maximum size than this setting, the client's maximum size will be used. Every data frame sent has at least a 9-byte header, in addition to this frame size, prepended to it. • Type: UInt • Default value: "4096"
enabled	This setting allows the HTTP/2 protocol to be used by a HTTP virtual server. Unless use of HTTP/2 is negotiated by the client, the virtual server will fall back to HTTP 1.x automatically. • Type: Boolean • Default value: false
header_table_size	This setting controls the amount of memory allowed for header compression on each HTTP/2 connection. • Type: UInt • Default value: "4096"
headers_index_blacklist	A list of header names that should never be compressed using indexing. • Type: Set(String) • Default value: <none>
headers_index_default	The HTTP/2 HPACK compression scheme allows for HTTP headers to be compressed using indexing. Sensitive headers can be marked as "never index", which prevents them from being compressed using indexing. When this setting is Yes, only headers included in http2!headers_index_blacklist are marked as "never index". When this setting is No, all headers will be marked as "never index" unless they are included in http2!headers_index_whitelist. • Type: Boolean • Default value: false
headers_index_whitelist	A list of header names that can be compressed using indexing when the value of http2!headers_index_default is set to No. • Type: Set(String) • Default value: <none>

Property	Description
headers_size_limit	The maximum size, in bytes, of decompressed headers for an HTTP/2 request. If the limit is exceeded, the connection on which the request was sent will be dropped. A value of 0 disables the limit check. If a service protection class with http!max_header_length configured is associated with this service then that setting will take precedence. • Type: UInt • Default value: "262144"
idle_timeout_no_streams	The time, in seconds, to wait for a new HTTP/2 request on a previously used HTTP/2 connection that has no open HTTP/2 streams. If an HTTP/2 request is not received within this time, the connection will be closed. A value of 0 (zero) will disable the timeout. • Type: UInt • Default value: "120"
idle_timeout_open_streams	The time, in seconds, to wait for data on an idle HTTP/2 connection, which has open streams, when no data has been sent recently (e.g. for long-polled requests). If data is not sent within this time, all open streams and the HTTP/2 connection will be closed. A value of 0 (zero) will disable the timeout. • Type: UInt • Default value: "600"
max_concurrent_streams	This setting controls the number of streams a client is permitted to open concurrently on a single connection. • Type: UInt • Default value: "200"
max_frame_size	This setting controls the maximum HTTP/2 frame size clients are permitted to send to the traffic manager. • Type: UInt • Default value: "16384"
max_header_padding	The maximum size, in bytes, of the random-length padding to add to HTTP/2 header frames. The padding, a random number of zero bytes up to the maximum specified. • Type: UInt • Default value: <none>
merge_cookie_headers	Whether Cookie headers received from an HTTP/2 client should be merged into a single Cookie header using RFC6265 rules before forwarding to an HTTP/1.1 server. Some web applications do not handle multiple Cookie headers correctly. • Type: Boolean • Default value: false
stream_window_size	This setting controls the flow control window for each HTTP/2 stream. This will limit the memory used for buffering when the client is sending body data faster than the pool node is reading it. • Type: UInt • Default value: "65535"
Properties for the "kerberos_protocol_transition" section:	

Property	Description
enabled	Whether or not the virtual server should use Kerberos Protocol Transition. • Type: Boolean • Default value: false
principal	The Kerberos principal this virtual server should use to perform Kerberos Protocol Transition. • Type: String • Default value: <none>
target	The Kerberos principal name of the service this virtual server targets. • Type: String • Default value: <none>
Properties for the "l4accel" section:	
rst_on_service_failure	Whether the virtual server should send a TCP RST packet or ICMP error message if a service is unavailable, or if an established connection to a node fails. • Type: Boolean • Default value: false
service_ip_snat	Whether or not backend connections should be configured to use the ingress service IP as the source IP for the back-end connection when Source NAT is enabled for the pool used by the service. Requires l4accel!state_sync to be enabled. • Type: Boolean • Default value: false
state_sync	Whether the state of active connections will be synchronized across the cluster for L4Accel services, such that connections will persist in the event of a failover. Note that the service must listen only on Traffic IP groups for this setting to be enabled. • Type: Boolean • Default value: false
tcp_msl	The maximum segment lifetime, in seconds, of a TCP segment being handled by the traffic manager. This setting determines for how long information about a connection will be retained after receiving a two-way FIN or RST. • Type: UInt • Default value: "8"
timeout	The number of seconds after which a connection will be closed if no further packets have been received on it. • Type: UInt • Default value: "1800"
udp_count_requests	Whether a connection should be closed when the number of UDP response datagrams received from the server is equal to the number of request datagrams that have been sent by the client. If set to No the connection will be closed after the first response has been received from the server. This setting takes precedence over l4accel!optimized_aging setting. • Type: Boolean • Default value: false

Property	Description
Properties for the "l4stateless" section:	
Properties for the "log" section:	
client_connection_failures	Should the virtual server log failures occurring on connections to clients. • Type: Boolean • Default value: false
enabled	Whether or not to log connections to the virtual server to a disk on the file system. • Type: Boolean • Default value: false
filename	The name of the file in which to store the request logs. The filename can contain macros which will be expanded by the traffic manager to generate the full filename. • Type: String • Default value: "%zeushome%/zxtm/log/%v.log"
format	The log file format. This specifies the line of text that will be written to the log file when a connection to the traffic manager is completed. Many parameters from the connection can be recorded using macros. • Type: String • Default value: "%h %l %u %t "%r" %s %b "%{Referer}i" "%{User-agent}i""
save_all	Whether to log all connections by default, or log no connections by default. Specific connections can be selected for addition to or exclusion from the log using the TrafficScript function requestlog.include(). • Type: Boolean • Default value: false
server_connection_failures	Should the virtual server log failures occurring on connections to nodes. • Type: Boolean • Default value: false
session_persistence_verbose	Should the virtual server log session persistence events. • Type: Boolean • Default value: false
ssl_failures	Should the virtual server log failures occurring on SSL secure negotiation. • Type: Boolean • Default value: false
Properties for the "recent_connections" section:	
enabled	Whether or not connections handled by this virtual server should be shown on the Activity > Connections page. • Type: Boolean • Default value: false

Property	Description
save_all	Whether or not all connections handled by this virtual server should be shown on the Connections page. Individual connections can be selectively shown on the Connections page using the recentconns.include() TrafficScript function. • Type: Boolean • Default value: false
Properties for the "request_tracing" section:	
enabled	Record a trace of major connection processing events for each request and response. • Type: Boolean • Default value: false
trace_io	Include details of individual I/O events in request and response traces. Requires request tracing to be enabled. • Type: Boolean • Default value: false
Properties for the "rtsp" section:	
streaming_port_range_high	If non-zero this controls the upper bound of the port range to use for streaming data connections. • Type: UInt • Default value: <none>
streaming_port_range_low	If non-zero this controls the lower bound of the port range to use for streaming data connections. • Type: UInt • Default value: <none>
streaming_timeout	If non-zero data-streams associated with RTSP connections will timeout if no data is transmitted for this many seconds. • Type: UInt • Default value: "30"
Properties for the "sip" section:	
dangerous_requests	The action to take when a SIP request with body data arrives that should be routed to an external IP. • Type: Enum(String) • Default value: "node" • Permitted values: "forbid": Send a 403 Forbidden response to the client "forward": Forward the request to its target URI (dangerous) "node": Send the request to a back-end node
follow_route	Should the virtual server follow routing information contained in SIP requests. If set to No requests will be routed to the chosen back-end node regardless of their URI or Route header. • Type: Boolean • Default value: false

Property	Description
max_connection_mem	SIP clients can have several pending requests at one time. To protect the traffic manager against DoS attacks, this setting limits the amount of memory each client can use. When the limit is reached new requests will be sent a 413 response. If the value is set to 0 (zero) the memory limit is disabled. • Type: UInt • Default value: "65536"
mode	The mode that this SIP virtual server should operate in. • Type: Enum(String) • Default value: "sip_gateway" • Permitted values: "full_gateway": All SIP requests and responses and all session data will pass through vTM. A port range to use for the session data and a timeout value for inactive data connections can be specified in the additional settings that are displayed when the Full Gateway mode is selected. "route": The first SIP request in a session will pass through vTM, along with its responses, but all future requests that are part of the same session will go directly to the back-end node that was chosen by the traffic manager. "sip_gateway": All SIP requests and responses will pass through the traffic manager.
rewrite_uri	Replace the Request-URI of SIP requests with the address of the selected back-end node. • Type: Boolean • Default value: false
streaming_port_range_high	If non-zero this controls the upper bound of the port range to use for streaming data connections. • Type: UInt • Default value: <none>
streaming_port_range_low	If non-zero, then this controls the lower bound of the port range to use for streaming data connections. • Type: UInt • Default value: <none>
streaming_timeout	If non-zero a UDP stream will timeout when no data has been seen within this time. • Type: UInt • Default value: "60"
timeout_messages	When timing out a SIP transaction, send a 'timed out' response to the client and, in the case of an INVITE transaction, a CANCEL request to the server. • Type: Boolean • Default value: false
transaction_timeout	The virtual server should discard a SIP transaction when no further messages have been seen within this time. • Type: UInt • Default value: "30"
Properties for the "smtp" section:	

Property	Description
expect_starttls	Whether or not the traffic manager should expect the connection to start off in plain text and then upgrade to SSL using STARTTLS when handling SMTP traffic. • Type: Boolean • Default value: false
Properties for the "ssl" section:	
add_http_headers	Whether or not the virtual server should add HTTP headers to each request to show the SSL connection parameters. • Type: Boolean • Default value: false
client_cert_cas	The certificate authorities that this virtual server should trust to validate client certificates. If no certificate authorities are selected, and client certificates are requested, then all client certificates will be accepted. • Type: Set(String) • Default value: <none>
elliptic_curves	The SSL elliptic curve preference list for SSL connections to this virtual server using TLS version 1.0 or higher. Leaving this empty will make the virtual server use the globally configured curve preference list. The named curves P256, P384 and P521 may be configured. • Type: List(String) • Default value: <none>
issued_certs_never_expire	When the virtual server verifies certificates signed by these certificate authorities, it doesn't check the 'not after' date, i.e., they are considered valid even after their expiration date has passed (but not if they have been revoked). • Type: Set(String) • Default value: <none>
issued_certs_never_expire_depth	This setting gives the number of certificates in a certificate chain beyond those listed as issued_certs_never_expire whose certificate expiry will not be checked. For example "0" will result in the expiry checks being made for certificates issued by issued_certs_never_expire certificates, "1" will result in no expiry checks being performed for the certificates directly issued by issued_certs_never_expire certificates, "2" will avoid checking expiry for certificates issued by certificates issued by the issued_certs_never_expire certificates as well, and so on. • Type: UInt • Default value: "1"
ocsp_enable	Whether or not the traffic manager should use OCSP to check the revocation status of client certificates. • Type: Boolean • Default value: false

Property	Description
ocsp_issuers	A table of certificate issuer specific OCSP settings. • Primary key: – issuer (String): The name of an issuer (or DEFAULT for default OCSP settings). • Sub keys: – aia (Boolean): Whether the traffic manager should use AIA information contained in a client certificate to determine which OCSP responder to contact. – nonce (Enum(String)): How to use the OCSP nonce extension, which protects against OCSP replay attacks. Some OCSP servers do not support nonces. Permitted values: "off": No nonce check "on": Use nonce, server does not have to reply with nonce "strict": Use nonce, server must reply with nonce – required (Enum(String)): Whether we should do an OCSP check for this issuer, and whether it is required or optional. Permitted values: "none": None "optional": OCSP check optional "strict": OCSP check required – responder_cert (String): The expected responder certificate. – signer (String): The certificate with which to sign the request, if any. – url (String): Which OCSP responders this virtual server should use to verify client certificates.
ocsp_max_response_age	The number of seconds for which an OCSP response is considered valid if it has not yet exceeded the time specified in the 'nextUpdate' field. If set to 0 (zero) then OCSP responses are considered valid until the time specified in their 'nextUpdate' field. • Type: UInt • Default value: <none>
ocsp_stapling	If OCSP URIs are present in certificates used by this virtual server, then enabling this option will allow the traffic manager to provide OCSP responses for these certificates as part of the handshake, if the client sends a TLS status_request extension in the ClientHello. • Type: Boolean • Default value: false
ocsp_time_tolerance	The number of seconds outside the permitted range for which the 'thisUpdate' and 'nextUpdate' fields of an OCSP response are still considered valid. • Type: UInt • Default value: "30"
ocsp_timeout	The number of seconds after which OCSP requests will be timed out. • Type: UInt • Default value: "10"

Property	Description
prefer_sslv3	Deprecated. Formerly allowed a preference for SSLv3 for performance reasons. • Type: Boolean • Default value: false
request_client_cert	Whether or not the virtual server should request an identifying SSL certificate from each client. • Type: Enum(UInt) • Default value: "dont_request" • Permitted values: "dont_request": Do not request a client certificate "request": Request, but do not require a client certificate "require": Require a client certificate
send_close_alerts	Whether or not to send an SSL/TLS "close alert" when the traffic manager is initiating an SSL socket disconnection. • Type: Boolean • Default value: false
server_cert_alt_certificates	The SSL certificates and corresponding private keys. • Type: List(String) • Default value: <none>
server_cert_default	The default SSL certificate to use for this virtual server. • Type: String • Default value: <none>
server_cert_host_mapping	Host specific SSL server certificate mappings. • Primary key: – host (String): Host which this entry refers to. • Sub keys: – certificate (String): The SSL server certificate for a particular destination site IP. – alt_certificates (List(String)): The SSL server certificates for a particular destination site IP.
signature_algorithms	The SSL signature algorithms preference list for SSL connections to this virtual server using TLS version 1.2 or higher. Leaving this empty will make the virtual server use the globally configured preference list, signature_algorithms in the ssl section of the global_settings resource. See there and in the online help for how to specify SSL signature algorithms. • Type: String • Default value: <none>
ssl_ciphers	The SSL/TLS ciphers to allow for connections to this virtual server. Leaving this empty will make the virtual server use the globally configured ciphers, see configuration key ssl!ssl3_ciphers in the Global Settings section of the System tab. See there for how to specify SSL/TLS ciphers. • Type: String • Default value: <none>

Property	Description
ssl_support_ssl2	No longer supported. Formerly controlled whether SSLv2 could be used for SSL connections to this virtual server. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable SSLv2 "enabled": Enable SSLv2 (not recommended) "use_default": Use the global setting for SSLv2
ssl_support_ssl3	Whether or not SSLv3 is enabled for this virtual server. Choosing the global setting means the value of configuration key ssl!support_ssl3 from the Global Settings section of the System tab will be enforced. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable SSLv3 "enabled": Enable SSLv3 "use_default": Use the global setting for SSLv3
ssl_support_tls1	Whether or not TLSv1.0 is enabled for this virtual server. Choosing the global setting means the value of configuration key ssl!support_tls1 from the Global Settings section of the System tab will be enforced. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable TLSv1.0 "enabled": Enable TLSv1.0 "use_default": Use the global setting for TLSv1.0
ssl_support_tls1_1	Whether or not TLSv1.1 is enabled for this virtual server. Choosing the global setting means the value of configuration key ssl!support_tls1.1 from the Global Settings section of the System tab will be enforced. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable TLSv1.1 "enabled": Enable TLSv1.1 "use_default": Use the global setting for TLSv1.1

Property	Description
ssl_support_tls1_2	Whether or not TLSv1.2 is enabled for this virtual server. Choosing the global setting means the value of configuration key ssl!support_tls1.2 from the Global Settings section of the System tab will be enforced. • Type: Enum(String) • Default value: "use_default" • Permitted values: "disabled": Disable TLSv1.2 "enabled": Enable TLSv1.2 "use_default": Use the global setting for TLSv1.2
trust_magic	If the traffic manager is receiving traffic sent from another traffic manager, then enabling this option will allow it to decode extra information on the true origin of the SSL connection. This information is supplied by the first traffic manager. • Type: Boolean • Default value: false
Properties for the "syslog" section:	
enabled	Whether or not to log connections to the virtual server to a remote syslog host. • Type: Boolean • Default value: false
format	The log format for the remote syslog. This specifies the line of text that will be sent to the remote syslog when a connection to the traffic manager is completed. Many parameters from the connection can be recorded using macros. • Type: String • Default value: "%h %l %u %t "%r" %s %b "%{Referer}i" "%{User-agent}i""
ip_end_point	The remote host and port (default is 514) to send request log lines to. • Type: String • Default value: <none>
msg_len_limit	Maximum length in bytes of a message sent to the remote syslog. Messages longer than this will be truncated before they are sent. • Type: UInt • Default value: "1024"
Properties for the "tcp" section:	
proxy_close	If set to Yes the traffic manager will send the client FIN to the back-end server and wait for a server response instead of closing the connection immediately. This is only necessary for protocols that require half-close support to function correctly, such as "rsh". If the traffic manager is responding to the request itself, setting this key to Yes will cause the traffic manager to continue writing the response even after it has received a FIN from the client. • Type: Boolean • Default value: false
Properties for the "transaction_export" section:	

Property	Description
brief	Whether to export a restricted set of metadata about transactions processed by this virtual server. If enabled, more verbose information such as client and server headers and request tracing events will be omitted from the exported data. • Type: Boolean • Default value: false
enabled	Export metadata about transactions handled by this service to the globally configured endpoint. Data will be exported only if the global transaction_export!enabled setting is enabled. • Type: Boolean • Default value: false
hi_res	Whether the transaction processing timeline included in the metadata export is recorded with a high, microsecond, resolution. If set to No, timestamps will be recorded with a resolution of milliseconds. • Type: Boolean • Default value: false
http_header_blacklist	The set of HTTP header names for which corresponding values should be redacted from the metadata exported by this virtual server. • Type: Set(String) • Default value: "Authorization"
Properties for the "udp" section:	
end_point_persistence	Whether UDP datagrams received from the same IP address and port are sent to the same pool node if they match an existing UDP session. Sessions are defined by the protocol being handled, for example SIP datagrams are grouped based on the value of the Call-ID header. • Type: Boolean • Default value: false
port_smp	Whether or not UDP datagrams should be distributed across all traffic manager processes. This setting is not recommended if the traffic manager will be handling connection-based UDP protocols. • Type: Boolean • Default value: false
response_datagrams_expected	The virtual server should discard any UDP connection and reclaim resources when the node has responded with this number of datagrams. For simple request/response protocols this can be often set to 1. If set to -1, the connection will not be discarded until the timeout is reached. • Type: Int • Default value: "1"
timeout	The virtual server should discard any UDP connection and reclaim resources when no further UDP traffic has been seen within this time. • Type: UInt • Default value: "7"
Properties for the "web_cache" section:	

Property	Description
control_out	The "Cache-Control" header to add to every cached HTTP response, no-cache or max-age=600 for example. • Type: String • Default value: <none>
enabled	If set to Yes the traffic manager will attempt to cache web server responses. • Type: Boolean • Default value: false
error_page_time	Time period to cache error pages for. • Type: UInt • Default value: "30"
max_time	Maximum time period to cache web pages for. • Type: UInt • Default value: "600"
refresh_time	If a cached page is about to expire within this time, the traffic manager will start to forward some new requests on to the web servers. A maximum of one request per second will be forwarded; the remainder will continue to be served from the cache. This prevents "bursts" of traffic to your web servers when an item expires from the cache. Setting this value to 0 will stop the traffic manager updating the cache before it expires. • Type: UInt • Default value: "2"

Web Accelerator Profile

URI Path: aptimizer/profiles

A Web Accelerator profile can be applied to an HTTP virtual server to enable automatic web content optimization.

Property	Description
background_after	If Web Accelerator can finish optimizing the resource within this time limit then serve the optimized content to the client, otherwise complete the optimization in the background and return the original content to the client. If set to 0, Web Accelerator will always wait for the optimization to complete before sending a response to the client. • Type: UInt • Default value: <none>
background_on_additional_resources	If a web page contains resources that have not yet been optimized, fetch and optimize those resources in the background and send a partially optimized web page to clients until all resources on that page are ready. • Type: Boolean • Default value: false

Property	Description
mode	Set the Web Accelerator mode to turn acceleration on or off. • Type: Enum(String) • Default value: "active" • Permitted values: "active": On - Web Accelerator acceleration is enabled "idle": Off - Acceleration is disabled, but requests for Web Accelerator resources are served "stealth": Stealth - Acceleration is controlled by a cookie
show_info_bar	Show the Web Accelerator information bar on optimized web pages. This requires HTML optimization to be enabled in the acceleration settings. • Type: Boolean • Default value: false

SNMP Counter Values

Actions

URI Path: statistics/actions/*

Actions statistics values.

Counter	Description
processed	Number of times this action has been processed. • Type: UInt • SNMP name: "actionsProcessed"

Asp session cache

URI Path: statistics/cache/asp_session_cache

Asp session cache statistics values.

Counter	Description
entries	The total number of ASP sessions stored in the cache. • Type: UInt • SNMP name: "aspSessionCacheEntries"
entries_max	The maximum number of ASP sessions in the cache. • Type: UInt • SNMP name: "aspSessionCacheEntriesMax"

Counter	Description
hit_rate	The percentage of ASP session lookups that succeeded. • Type: UInt • SNMP name: "aspSessionCacheHitRate"
hits	Number of times a ASP session entry has been successfully found in the cache. • Type: UInt • SNMP name: "aspSessionCacheHits"
lookups	Number of times a ASP session entry has been looked up in the cache. • Type: UInt • SNMP name: "aspSessionCacheLookups"
misses	Number of times a ASP session entry has not been available in the cache. • Type: UInt • SNMP name: "aspSessionCacheMisses"
oldest	The age of the oldest ASP session in the cache (in seconds). • Type: UInt • SNMP name: "aspSessionCacheOldest"

Bandwidth

URI Path: statistics/bandwidth/*

Bandwidth statistics values.

Counter	Description
bytes_drop	Bytes dropped by this bandwidth class. • Type: UInt64 • SNMP name: "bandwidthClassBytesDrop"
bytes_drop_hi	Bytes dropped by this bandwidth class (high 32bits). • Type: UInt • SNMP name: "bandwidthClassBytesDropHi"
bytes_drop_lo	Bytes dropped by this bandwidth class (low 32bits). • Type: UInt • SNMP name: "bandwidthClassBytesDropLo"
bytes_out	Bytes output by connections assigned to this bandwidth class. • Type: UInt64 • SNMP name: "bandwidthClassBytesOut"
bytes_out_hi	Bytes output by connections assigned to this bandwidth class (high 32bits). • Type: UInt • SNMP name: "bandwidthClassBytesOutHi"

Counter	Description
bytes_out_lo	Bytes output by connections assigned to this bandwidth class (low 32bits). • Type: UInt • SNMP name: "bandwidthClassBytesOutLo"
guarantee	Guaranteed bandwidth class limit (kbits/s). Currently unused. • Type: UInt • SNMP name: "bandwidthClassGuarantee"
maximum	Maximum bandwidth class limit (kbits/s). • Type: UInt • SNMP name: "bandwidthClassMaximum"
pkts_drop	Number of packets dropped by this bandwidth class. • Type: UInt64 • SNMP name: "bandwidthClassPktsDrop"
pkts_drop_hi	Number of packets dropped by this bandwidth class (high 32bits). • Type: UInt • SNMP name: "bandwidthClassPktsDropHi"
pkts_drop_lo	Number of packets dropped by this bandwidth class (low 32bits). • Type: UInt • SNMP name: "bandwidthClassPktsDropLo"

Cloud api credentials

URI Path: statistics/cloud_api_credentials/*

Cloud api credentials statistics values.

Counter	Description
node_creations	The number of instance creation API requests made with this set of cloud credentials. • Type: UInt • SNMP name: "cloudcredentialsNodeCreations"
node_deletions	The number of instance destruction API requests made with this set of cloud credentials. • Type: UInt • SNMP name: "cloudcredentialsNodeDeletions"
status_requests	The number of status API requests made with this set of cloud credentials. • Type: UInt • SNMP name: "cloudcredentialsStatusRequests"

Connection rate limit

URI Path: statistics/connection_rate_limit/*

Connection rate limit statistics values.

Counter	Description
conns_entered	Connections that have entered the rate class and have been queued. • Type: UInt • SNMP name: "rateClassConnsEntered"
conns_left	Connections that have left the rate class. • Type: UInt • SNMP name: "rateClassConnsLeft"
current_rate	The average rate that requests are passing through this rate class. • Type: UInt • SNMP name: "rateClassCurrentRate"
dropped	Requests dropped from this rate class without being processed (e.g. timeouts). • Type: UInt • SNMP name: "rateClassDropped"
max_rate_per_min	The maximum rate that requests may pass through this rate class (requests/min). • Type: UInt • SNMP name: "rateClassMaxRatePerMin"
max_rate_per_sec	The maximum rate that requests may pass through this rate class (requests/sec). • Type: UInt • SNMP name: "rateClassMaxRatePerSec"
queue_length	The current number of requests queued by this rate class. • Type: UInt • SNMP name: "rateClassQueueLength"

Events

URI Path: statistics/events/*

Events statistics values.

Counter	Description
matched	Number of times this event configuration has matched. • Type: UInt • SNMP name: "eventsMatched"

Glb services

URI Path: statistics/glb_services/*

Glb services statistics values.

Counter	Description
discarded	Number of A records this GLB Service has discarded. • Type: UInt • SNMP name: "glbServiceDiscarded"
responses	Number of A records this GLB Service has altered. • Type: UInt • SNMP name: "glbServiceResponses"
unmodified	Number of A records this GLB Service has passed through unmodified. • Type: UInt • SNMP name: "glbServiceUnmodified"

Globals

URI Path: statistics/globals

Globals statistics values.

Counter	Description
analytics_transactions_dropped	Count of transaction metadata records that have been dropped • Type: UInt • SNMP name: "analyticsTransactionsDropped"
analytics_transactions_exported	Count of transaction metadata records that have been exported • Type: UInt • SNMP name: "analyticsTransactionsExported"
analytics_transactions_memory_usage	Number of bytes queued in the transaction export transmit buffers. • Type: UInt • SNMP name: "analyticsTransactionsMemoryUsage"
data_entries	Number of entries in the TrafficScript data.get()/set() storage. • Type: UInt • SNMP name: "dataEntries"
data_memory_usage	Number of bytes used in the TrafficScript data.get()/set() storage. • Type: UInt • SNMP name: "dataMemoryUsage"

Counter	Description
events_seen	Events seen by the traffic Manager's event handling process. • Type: UInt • SNMP name: "eventsSeen"
hourly_peak_bytes_in_per_second	The peak bytes received from clients per second in the last hour. • Type: UInt • SNMP name: "hourlyPeakBytesInPerSecond"
hourly_peak_bytes_out_per_second	The peak bytes sent to clients per second in the last hour. • Type: UInt • SNMP name: "hourlyPeakBytesOutPerSecond"
hourly_peak_requests_per_second	The peak requests per second in the last hour. • Type: UInt • SNMP name: "hourlyPeakRequestsPerSecond"
hourly_peak_ssl_connections_per_second	The peak ssl connections per second in the last hour. • Type: UInt • SNMP name: "hourlyPeakSSLConnectionsPerSecond"
num_idle_connections	Total number of idle HTTP connections to all nodes (used for future HTTP requests). • Type: UInt • SNMP name: "numIdleConnections"
number_child_processes	The number of traffic manager child processes. • Type: UInt • SNMP name: "numberChildProcesses"
number_dnsa_cache_hits	Requests for DNS A records resolved from the traffic manager's local cache. • Type: UInt • SNMP name: "numberDNSACacheHits"
number_dnsa_requests	Requests for DNS A records (hostname->IP address) made by the traffic manager. • Type: UInt • SNMP name: "numberDNSARequests"
number_dnsptr_cache_hits	Requests for DNS PTR records resolved from the traffic manager's local cache. • Type: UInt • SNMP name: "numberDNSPTRCacheHits"
number_dnsptr_requests	Requests for DNS PTR records (IP address->hostname) made by the traffic manager. • Type: UInt • SNMP name: "numberDNSPTRRequests"
number_snmp_bad_requests	Malformed SNMP requests received. • Type: UInt • SNMP name: "numberSNMPBadRequests"

Counter	Description
number_snmp_get_bulk_requests	SNMP GetBulkRequests received. • Type: UInt • SNMP name: "numberSNMPGetBulkRequests"
number_snmp_get_next_requests	SNMP GetNextRequests received. • Type: UInt • SNMP name: "numberSNMPGetNextRequests"
number_snmp_get_requests	SNMP GetRequests received. • Type: UInt • SNMP name: "numberSNMPGetRequests"
number_snmp_unauthorised_requests	SNMP requests dropped due to access restrictions. • Type: UInt • SNMP name: "numberSNMPUnauthorisedRequests"
ssl_cipher_3des_decrypts	Bytes decrypted with 3DES. • Type: UInt • SNMP name: "sslCipher3DESDecrypts"
ssl_cipher_3des_encrypts	Bytes encrypted with 3DES. • Type: UInt • SNMP name: "sslCipher3DESEncrypts"
ssl_cipher_aes_decrypts	Bytes decrypted with AES. • Type: UInt • SNMP name: "sslCipherAESDecrypts"
ssl_cipher_aes_encrypts	Bytes encrypted with AES. • Type: UInt • SNMP name: "sslCipherAESEncrypts"
ssl_cipher_aes_gcm_decrypts	Bytes decrypted with AES-GCM. • Type: UInt • SNMP name: "sslCipherAESGCMDecrypts"
ssl_cipher_aes_gcm_encrypts	Bytes encrypted with AES-GCM. • Type: UInt • SNMP name: "sslCipherAESGCMEncrypts"
ssl_cipher_decrypts	Bytes decrypted with a symmetric cipher. • Type: UInt • SNMP name: "sslCipherDecrypts"
ssl_cipher_des_decrypts	Bytes decrypted with DES. • Type: UInt • SNMP name: "sslCipherDESDecrypts"
ssl_cipher_des_encrypts	Bytes encrypted with DES. • Type: UInt • SNMP name: "sslCipherDESEncrypts"

Counter	Description
ssl_cipher_dh_agreements	Number of Diffie Hellman key agreements. • Type: UInt • SNMP name: "sslCipherDHAgreements"
ssl_cipher_dh_generates	Number of Diffie Hellman keys generated. • Type: UInt • SNMP name: "sslCipherDHGenerates"
ssl_cipher_dsa_signs	Number of DSA signing operations. • Type: UInt • SNMP name: "sslCipherDSASigns"
ssl_cipher_dsa_verifies	Number of DSA verifications. • Type: UInt • SNMP name: "sslCipherDSAVerifies"
ssl_cipher_ecdh_agreements	Number of Elliptic Curve Diffie Hellman key agreements. • Type: UInt • SNMP name: "sslCipherECDHAgreements"
ssl_cipher_ecdh_generates	Number of Elliptic Curve Diffie Hellman keys generated. • Type: UInt • SNMP name: "sslCipherECDHGenerates"
ssl_cipher_ecdsa_signs	Number of ECDSA signing operations. • Type: UInt • SNMP name: "sslCipherECDSASigns"
ssl_cipher_ecdsa_verifies	Number of ECDSA verifications. • Type: UInt • SNMP name: "sslCipherECDSAVerifies"
ssl_cipher_encrypts	Bytes encrypted with a symmetric cipher. • Type: UInt • SNMP name: "sslCipherEncrypts"
ssl_cipher_rc4_decrypts	Bytes decrypted with RC4. • Type: UInt • SNMP name: "sslCipherRC4Decrypts"
ssl_cipher_rc4_encrypts	Bytes encrypted with RC4. • Type: UInt • SNMP name: "sslCipherRC4Encrypts"
ssl_cipher_rsa_decrypts	Number of RSA decrypts. • Type: UInt • SNMP name: "sslCipherRSADecrypts"
ssl_cipher_rsa_decrypts_external	Number of external RSA decrypts. • Type: UInt • SNMP name: "sslCipherRSADecryptsExternal"

Counter	Description
ssl_cipher_rsa_encrypts	Number of RSA encrypts. • Type: UInt • SNMP name: "sslCipherRSAEncrypts"
ssl_cipher_rsa_encrypts_external	Number of external RSA encrypts. • Type: UInt • SNMP name: "sslCipherRSAEncryptsExternal"
ssl_client_cert_expired	Number of times a client certificate has expired. • Type: UInt • SNMP name: "sslClientCertExpired"
ssl_client_cert_invalid	Number of times a client certificate was invalid. • Type: UInt • SNMP name: "sslClientCertInvalid"
ssl_client_cert_not_sent	Number of times a client certificate was required but not supplied. • Type: UInt • SNMP name: "sslClientCertNotSent"
ssl_client_cert_revoked	Number of times a client certificate was revoked. • Type: UInt • SNMP name: "sslClientCertRevoked"
ssl_connections	Number of SSL connections negotiated. • Type: UInt • SNMP name: "sslConnections"
ssl_handshake_sslv2	Formerly provided the number of SSLv2 handshakes, now deprecated. • Type: UInt • SNMP name: "sslHandshakeSSLv2"
ssl_handshake_sslv3	Number of SSLv3 handshakes. • Type: UInt • SNMP name: "sslHandshakeSSLv3"
ssl_handshake_t_l_sv1	Number of TLSv1.0 handshakes. • Type: UInt • SNMP name: "sslHandshakeTLSv1"
ssl_handshake_t_l_sv11	Number of TLSv1.1 handshakes. • Type: UInt • SNMP name: "sslHandshakeTLSv11"
ssl_handshake_t_l_sv12	Number of TLSv1.2 handshakes. • Type: UInt • SNMP name: "sslHandshakeTLSv12"

Counter	Description
ssl_session_id_disk_cache_hit	Number of times the SSL session id was found in the disk cache and reused (deprecated, will always return 0). • Type: UInt • SNMP name: "sslSessionIDDiskCacheHit"
ssl_session_id_disk_cache_miss	Number of times the SSL session id was not found in the disk cache (deprecated, will always return 0). • Type: UInt • SNMP name: "sslSessionIDDiskCacheMiss"
ssl_session_id_mem_cache_hit	Number of times the SSL session id was found in the cache and reused. • Type: UInt • SNMP name: "sslSessionIDMemCacheHit"
ssl_session_id_mem_cache_miss	Number of times the SSL session id was not found in the cache. • Type: UInt • SNMP name: "sslSessionIDMemCacheMiss"
sys_cpu_busy_percent	Percentage of time that the CPUs are busy. • Type: UInt • SNMP name: "sysCPUBusyPercent"
sys_cpu_idle_percent	Percentage of time that the CPUs are idle. • Type: UInt • SNMP name: "sysCPUIdlePercent"
sys_cpu_system_busy_percent	Percentage of time that the CPUs are busy running system code. • Type: UInt • SNMP name: "sysCPUSystemBusyPercent"
sys_cpu_user_busy_percent	Percentage of time that the CPUs are busy running user-space code. • Type: UInt • SNMP name: "sysCPUUserBusyPercent"
sys_fds_free	Number of free file descriptors. • Type: UInt • SNMP name: "sysFDsFree"
sys_mem_buffered	Buffer memory (MBytes). • Type: UInt • SNMP name: "sysMemBuffered"
sys_mem_free	Free memory (MBytes). • Type: UInt • SNMP name: "sysMemFree"
sys_mem_in_use	Memory used (MBytes). • Type: UInt • SNMP name: "sysMemInUse"

Counter	Description
sys_mem_swap_total	Total swap space (MBytes). • Type: UInt • SNMP name: "sysMemSwapTotal"
sys_mem_swapped	Amount of swap space in use (MBytes). • Type: UInt • SNMP name: "sysMemSwapped"
sys_mem_total	Total memory (MBytes). • Type: UInt • SNMP name: "sysMemTotal"
time_last_config_update	The time (in hundredths of a second) since the configuration of traffic manager was updated (this value will wrap if no configuration changes are made for 497 days). • Type: UInt • SNMP name: "timeLastConfigUpdate"
total_backend_server_errors	Total errors returned from the backend servers. • Type: UInt • SNMP name: "totalBackendServerErrors"
total_bad_dns_packets	Total number of malformed DNS response packets encountered from the backend servers. • Type: UInt • SNMP name: "totalBadDNSPackets"
total_bytes_in	Bytes received by the traffic manager from clients. • Type: UInt64 • SNMP name: "totalBytesIn"
total_bytes_in_hi	Bytes received by the traffic manager from clients (high 32bits). • Type: UInt • SNMP name: "totalBytesInHi"
total_bytes_in_lo	Bytes received by the traffic manager from clients (low 32bits). • Type: UInt • SNMP name: "totalBytesInLo"
total_bytes_out	Bytes sent by the traffic manager to clients. • Type: UInt64 • SNMP name: "totalBytesOut"
total_bytes_out_hi	Bytes sent by the traffic manager to clients (high 32bits). • Type: UInt • SNMP name: "totalBytesOutHi"
total_bytes_out_lo	Bytes sent by the traffic manager to clients (low 32bits). • Type: UInt • SNMP name: "totalBytesOutLo"

Counter	Description
total_conn	Total number of TCP connections received. • Type: UInt • SNMP name: "totalConn"
total_current_conn	Number of TCP connections currently established. • Type: UInt • SNMP name: "totalCurrentConn"
total_dns_responses	Total number of DNS response packets handled. • Type: UInt • SNMP name: "totalDNSResponses"
total_requests	Total number of TCP requests received. • Type: UInt • SNMP name: "totalRequests"
total_transactions	Total number of TCP requests being processed, after applying TPS limits. • Type: UInt • SNMP name: "totalTransactions"
up_time	The time (in hundredths of a second) that vTM software has been operational for (this value will wrap if it has been running for more than 497 days). • Type: UInt • SNMP name: "upTime"

Ip gateway

URI Path: statistics/traffic_ips/ip_gateway

Ip gateway statistics values.

Counter	Description
arp_message	Number of ARP messages sent for raised Traffic IP Addresses. • Type: UInt • SNMP name: "trafficIPARPMessage"
gateway_ping_requests	Number of ping requests sent to the gateway machine. • Type: UInt • SNMP name: "trafficIPGatewayPingRequests"
gateway_ping_responses	Number of ping responses received from the gateway machine. • Type: UInt • SNMP name: "trafficIPGatewayPingResponses"
node_ping_requests	Number of ping requests sent to the backend nodes. • Type: UInt • SNMP name: "trafficIPNodePingRequests"

Counter	Description
node_ping_responses	Number of ping responses received from the backend nodes. • Type: UInt • SNMP name: "trafficIPNodePingResponses"
number	The number of traffic IPv4 addresses on this system. • Type: UInt • SNMP name: "trafficIPNumber"
number_inet46	The number of traffic IP addresses on this system (includes IPv4 and IPv6 addresses). • Type: UInt • SNMP name: "trafficIPNumberInet46"
number_raised	The number of traffic IPv4 addresses currently raised on this system. • Type: UInt • SNMP name: "trafficIPNumberRaised"
number_raised_inet46	The number of traffic IP addresses currently raised on this system (includes IPv4 and IPv6 addresses). • Type: UInt • SNMP name: "trafficIPNumberRaisedInet46"
ping_response_errors	Number of ping response errors. • Type: UInt • SNMP name: "trafficIPPingResponseErrors"

Ip session cache

URI Path: statistics/cache/ip_session_cache

Ip session cache statistics values.

Counter	Description
entries	The total number of IP sessions stored in the cache. • Type: UInt • SNMP name: "ipSessionCacheEntries"
entries_max	The maximum number of IP sessions in the cache. • Type: UInt • SNMP name: "ipSessionCacheEntriesMax"
hit_rate	The percentage of IP session lookups that succeeded. • Type: UInt • SNMP name: "ipSessionCacheHitRate"
hits	Number of times a IP session entry has been successfully found in the cache. • Type: UInt • SNMP name: "ipSessionCacheHits"

Counter	Description
lookups	Number of times a IP session entry has been looked up in the cache. • Type: UInt • SNMP name: "ipSessionCacheLookups"
misses	Number of times a IP session entry has not been available in the cache. • Type: UInt • SNMP name: "ipSessionCacheMisses"
oldest	The age of the oldest IP session in the cache (in seconds). • Type: UInt • SNMP name: "ipSessionCacheOldest"

J2ee session cache

URI Path: statistics/cache/j2ee_session_cache

J2ee session cache statistics values.

Counter	Description
entries	The total number of J2EE sessions stored in the cache. • Type: UInt • SNMP name: "j2eeSessionCacheEntries"
entries_max	The maximum number of J2EE sessions in the cache. • Type: UInt • SNMP name: "j2eeSessionCacheEntriesMax"
hit_rate	The percentage of J2EE session lookups that succeeded. • Type: UInt • SNMP name: "j2eeSessionCacheHitRate"
hits	Number of times a J2EE session entry has been successfully found in the cache. • Type: UInt • SNMP name: "j2eeSessionCacheHits"
lookups	Number of times a J2EE session entry has been looked up in the cache. • Type: UInt • SNMP name: "j2eeSessionCacheLookups"
misses	Number of times a J2EE session entry has not been available in the cache. • Type: UInt • SNMP name: "j2eeSessionCacheMisses"
oldest	The age of the oldest J2EE session in the cache (in seconds). • Type: UInt • SNMP name: "j2eeSessionCacheOldest"

Listen ips

URI Path: statistics/listen_ips/*

Listen ips statistics values.

Counter	Description
bytes_in	Bytes sent to this listening IP. • Type: UInt64 • SNMP name: "listenIPBytesIn"
bytes_in_hi	Bytes sent to this listening IP (high 32bits). • Type: UInt • SNMP name: "listenIPBytesInHi"
bytes_in_lo	Bytes sent to this listening IP (low 32bits). • Type: UInt • SNMP name: "listenIPBytesInLo"
bytes_out	Bytes sent from this listening IP. • Type: UInt64 • SNMP name: "listenIPBytesOut"
bytes_out_hi	Bytes sent from this listening IP (high 32bits). • Type: UInt • SNMP name: "listenIPBytesOutHi"
bytes_out_lo	Bytes sent from this listening IP (low 32bits). • Type: UInt • SNMP name: "listenIPBytesOutLo"
current_conn	TCP connections currently established to this listening IP. • Type: UInt • SNMP name: "listenIPCurrentConn"
max_conn	Maximum number of simultaneous TCP connections this listening IP has processed at any one time. • Type: UInt • SNMP name: "listenIPMaxConn"
total_conn	Formerly provided the number of requests sent to this listening IP, now deprecated. • Type: UInt • SNMP name: "listenIPTotalConn"
total_requests	Requests sent to this listening IP. • Type: UInt64 • SNMP name: "listenIPTotalRequests"

Counter	Description
total_requests_hi	Requests sent to this listening IP. • Type: UInt • SNMP name: "listenIPTotalRequestsHi"
total_requests_lo	Requests sent to this listening IP. • Type: UInt • SNMP name: "listenIPTotalRequestsLo"

Locations

URI Path: statistics/locations/*

Locations statistics values.

Counter	Description
load	The mean load metric for this location. • Type: UInt • SNMP name: "locationLoad"
responses	Number of A records that have been altered to point to this location. • Type: UInt • SNMP name: "locationResponses"

Network interface

URI Path: statistics/network_interface/*

Network interface statistics values.

Counter	Description
collisions	The number of collisions reported by this interface. • Type: UInt • SNMP name: "interfaceCollisions"
rx_bytes	Bytes received by this interface. • Type: UInt64 • SNMP name: "interfaceRxBytes"
rx_bytes_hi	Bytes received by this interface (high 32bits). • Type: UInt • SNMP name: "interfaceRxBytesHi"
rx_bytes_lo	Bytes received by this interface (low 32bits). • Type: UInt • SNMP name: "interfaceRxBytesLo"

Counter	Description
rx_errors	The number of receive errors reported by this interface. • Type: UInt • SNMP name: "interfaceRxErrors"
rx_packets	The number of packets received by this interface. • Type: UInt • SNMP name: "interfaceRxPackets"
tx_bytes	Bytes transmitted by this interface. • Type: UInt64 • SNMP name: "interfaceTxBytes"
tx_bytes_hi	Bytes transmitted by this interface (high 32bits). • Type: UInt • SNMP name: "interfaceTxBytesHi"
tx_bytes_lo	Bytes transmitted by this interface (low 32bits). • Type: UInt • SNMP name: "interfaceTxBytesLo"
tx_errors	The number of transmit errors reported by this interface. • Type: UInt • SNMP name: "interfaceTxErrors"
tx_packets	The number of packets transmitted by this interface. • Type: UInt • SNMP name: "interfaceTxPackets"

Node

URI Path: statistics/nodes/node/*

Node statistics values.

Counter	Description
bytes_from_node_hi	Bytes received from this node (high 32bits). • Type: UInt • SNMP name: "nodeBytesFromNodeHi"
bytes_from_node_lo	Bytes received from this node (low 32bits). • Type: UInt • SNMP name: "nodeBytesFromNodeLo"
bytes_to_node_hi	Bytes sent to this node (high 32bits). • Type: UInt • SNMP name: "nodeBytesToNodeHi"

Counter	Description
bytes_to_node_lo	Bytes sent to this node (low 32bits). • Type: UInt • SNMP name: "nodeBytesToNodeLo"
current_conn	Requests currently established to this node. (does not include idle keepalives). • Type: UInt • SNMP name: "nodeCurrentConn"
current_requests	Connections currently established to this node. • Type: UInt • SNMP name: "nodeCurrentRequests"
errors	Number of timeouts, connection problems and other errors for this node. • Type: UInt • SNMP name: "nodeErrors"
failures	Failures of this node. • Type: UInt • SNMP name: "nodeFailures"
new_conn	Requests that created a new connection to this node. • Type: UInt • SNMP name: "nodeNewConn"
pooled_conn	Requests that reused an existing pooled/keepalive connection rather than creating a new TCP connection. • Type: UInt • SNMP name: "nodePooledConn"
port	The port this node listens on. • Type: UInt • SNMP name: "nodePort"
response_max	Maximum response time (ms) in the last second for this node. • Type: UInt • SNMP name: "nodeResponseMax"
response_mean	Mean response time (ms) in the last second for this node. • Type: UInt • SNMP name: "nodeResponseMean"
response_min	Minimum response time (ms) in the last second for this node. • Type: UInt • SNMP name: "nodeResponseMin"

Counter	Description
state	The state of this node. • Type: Enum(String) • SNMP name: "nodeState" • Permitted values: "alive": alive(1) "dead": dead(2) "unknown": unknown(3)
total_conn	Requests sent to this node. • Type: UInt • SNMP name: "nodeTotalConn"

Node inet46

URI Path: statistics/nodes/node_inet46/*

Node inet46 statistics values.

Counter	Description
bytes_from_node	Bytes received from this node. • Type: UInt64 • SNMP name: "nodeInet46BytesFromNode"
bytes_from_node_hi	Bytes received from this node (high 32bits). • Type: UInt • SNMP name: "nodeInet46BytesFromNodeHi"
bytes_from_node_lo	Bytes received from this node (low 32bits). • Type: UInt • SNMP name: "nodeInet46BytesFromNodeLo"
bytes_to_node	Bytes sent to this node. • Type: UInt64 • SNMP name: "nodeInet46BytesToNode"
bytes_to_node_hi	Bytes sent to this node (high 32bits). • Type: UInt • SNMP name: "nodeInet46BytesToNodeHi"
bytes_to_node_lo	Bytes sent to this node (low 32bits). • Type: UInt • SNMP name: "nodeInet46BytesToNodeLo"
current_conn	Current connections established to this node, includes idle connections. • Type: UInt • SNMP name: "nodeInet46CurrentConn"

Counter	Description
current_requests	Active connections established to this node, does not include idle connections. • Type: UInt • SNMP name: "nodeInet46CurrentRequests"
errors	Number of timeouts, connection problems and other errors for this node. • Type: UInt • SNMP name: "nodeInet46Errors"
failures	Failures of this node. • Type: UInt • SNMP name: "nodeInet46Failures"
idle_conns	Number of idle HTTP connections to this node. • Type: UInt • SNMP name: "nodeInet46IdleConns"
new_conn	Requests that created a new connection to this node. • Type: UInt • SNMP name: "nodeInet46NewConn"
pooled_conn	Requests that reused an existing pooled/keepalive connection rather than creating a new TCP connection. • Type: UInt • SNMP name: "nodeInet46PooledConn"
port	The port this node listens on. • Type: UInt • SNMP name: "nodeInet46Port"
response_max	Maximum response time (ms) in the last second for this node. • Type: UInt • SNMP name: "nodeInet46ResponseMax"
response_mean	Mean response time (ms) in the last second for this node. • Type: UInt • SNMP name: "nodeInet46ResponseMean"
response_min	Minimum response time (ms) in the last second for this node. • Type: UInt • SNMP name: "nodeInet46ResponseMin"

Counter	Description
state	The state of this node. • Type: Enum(String) • SNMP name: "nodeInet46State" • Permitted values: "alive": alive(1) "dead": dead(2) "unknown": unknown(3)
total_conn	Requests sent to this node. • Type: UInt • SNMP name: "nodeInet46TotalConn"

Per location service

URI Path: statistics/per_location_service/*

Per location service statistics values.

Counter	Description
draining	The draining state of this location for this GLB Service. • Type: Enum(String) • SNMP name: "perLocationServiceDraining" • Permitted values: "draining": draining(1) "active": active(2)
frontend_state	The frontend state of this location for this GLB Service. • Type: Enum(String) • SNMP name: "perLocationServiceFrontendState" • Permitted values: "alive": alive(1) "dead": dead(2)
load	The load metric for this location for this GLB Service. • Type: UInt • SNMP name: "perLocationServiceLoad"
monitor_state	The monitor state of this location for this GLB Service. • Type: Enum(String) • SNMP name: "perLocationServiceMonitorState" • Permitted values: "alive": alive(1) "dead": dead(2)

Counter	Description
responses	Number of A records that have been altered to point to this location for this GLB Service. • Type: UInt • SNMP name: "perLocationServiceResponses"
state	The state of this location for this GLB Service. • Type: Enum(String) • SNMP name: "perLocationServiceState" • Permitted values: "alive": alive(1) "dead": dead(2)

Per node service level

URI Path: statistics/per_node_slm/per_node_service_level/*

Per node service level statistics values.

Counter	Description
node_port	The port number of this node. • Type: UInt • SNMP name: "perNodeServiceLevelNodePort"
response_max	Maximum response time (ms) in the last second for this SLM class to this node. • Type: UInt • SNMP name: "perNodeServiceLevelResponseMax"
response_mean	Mean response time (ms) in the last second for this SLM class to this node. • Type: UInt • SNMP name: "perNodeServiceLevelResponseMean"
response_min	Minimum response time (ms) in the last second for this SLM class to this node. • Type: UInt • SNMP name: "perNodeServiceLevelResponseMin"
total_conn	Requests handled by this SLM class to this node. • Type: UInt • SNMP name: "perNodeServiceLevelTotalConn"
total_non_conf	Non-conforming requests handled by this SLM class to this node. • Type: UInt • SNMP name: "perNodeServiceLevelTotalNonConf"

Per node service level inet46

URI Path: statistics/per_node_slm/per_node_service_level_inet46/*

Per node service level inet46 statistics values.

Counter	Description
node_port	The port number of this node. • Type: UInt • SNMP name: "perNodeServiceLevelInet46NodePort"
response_max	Maximum response time (ms) in the last second for this SLM class to this node. • Type: UInt • SNMP name: "perNodeServiceLevelInet46ResponseMax"
response_mean	Mean response time (ms) in the last second for this SLM class to this node. • Type: UInt • SNMP name: "perNodeServiceLevelInet46ResponseMean"
response_min	Minimum response time (ms) in the last second for this SLM class to this node. • Type: UInt • SNMP name: "perNodeServiceLevelInet46ResponseMin"
total_conn	Requests handled by this SLM class to this node. • Type: UInt • SNMP name: "perNodeServiceLevelInet46TotalConn"
total_non_conf	Non-conforming requests handled by this SLM class to this node. • Type: UInt • SNMP name: "perNodeServiceLevelInet46TotalNonConf"

Per pool node

URI Path: statistics/nodes/per_pool_node/*

Per pool node statistics values.

Counter	Description
bytes_from_node	Bytes received from this node. • Type: UInt64 • SNMP name: "perPoolNodeBytesFromNode"
bytes_from_node_hi	Bytes received from this node (high 32bits). • Type: UInt • SNMP name: "perPoolNodeBytesFromNodeHi"
bytes_from_node_lo	Bytes received from this node (low 32bits). • Type: UInt • SNMP name: "perPoolNodeBytesFromNodeLo"

Counter	Description
bytes_to_node	Bytes sent to this node. • Type: UInt64 • SNMP name: "perPoolNodeBytesToNode"
bytes_to_node_hi	Bytes sent to this node (high 32bits). • Type: UInt • SNMP name: "perPoolNodeBytesToNodeHi"
bytes_to_node_lo	Bytes sent to this node (low 32bits). • Type: UInt • SNMP name: "perPoolNodeBytesToNodeLo"
current_conn	Current connections established to a node, includes idle connections. • Type: UInt • SNMP name: "perPoolNodeCurrentConn"
current_requests	Active connections established to this node, does not include idle connections. • Type: UInt • SNMP name: "perPoolNodeCurrentRequests"
errors	Number of timeouts, connection problems and other errors for this node. • Type: UInt • SNMP name: "perPoolNodeErrors"
failures	Failures of this node. • Type: UInt • SNMP name: "perPoolNodeFailures"
idle_conns	Number of idle HTTP connections to this node. • Type: UInt • SNMP name: "perPoolNodeIdleConns"
l4_stateless_buckets	Number of hash buckets occupied for this node for L4 stateless processing. • Type: UInt • SNMP name: "perPoolNodeL4StatelessBuckets"
new_conn	Requests that created a new connection to this node. • Type: UInt • SNMP name: "perPoolNodeNewConn"
node_port	The port that this node listens on. • Type: UInt • SNMP name: "perPoolNodeNodePort"
pkts_from_node	Packets received from this node. • Type: UInt64 • SNMP name: "perPoolNodePktsFromNode"
pkts_from_node_hi	Packets received from this node (high 32bits). • Type: UInt • SNMP name: "perPoolNodePktsFromNodeHi"

Counter	Description
pkts_from_node_lo	Packets received from this node (low 32bits). • Type: UInt • SNMP name: "perPoolNodePktsFromNodeLo"
pkts_to_node	Packets sent to this node. • Type: UInt64 • SNMP name: "perPoolNodePktsToNode"
pkts_to_node_hi	Packets sent to this node (high 32bits). • Type: UInt • SNMP name: "perPoolNodePktsToNodeHi"
pkts_to_node_lo	Packets sent to this node (low 32bits). • Type: UInt • SNMP name: "perPoolNodePktsToNodeLo"
pooled_conn	Requests that reused an existing pooled/keepalive connection rather than creating a new TCP connection. • Type: UInt • SNMP name: "perPoolNodePooledConn"
response_max	Maximum response time (ms) in the last second for this node. • Type: UInt • SNMP name: "perPoolNodeResponseMax"
response_mean	Mean response time (ms) in the last second for this node. • Type: UInt • SNMP name: "perPoolNodeResponseMean"
response_min	Minimum response time (ms) in the last second for this node. • Type: UInt • SNMP name: "perPoolNodeResponseMin"
state	The state of this node. • Type: Enum(String) • SNMP name: "perPoolNodeState" • Permitted values: "alive": alive(1) "dead": dead(2) "unknown": unknown(3) "draining": draining(4) "drainingtodelete": drainingtodelete(5)
total_conn	Requests sent to this node. • Type: UInt • SNMP name: "perPoolNodeTotalConn"

Pools

URI Path: statistics/pools/*

Pools statistics values.

Counter	Description
algorithm	The load-balancing algorithm the pool uses. • Type: Enum(String) • SNMP name: "poolAlgorithm" • Permitted values: "roundrobin": roundrobin(1) "weightedRoundRobin": weightedRoundRobin(2) "perceptive": perceptive(3) "leastConnections": leastConnections(4) "fastestResponseTime": fastestResponseTime(5) "random": random(6) "weightedLeastConnections": weightedLeastConnections(7)
bw_limit_bytes_drop	Bytes dropped by this pool due to BW Limits. • Type: UInt64 • SNMP name: "poolBwLimitBytesDrop"
bw_limit_bytes_drop_hi	Bytes dropped by this pool due to BW Limits (high 32bits). • Type: UInt • SNMP name: "poolBwLimitBytesDropHi"
bw_limit_bytes_drop_lo	Bytes dropped by this pool due to BW Limits (low 32bits). • Type: UInt • SNMP name: "poolBwLimitBytesDropLo"
bw_limit_pkts_drop	Number of packets dropped by this pool due to BW Limits. • Type: UInt64 • SNMP name: "poolBwLimitPktsDrop"
bw_limit_pkts_drop_hi	Number of packets dropped by this pool due to BW Limits (high 32bits). • Type: UInt • SNMP name: "poolBwLimitPktsDropHi"
bw_limit_pkts_drop_lo	Number of packets dropped by this pool due to BW Limits (low 32bits). • Type: UInt • SNMP name: "poolBwLimitPktsDropLo"
bytes_in	Bytes received by this pool from nodes. • Type: UInt64 • SNMP name: "poolBytesIn"

Counter	Description
bytes_in_hi	Bytes received by this pool from nodes (high 32bits). • Type: UInt • SNMP name: "poolBytesInHi"
bytes_in_lo	Bytes received by this pool from nodes (low 32bits). • Type: UInt • SNMP name: "poolBytesInLo"
bytes_out	Bytes sent by this pool to nodes. • Type: UInt64 • SNMP name: "poolBytesOut"
bytes_out_hi	Bytes sent by this pool to nodes (high 32bits). • Type: UInt • SNMP name: "poolBytesOutHi"
bytes_out_lo	Bytes sent by this pool to nodes (low 32bits). • Type: UInt • SNMP name: "poolBytesOutLo"
conns_queued	Total connections currently queued to this pool. • Type: UInt • SNMP name: "poolConnsQueued"
disabled	The number of nodes in this pool that are disabled. • Type: UInt • SNMP name: "poolDisabled"
draining	The number of nodes in this pool which are draining. • Type: UInt • SNMP name: "poolDraining"
max_queue_time	Maximum time a connection was queued for, over the last second. • Type: UInt • SNMP name: "poolMaxQueueTime"
mean_queue_time	Mean time a connection was queued for, over the last second. • Type: UInt • SNMP name: "poolMeanQueueTime"
min_queue_time	Minimum time a connection was queued for, over the last second. • Type: UInt • SNMP name: "poolMinQueueTime"
nodes	The number of nodes registered with this pool. • Type: UInt • SNMP name: "poolNodes"

Counter	Description
persistence	The session persistence method this pool uses • Type: Enum(String) • SNMP name: "poolPersistence" • Permitted values: "none": none(1) "ip": ip(2) "rule": rule(3) "transparent": transparent(4) "applicationCookie": applicationCookie(5) "xZeusBackend": xZeusBackend(6) "ssl": ssl(7)
queue_timeouts	Total connections that timed-out while queued. • Type: UInt • SNMP name: "poolQueueTimeouts"
session_migrated	Sessions migrated to a new node because the desired node was unavailable. • Type: UInt • SNMP name: "poolSessionMigrated"
state	The state of this pool. • Type: Enum(String) • SNMP name: "poolState" • Permitted values: "active": active(1) "disabled": disabled(2) "draining": draining(3) "unused": unused(4) "unknown": unknown(5)
total_conn	Requests sent to this pool. • Type: UInt • SNMP name: "poolTotalConn"

Rule authenticators

URI Path: statistics/rule_authenticators/*

Rule authenticators statistics values.

Counter	Description
errors	Number of connection errors that have occurred when trying to connect to an authentication server. • Type: UInt • SNMP name: "authenticatorErrors"
fails	Number of times this Authenticator has failed to authenticate. • Type: UInt • SNMP name: "authenticatorFails"
passes	Number of times this Authenticator has successfully authenticated. • Type: UInt • SNMP name: "authenticatorPasses"
requests	Number of times this Authenticator has been asked to authenticate. • Type: UInt • SNMP name: "authenticatorRequests"

Rules

URI Path: statistics/rules/*

Rules statistics values.

Counter	Description
aborts	Number of times this TrafficScript rule has aborted. • Type: UInt • SNMP name: "ruleAborts"
discards	Number of times this TrafficScript rule has discarded the connection. • Type: UInt • SNMP name: "ruleDiscards"
execution_time_warnings	Number of times this TrafficScript rule has exceeded the execution time warning threshold. • Type: UInt • SNMP name: "ruleExecutionTimeWarnings"
executions	Number of times this TrafficScript rule has been executed. • Type: UInt • SNMP name: "ruleExecutions"
pool_select	Number of times this TrafficScript rule has selected a pool to use. • Type: UInt • SNMP name: "rulePoolSelect"

Counter	Description
responds	Number of times this TrafficScript rule has responded directly to the client. • Type: UInt • SNMP name: "ruleResponds"
retries	Number of times this TrafficScript rule has forced the request to be retried. • Type: UInt • SNMP name: "ruleRetries"

Service level monitors

URI Path: statistics/service_level_monitors/*

Service level monitors statistics values.

Counter	Description
conforming	Percentage of requests associated with this SLM class that are conforming • Type: UInt • SNMP name: "serviceLevelConforming"
current_conns	The number of connections currently associated with this SLM class. • Type: UInt • SNMP name: "serviceLevelCurrentConns"
is_o_k	Indicates if this SLM class is currently conforming. • Type: Enum(String) • SNMP name: "serviceLevelIsOK" • Permitted values: "notok": notok(1) "ok": ok(2)
response_max	Maximum response time (ms) in the last second for this SLM class. • Type: UInt • SNMP name: "serviceLevelResponseMax"
response_mean	Mean response time (ms) in the last second for this SLM class. • Type: UInt • SNMP name: "serviceLevelResponseMean"
response_min	Minimum response time (ms) in the last second for this SLM class. • Type: UInt • SNMP name: "serviceLevelResponseMin"

Counter	Description
total_conn	Requests handled by this SLM class. • Type: UInt • SNMP name: "serviceLevelTotalConn"
total_non_conf	Non-conforming requests handled by this SLM class. • Type: UInt • SNMP name: "serviceLevelTotalNonConf"

Service protection

URI Path: statistics/service_protection/*

Service protection statistics values.

Counter	Description
last_refusal_time	The time (in hundredths of a second) since this service protection class last refused a connection (this value will wrap if no connections are refused in more than 497 days). • Type: UInt • SNMP name: "serviceProtLastRefusalTime"
refusal_binary	Connections refused by this service protection class because the request contained disallowed binary content. • Type: UInt • SNMP name: "serviceProtRefusalBinary"
refusal_conc10_ip	Connections refused by this service protection class because the top 10 source IP addresses issued too many concurrent connections. • Type: UInt • SNMP name: "serviceProtRefusalConc10IP"
refusal_conc1_ip	Connections refused by this service protection class because the source IP address issued too many concurrent connections. • Type: UInt • SNMP name: "serviceProtRefusalConc1IP"
refusal_conn_rate	Connections refused by this service protection class because the source IP address issued too many connections within 60 seconds. • Type: UInt • SNMP name: "serviceProtRefusalConnRate"
refusal_ip	Connections refused by this service protection class because the source IP address was banned. • Type: UInt • SNMP name: "serviceProtRefusalIP"

Counter	Description
refusal_rfc2396	Connections refused by this service protection class because the HTTP request was not RFC 2396 compliant. • Type: UInt • SNMP name: "serviceProtRefusalRFC2396"
refusal_size	Connections refused by this service protection class because the request was larger than the defined limits allowed. • Type: UInt • SNMP name: "serviceProtRefusalSize"
total_refusal	Connections refused by this service protection class. • Type: UInt • SNMP name: "serviceProtTotalRefusal"

Ssl cache

URI Path: statistics/cache/ssl_cache

Ssl cache statistics values.

Counter	Description
entries	The total number of SSL sessions stored in the server cache. • Type: UInt • SNMP name: "sslCacheEntries"
entries_max	The maximum number of SSL entries in the server cache. • Type: UInt • SNMP name: "sslCacheEntriesMax"
hit_rate	The percentage of SSL server cache lookups that succeeded. • Type: UInt • SNMP name: "sslCacheHitRate"
hits	Number of times a SSL entry has been successfully found in the server cache. • Type: UInt • SNMP name: "sslCacheHits"
lookups	Number of times a SSL entry has been looked up in the server cache. • Type: UInt • SNMP name: "sslCacheLookups"
misses	Number of times a SSL entry has not been available in the server cache. • Type: UInt • SNMP name: "sslCacheMisses"
oldest	The age of the oldest SSL session in the server cache (in seconds). • Type: UInt • SNMP name: "sslCacheOldest"

Ssl ocsp stapling

URI Path: statistics/ssl_ocsp_stapling

Ssl ocsp stapling statistics values.

Counter	Description
cache_count	The number of entries in the OCSP stapling cache. • Type: UInt • SNMP name: "sslOcspStaplingCacheCount"
count	The number of outgoing OCSP requests for OCSP stapling. • Type: UInt • SNMP name: "sslOcspStaplingCount"
failure_count	The number of failed outgoing OCSP requests for OCSP stapling. • Type: UInt • SNMP name: "sslOcspStaplingFailureCount"
good_count	The number of 'good' OCSP responses for OCSP stapling. • Type: UInt • SNMP name: "sslOcspStaplingGoodCount"
revoked_count	The number of 'revoked' OCSP responses for OCSP stapling. • Type: UInt • SNMP name: "sslOcspStaplingRevokedCount"
success_count	The number of successful outgoing OCSP requests for OCSP stapling. • Type: UInt • SNMP name: "sslOcspStaplingSuccessCount"
unknown_count	The number of 'unknown' OCSP requests for OCSP stapling. • Type: UInt • SNMP name: "sslOcspStaplingUnknownCount"

Ssl session cache

URI Path: statistics/cache/ssl_session_cache

Ssl session cache statistics values.

Counter	Description
entries	The total number of SSL session persistence entries stored in the cache. • Type: UInt • SNMP name: "sslSessionCacheEntries"
entries_max	The maximum number of SSL session persistence entries in the cache. • Type: UInt • SNMP name: "sslSessionCacheEntriesMax"
hit_rate	The percentage of SSL session persistence lookups that succeeded. • Type: UInt • SNMP name: "sslSessionCacheHitRate"
hits	Number of times a SSL session persistence entry has been successfully found in the cache. • Type: UInt • SNMP name: "sslSessionCacheHits"
lookups	Number of times a SSL session persistence entry has been looked up in the cache. • Type: UInt • SNMP name: "sslSessionCacheLookups"
misses	Number of times a SSL session persistence entry has not been available in the cache. • Type: UInt • SNMP name: "sslSessionCacheMisses"
oldest	The age of the oldest SSL session in the cache (in seconds). • Type: UInt • SNMP name: "sslSessionCacheOldest"

Traffic ip

URI Path: statistics/traffic_ips/traffic_ip/*

Traffic ip statistics values.

Counter	Description
state	Whether this traffic IP address is currently being hosted by this traffic manager. • Type: Enum(String) • SNMP name: "trafficIPState" • Permitted values: "raised": raised(1) "lowered": lowered(2)
time	The time (in hundredths of a second) since trafficIPState last changed (this value will wrap if the state hasn't changed for 497 days). • Type: UInt • SNMP name: "trafficIPTime"

Traffic ip inet46

URI Path: statistics/traffic_ips/traffic_ip_inet46/*

Traffic ip inet46 statistics values.

Counter	Description
state	Whether this traffic IP address is currently being hosted by this traffic manager. • Type: Enum(String) • SNMP name: "trafficIPInet46State" • Permitted values: "raised": raised(1) "lowered": lowered(2)
time	The time (in hundredths of a second) since trafficIPState last changed (this value will wrap if the state hasn't changed for 497 days). • Type: UInt • SNMP name: "trafficIPInet46Time"

Uni session cache

URI Path: statistics/cache/uni_session_cache

Uni session cache statistics values.

Counter	Description
entries	The total number of universal sessions stored in the cache. • Type: UInt • SNMP name: "uniSessionCacheEntries"
entries_max	The maximum number of universal sessions in the cache. • Type: UInt • SNMP name: "uniSessionCacheEntriesMax"
hit_rate	The percentage of universal session lookups that succeeded. • Type: UInt • SNMP name: "uniSessionCacheHitRate"
hits	Number of times a universal session entry has been successfully found in the cache. • Type: UInt • SNMP name: "uniSessionCacheHits"
lookups	Number of times a universal session entry has been looked up in the cache. • Type: UInt • SNMP name: "uniSessionCacheLookups"
misses	Number of times a universal session entry has not been available in the cache. • Type: UInt • SNMP name: "uniSessionCacheMisses"
oldest	The age of the oldest universal session in the cache (in seconds). • Type: UInt • SNMP name: "uniSessionCacheOldest"

User counters 32

URI Path: statistics/extras/user_counters_32

User counters 32 statistics values.

Counter	Description
counter	The value of the user counter. • Type: UInt • SNMP name: "userCounterValue"

User counters 64

URI Path: statistics/extras/user_counters_64

User counters 64 statistics values.

Counter	Description
counter	The value of the 64-bit user counter. • Type: UInt64 • SNMP name: "userCounter64Value"

Virtual servers

URI Path: statistics/virtual_servers/*

Virtual servers statistics values.

Counter	Description
bw_limit_bytes_drop	Number of bytes dropped by this virtual server due to BW Limits. • Type: UInt64 • SNMP name: "virtualserverBwLimitBytesDrop"
bw_limit_bytes_drop_hi	Number of bytes dropped by this virtual server due to BW Limits (high 32bits). • Type: UInt • SNMP name: "virtualserverBwLimitBytesDropHi"
bw_limit_bytes_drop_lo	Number of bytes dropped by this virtual server due to BW Limits (low 32bits). • Type: UInt • SNMP name: "virtualserverBwLimitBytesDropLo"
bw_limit_pkts_drop	Number of packets dropped by this virtual server due to BW Limits. • Type: UInt64 • SNMP name: "virtualserverBwLimitPktsDrop"
bw_limit_pkts_drop_hi	Number of packets dropped by this virtual server due to BW Limits (high 32bits). • Type: UInt • SNMP name: "virtualserverBwLimitPktsDropHi"
bw_limit_pkts_drop_lo	Number of packets dropped by this virtual server due to BW Limits (low 32bits). • Type: UInt • SNMP name: "virtualserverBwLimitPktsDropLo"
bytes_in	Bytes received by this virtual server from clients. • Type: UInt64 • SNMP name: "virtualserverBytesIn"
bytes_in_hi	Bytes received by this virtual server from clients (high 32bits). • Type: UInt • SNMP name: "virtualserverBytesInHi"

Counter	Description
bytes_in_lo	Bytes received by this virtual server from clients (low 32bits). • Type: UInt • SNMP name: "virtualserverBytesInLo"
bytes_out	Bytes sent by this virtual server to clients. • Type: UInt64 • SNMP name: "virtualserverBytesOut"
bytes_out_hi	Bytes sent by this virtual server to clients (high 32bits). • Type: UInt • SNMP name: "virtualserverBytesOutHi"
bytes_out_lo	Bytes sent by this virtual server to clients (low 32bits). • Type: UInt • SNMP name: "virtualserverBytesOutLo"
cert_status_requests	Number of incoming TLS handshakes for this virtual server with certificate status requests. • Type: UInt • SNMP name: "virtualserverCertStatusRequests"
cert_status_responses	Number of incoming TLS handshakes for this virtual server to which certificate status responses were attached. • Type: UInt • SNMP name: "virtualserverCertStatusResponses"
connect_timed_out	Connections closed by this virtual server because the 'connect_timeout' interval was exceeded. • Type: UInt • SNMP name: "virtualserverConnectTimedOut"
connection_errors	Number of transaction or protocol errors in this virtual server. • Type: UInt • SNMP name: "virtualserverConnectionErrors"
connection_failures	Number of connection failures in this virtual server. • Type: UInt • SNMP name: "virtualserverConnectionFailures"
current_conn	TCP connections currently established to this virtual server. • Type: UInt • SNMP name: "virtualserverCurrentConn"
data_timed_out	Connections closed by this virtual server because the 'timeout' interval was exceeded. • Type: UInt • SNMP name: "virtualserverDataTimedOut"
direct_replies	Direct replies from this virtual server, without forwarding to a node. • Type: UInt • SNMP name: "virtualserverDirectReplies"

Counter	Description
discard	Connections discarded by this virtual server. • Type: UInt • SNMP name: "virtualserverDiscard"
gzip	Responses which have been compressed by content compression. • Type: UInt • SNMP name: "virtualserverGzip"
gzip_bytes_saved	Bytes of network traffic saved by content compression. • Type: UInt64 • SNMP name: "virtualserverGzipBytesSaved"
gzip_bytes_saved_hi	Bytes of network traffic saved by content compression (high 32bits). • Type: UInt • SNMP name: "virtualserverGzipBytesSavedHi"
gzip_bytes_saved_lo	Bytes of network traffic saved by content compression (low 32bits). • Type: UInt • SNMP name: "virtualserverGzipBytesSavedLo"
http_cache_hit_rate	Percentage hit rate of the web cache for this virtual server. • Type: UInt • SNMP name: "virtualserverHttpCacheHitRate"
http_cache_hits	HTTP responses sent directly from the web cache by this virtual server. • Type: UInt • SNMP name: "virtualserverHttpCacheHits"
http_cache_lookups	HTTP requests that are looked up in the web cache by this virtual server. • Type: UInt • SNMP name: "virtualserverHttpCacheLookups"
http_rewrite_cookie	HTTP Set-Cookie headers, supplied by a node, that have been rewritten. • Type: UInt • SNMP name: "virtualserverHttpRewriteCookie"
http_rewrite_location	HTTP Location headers, supplied by a node, that have been rewritten. • Type: UInt • SNMP name: "virtualserverHttpRewriteLocation"
keepalive_timed_out	Connections closed by this virtual server because the 'keepalive_timeout' interval was exceeded. • Type: UInt • SNMP name: "virtualserverKeepaliveTimedOut"
max_conn	Maximum number of simultaneous TCP connections this virtual server has processed at any one time. • Type: UInt • SNMP name: "virtualserverMaxConn"

Counter	Description
max_duration_timed_out	Connections closed by this virtual server because the 'max_transaction_duration' interval was exceeded. • Type: UInt • SNMP name: "virtualserverMaxDurationTimedOut"
pkts_in	Packets received by this virtual server from clients. • Type: UInt64 • SNMP name: "virtualserverPktsIn"
pkts_in_hi	Packets received by this virtual server from clients (high 32bits). • Type: UInt • SNMP name: "virtualserverPktsInHi"
pkts_in_lo	Packets received by this virtual server from clients (low 32bits). • Type: UInt • SNMP name: "virtualserverPktsInLo"
pkts_out	Packets sent by this virtual server to clients. • Type: UInt64 • SNMP name: "virtualserverPktsOut"
pkts_out_hi	Packets sent by this virtual server to clients (high 32bits). • Type: UInt • SNMP name: "virtualserverPktsOutHi"
pkts_out_lo	Packets sent by this virtual server to clients (low 32bits). • Type: UInt • SNMP name: "virtualserverPktsOutLo"
port	The port the virtual server listens on. • Type: UInt • SNMP name: "virtualserverPort"
processing_timed_out	Connections closed by this virtual server because the 'timeout' interval was exceeded while waiting for rules or external processing. • Type: UInt • SNMP name: "virtualserverProcessingTimedOut"

Counter	Description
protocol	The protocol the virtual server is operating. • Type: Enum(String) • SNMP name: "virtualserverProtocol" • Permitted values: "http": http(1) "https": https(2) "ftp": ftp(3) "imaps": imaps(4) "imapv2": imapv2(5) "imapv3": imapv3(6) "imapv4": imapv4(7) "pop3": pop3(8) "pop3s": pop3s(9) "smtp": smtp(10) "ldap": ldap(11) "ldaps": ldaps(12) "telnet": telnet(13) "sslforwarding": sslforwarding(14) "udpstreaming": udpstreaming(15) "udp": udp(16) "dns": dns(17) "genericserverfirst": genericserverfirst(18) "genericclientfirst": genericclientfirst(19) "dnstcp": dnstcp(20) "sipudp": sipudp(21) "siptcp": siptcp(22) "rtsp": rtsp(23) "stream": stream(24) "l4acceltcp": l4acceltcp(25) "l4acceludp": l4acceludp(26) "l4accelgeneric": l4accelgeneric(27) "l4accelstateless": l4accelstateless(28)
sip_rejected_requests	Number of SIP requests rejected due to them exceeding the maximum amount of memory allocated to the connection. • Type: UInt • SNMP name: "virtualserverSIPRejectedRequests"
sip_total_calls	Total number of SIP INVITE requests seen by this virtual server. • Type: UInt • SNMP name: "virtualserverSIPTotalCalls"

Counter	Description
total_conn	Formerly provided the number of requests received by this virtual server, now deprecated. • Type: UInt • SNMP name: "virtualserverTotalConn"
total_dgram	UDP datagrams processed by this virtual server. • Type: UInt • SNMP name: "virtualserverTotalDgram"
total_http1_requests	HTTP/1.x Requests received by this virtual server. • Type: UInt64 • SNMP name: "virtualserverTotalHTTP1Requests"
total_http1_requests_hi	HTTP/1.x Requests received by this virtual server. • Type: UInt • SNMP name: "virtualserverTotalHTTP1RequestsHi"
total_http1_requests_lo	HTTP/1.x Requests received by this virtual server. • Type: UInt • SNMP name: "virtualserverTotalHTTP1RequestsLo"
total_http2_requests	HTTP/2 Requests received by this virtual server. • Type: UInt64 • SNMP name: "virtualserverTotalHTTP2Requests"
total_http2_requests_hi	HTTP/2 Requests received by this virtual server. • Type: UInt • SNMP name: "virtualserverTotalHTTP2RequestsHi"
total_http2_requests_lo	HTTP/2 Requests received by this virtual server. • Type: UInt • SNMP name: "virtualserverTotalHTTP2RequestsLo"
total_http_requests	HTTP Requests received by this virtual server. • Type: UInt64 • SNMP name: "virtualserverTotalHTTPRequests"
total_http_requests_hi	HTTP Requests received by this virtual server. • Type: UInt • SNMP name: "virtualserverTotalHTTPRequestsHi"
total_http_requests_lo	HTTP Requests received by this virtual server. • Type: UInt • SNMP name: "virtualserverTotalHTTPRequestsLo"
total_requests	Requests received by this virtual server. • Type: UInt64 • SNMP name: "virtualserverTotalRequests"

Counter	Description
total_requests_hi	Requests received by this virtual server. • Type: UInt • SNMP name: "virtualserverTotalRequestsHi"
total_requests_lo	Requests received by this virtual server. • Type: UInt • SNMP name: "virtualserverTotalRequestsLo"
total_tcp_reset	Number of TCP connections reset by this virtual server because the forward traffic cannot be processed. • Type: UInt • SNMP name: "virtualserverL4TCPConnectResets"
total_udp_unreachables	Number of ICMP error responses sent to the client by this virtual server because the forward traffic cannot be processed. • Type: UInt • SNMP name: "virtualserverL4UDPUnreachables"
udp_timed_out	Connections closed by this virtual server because the 'udp_timeout' interval was exceeded. • Type: UInt • SNMP name: "virtualserverUdpTimedOut"

Web cache

URI Path: statistics/cache/web_cache

Web cache statistics values.

Counter	Description
entries	The number of items in the web cache. • Type: UInt • SNMP name: "webCacheEntries"
hit_rate	The percentage of web cache lookups that succeeded. • Type: UInt • SNMP name: "webCacheHitRate"
hits	Number of times a page has been successfully found in the web cache. • Type: UInt64 • SNMP name: "webCacheHits"
hits_hi	Number of times a page has been successfully found in the web cache (high 32 bits). • Type: UInt • SNMP name: "webCacheHitsHi"

Counter	Description
hits_lo	Number of times a page has been successfully found in the web cache (low 32 bits). • Type: UInt • SNMP name: "webCacheHitsLo"
lookups	Number of times a page has been looked up in the web cache. • Type: UInt64 • SNMP name: "webCacheLookups"
lookups_hi	Number of times a page has been looked up in the web cache (high 32 bits). • Type: UInt • SNMP name: "webCacheLookupsHi"
lookups_lo	Number of times a page has been looked up in the web cache (low 32 bits). • Type: UInt • SNMP name: "webCacheLookupsLo"
max_entries	The maximum number of items in the web cache. • Type: UInt • SNMP name: "webCacheMaxEntries"
mem_maximum	The maximum amount of memory the web cache can use in kilobytes. • Type: UInt • SNMP name: "webCacheMemMaximum"
mem_used	Total memory used by the web cache in kilobytes. • Type: UInt • SNMP name: "webCacheMemUsed"
misses	Number of times a page has not been found in the web cache. • Type: UInt64 • SNMP name: "webCacheMisses"
misses_hi	Number of times a page has not been found in the web cache (high 32 bits). • Type: UInt • SNMP name: "webCacheMissesHi"
misses_lo	Number of times a page has not been found in the web cache (low 32 bits). • Type: UInt • SNMP name: "webCacheMissesLo"
oldest	The age of the oldest item in the web cache (in seconds). • Type: UInt • SNMP name: "webCacheOldest"
url_store_allocated	Amount of allocated space in the web cache URL store. • Type: UInt64 • SNMP name: "webCacheURLStoreAllocated"

Counter	Description
url_store_free	Amount of free space in the web cache URL store. • Type: UInt64 • SNMP name: "webCacheURLStoreFree"
url_store_size	Total amount of space in the web cache URL store. • Type: UInt64 • SNMP name: "webCacheURLStoreSize"
url_store_total_allocations	Total number of allocations for the web cache URL store. • Type: UInt64 • SNMP name: "webCacheURLStoreTotalAllocations"
url_store_total_failures	Total number of allocation failures for the web cache URL store. • Type: UInt64 • SNMP name: "webCacheURLStoreTotalFailures"
url_store_total_frees	Total number of blocks freed in the web cache URL store. • Type: UInt64 • SNMP name: "webCacheURLStoreTotalFrees"

System Information Resources

Backups

URI Path: backups/full/*

Full backups.

Property	Description
Properties for the "backup" section:	
description	Description of the backup • Type: String
time_stamp	Time the backup was created. Expressed as a UTC value. • Type: Int
version	Version of Brocade vTM used to create the backup • Type: String

Information

URI Path: information

Static information for the system.

Property	Description
tm_version	Version number of the Traffic Manager instance. • Type: String
uuid	The universally unique identifier for the Traffic Manager instance. • Type: String

State

URI Path: state

State information for the Brocade vTM.

Property	Description
Properties for the "data_plane_acceleration" section:	
capable	Whether or not the traffic manager is capable of running in Data Plane Acceleration Mode. • Type: Boolean
configured	Whether or not the traffic manager configuration requests Data Plane Acceleration Mode. • Type: Boolean
failed_to_start	Whether or not the traffic manager failed to start Data Plane Acceleration Mode. • Type: Boolean
running	Whether or not the traffic manager is running in Data Plane Acceleration Mode. • Type: Boolean
Properties for the "state" section:	
error_level	The error_level of the traffic manager. • Type: Enum(String) • Permitted values: "ok": System has no problems "warn": System has minor issues "error": System has major issues "fatal": System has issues which causes it to die/crash/fail to startup
errors	List of configuration errors for the traffic manager • Type: Set(String)

Property	Description
failed_nodes	A table of nodes which have failed on the traffic manager • Primary key: – node (String): A node which has failed • Sub keys: – pools (Set(String)): List of pools which use this node.
license	Current active license or Developer_Mode • Type: String
pools	• Primary key: – name (String): Name of the pool • Sub keys: – failure_pool (String): Failure pool associated with this pool. – active_nodes (Set(String)): List of nodes which are in the active state. – draining_nodes (Set(String)): List of nodes which are in the draining state. – disabled_nodes (Set(String)): List of nodes which are in the disabled state.
tip_errors	List of traffic IP errors for the traffic manager • Type: Set(String)
virtual_servers	A table of virtual server status • Primary key: – name (String): Name of the virtual server • Sub keys: – pool (String): Pool associated with this virtual server. – port (UInt): Port the virtual server listens on. – throughput (UInt64): Through put for the virtual server. – ts_redirect_pools (Set(String)): List of pools which may be selected by Traffic Script Request Rules.

APPENDIX A

Handling Updates to the REST API

REST API Support in the Traffic Manager

The Traffic Manager typically supports several major versions of its REST API in any one release, and can additionally support several minor versions of each major version. The Traffic Manager employs this system to facilitate upgrading from an older supported version to a newer version with the least possible constraints on development time. Brocade operates a policy of always warning about the impending removal of an API version in the release notes of the preceding Traffic Manager release.

The REST API version number is split into a major and minor component, and is two sets of digits separated by a dot. Using BNF notation, the follows rules apply:

```
<REST API version> ::= <major> "." <minor>
<major> ::= <non-negative integer>
<minor> ::= <non-negative integer>
```

Brocade typically increments the version number in each release of the Traffic Manager, with the extent of the increment dependent on the changes introduced to the API since the preceding release. A significant API update necessitates incrementing the major component of the version number (with the minor version reset to 0). For example, from 3.4 to 4.0. A smaller API change increments only the minor component by 1, for example, from 3.4 to 3.5.

Major changes typically occur in the following situations:

- Removing or renaming a resource.
- Removing or renaming a property within a resource.
- Adding a property without a default to an existing resource.

Minor changes typically occur in the following situations:

- Adding a new resource to the API.
- Adding a new property with a default value to an existing JSON resource.

For more information, see Brocade Support.

Updating Your Applications to Use a New Version of the API

If you are considering adopting a new version of the Traffic Manager REST API for your existing scripts and applications, Brocade strongly recommends you take the following steps in order to minimize the impact of the update:

- Avoid hard-coding the API version number in multiple places in a script or application.
- Avoid mixing requests to different major API versions across a single Traffic Manager cluster. For example, it is unsafe to access a resource at a 3.x API version in one script and a 4.x API version in another.

Changes Involving a Minor API Update

For upgrades involving a minor increment to a currently used major version, for example 3.4 to 3.5, PUTs and GETs made to the previous API version are expected to operate as intended with the newer API version. Brocade recommends you verify that the client behaves as expected when any new keys or resources are returned.

Changes Involving a Major API Update

For upgrades involving a new major API version, for example 3.11 to 4.0, first audit your scripts and applications to determine the list of resources and properties you are accessing. Next, refer to Appendix B, "REST API Change History" for the change history since your current version up to and including the desired new version. Compare your set of accessed resources and properties to the changes listed and note where a resource or properties has been affected, with specific reference to any renames or deletes. Finally, modify your scripts and applications accordingly.

APPENDIX B REST API Change History

Brocade Virtual Traffic Manager release 17.2 includes version 4.0 of the REST API.

This appendix lists the changes introduced in each supported API version. Applications that were developed against older versions of the Traffic Manager REST API might be affected by the updates listed here. Use the information contained in this appendix to identify the necessary updates.

Changes in Version 4.0

The following resources were added:

- Resource 'Log Export' (/api/tm/4.0/config/active/log_export)

The following properties in 'Global Settings' (/api/tm/4.0/config/active/global_settings) have been added:

- Property 'log_export/auth_hec_token' was added.
- Property 'log_export/auth_http' was added.
- Property 'log_export/auth_password' was added.
- Property 'log_export/auth_username' was added.
- Property 'log_export/enabled' was added.
- Property 'log_export/endpoint' was added.
- Property 'log_export/max_event_message_size' was added.
- Property 'log_export/max_request_bandwidth' was added.
- Property 'log_export/max_request_size' was added.
- Property 'log_export/max_response_size' was added.
- Property 'log_export/maximum_error_raising_period' was added.
- Property 'log_export/minimum_error_raising_period' was added.
- Property 'log_export/request_timeout' was added.
- Property 'log_export/tls_verify' was added.

- Property 'transaction_export/auto_brief' was added.
- Property 'transaction_export/enabled' was added.
- Property 'transaction_export/endpoint' was added.
- Property 'transaction_export/failure_interval' was added.
- Property 'transaction_export/memory' was added.
- Property 'transaction_export/reconnect_interval' was added.
- Property 'transaction_export/tls' was added.
- Property 'transaction_export/tls_timeout' was added.
- Property 'transaction_export/tls_verify' was added.

The following properties in 'Bandwidth' (/api/tm/4.0/status/local_tm/statistics/bandwidth/*) have been added:

- Property 'statistics/bytes_drop' was added.
- Property 'statistics/bytes_drop_hi' was added.
- Property 'statistics/bytes_drop_lo' was added.
- Property 'statistics/pkts_drop' was added.
- Property 'statistics/pkts_drop_hi' was added.
- Property 'statistics/pkts_drop_lo' was added.

The following properties in 'Virtual servers' (/api/tm/4.0/status/local_tm/statistics/virtual_servers/*) have been added:

- Property 'statistics/bw_limit_bytes_drop' was added.
- Property 'statistics/bw_limit_bytes_drop_hi' was added.
- Property 'statistics/bw_limit_bytes_drop_lo' was added.
- Property 'statistics/bw_limit_pkts_drop' was added.
- Property 'statistics/bw_limit_pkts_drop_hi' was added.
- Property 'statistics/bw_limit_pkts_drop_lo' was added.
- Property 'statistics/pkts_in' was added.
- Property 'statistics/pkts_in_hi' was added.
- Property 'statistics/pkts_in_lo' was added.
- Property 'statistics/pkts_out' was added.
- Property 'statistics/pkts_out_hi' was added.
- Property 'statistics/pkts_out_lo' was added.

The following properties in 'Pools' (/api/tm/4.0/status/local_tm/statistics/pools/*) have been added:

- Property 'statistics/bw_limit_bytes_drop' was added.
- Property 'statistics/bw_limit_bytes_drop_hi' was added.

- Property 'statistics/bw_limit_bytes_drop_lo' was added.
- Property 'statistics/bw_limit_pkts_drop' was added.
- Property 'statistics/bw_limit_pkts_drop_hi' was added.
- Property 'statistics/bw_limit_pkts_drop_lo' was added.

The following properties in 'Per pool node' (/api/tm/4.0/status/local_tm/statistics/nodes/per_pool_node/*) have been added:

- Property 'statistics/l4_stateless_buckets' was added.
- Property 'statistics/pkts_from_node' was added.
- Property 'statistics/pkts_from_node_hi' was added.
- Property 'statistics/pkts_from_node_lo' was added.
- Property 'statistics/pkts_to_node' was added.
- Property 'statistics/pkts_to_node_hi' was added.
- Property 'statistics/pkts_to_node_lo' was added.

The following properties in 'Globals' (/api/tm/4.0/status/local_tm/statistics/globals) have been added:

- Property 'statistics/analytics_transactions_dropped' was added.
- Property 'statistics/analytics_transactions_exported' was added.
- Property 'statistics/analytics_transactions_memory_usage' was added.

The following properties in 'Virtual Server' (/api/tm/4.0/config/active/virtual_servers) have been added:

- Property 'basic/strip_x_forwarded_proto' was added.
- Property 'transaction_export/brief' was added.
- Property 'transaction_export/enabled' was added.
- Property 'transaction_export/hi_res' was added.
- Property 'transaction_export/http_header_blacklist' was added.

Changes in Version 3.11

The following properties in 'Global Settings' (/api/tm/3.11/config/active/global_settings) have been added:

- Property 'fault_tolerance/autofailback_delay' was added.
- Property 'rest_api/block_for_future_max' was added.
- Property 'rest_api/http_compress_min' was added.
- Property 'rest_api/http_keep_alive_timeout' was added.
- Property 'rest_api/http_max_resource_body_length' was added.

- Property 'rest_api/http_max_write_buffer' was added.
- Property 'rest_api/http_session_timeout' was added.

The following properties in 'Virtual Server' (/api/tm/3.11/config/active/virtual_servers) have been added:

- Property 'basic/proxy_protocol' was added.
- Property 'ssl/issued_certs_never_expire_depth' was added.

The following properties in 'State' (/api/tm/3.11/status/local_tm/state) have been added:

- Property 'state/pools' was added.
- Table property 'state/virtual_servers' field 'ts_redirect_pools' was added.

The following properties in 'Traffic Manager' (/api/tm/3.11/config/active/traffic_managers) have been added:

- Property 'fault_tolerance/lss_dedicated_ips' was added.
- Table property 'appliance/if' field 'mode' was added.

The type of the following properties in 'State' (/api/tm/3.11/status/local_tm/state) has changed:

- Table property 'state/virtual_servers' field 'throughput' changed type from 'Unsigned integer' to 'Unsigned integer 64'.

The following properties for 'Global Settings' (/api/tm/3.11/config/active/global_settings) have had flags change:

- Property 'data_plane_acceleration/tcp_delay_ack' has had the following flags added:
  [RESTART_SOFTWARE]
- Property 'data_plane_acceleration/tcp_win_scale' has had the following flags added:
  [RESTART_SOFTWARE]

Changes in Version 3.10

The following properties in 'Monitor' (/api/tm/3.10/config/active/monitors) have been added:

- Property 'basic/factory' was added.
- Property 'basic/health_only' was added.

The following properties in 'Traffic Manager' (/api/tm/3.10/config/active/traffic_managers) have been added:

- Property 'basic/developer_mode_accepted' was added.

The following properties in 'Virtual Server' (/api/tm/3.10/config/active/virtual_servers) have been added:

- Property 'basic/max_concurrent_connections' was added.
- Property 'basic/rules_on_connect' was added.
- Property 'basic/udp_end_transaction' was added.
- Property 'gzip/chunk' was added.
- Property 'http2/headers_size_limit' was added.

- Property 'l4accel/service_ip_snat' was added.
- Property 'l4stateless/initial_ring_size' was added.
- Property 'l4stateless/num_replicas' was added.

The following properties in 'Protection Class' (/api/tm/3.10/config/active/protection) have been added:

- Property 'basic/linger_time' was added.

The following properties in 'Global Settings' (/api/tm/3.10/config/active/global_settings) have been added:

- Property 'data_plane_acceleration/tcp_delay_ack' was added.
- Property 'data_plane_acceleration/tcp_win_scale' was added.
- Property 'ec2/awstool_timeout' was added.
- Property 'ec2/metadata_timeout' was added.
- Property 'l4accel/max_concurrent_connections' was added.
- Property 'remote_licensing/owner' was added.
- Property 'remote_licensing/owner_secret' was added.
- Property 'remote_licensing/policy_id' was added.

Changes in Version 3.9

The following resources were added:

- Resource 'Backups' (/api/tm/3.9/status/local_tm/backups/full/*)

The following properties in 'Virtual Server' (/api/tm/3.9/config/active/virtual_servers) have been added:

- Property 'basic/bypass_data_plane_acceleration' was added.
- Property 'l4accel/rst_on_service_failure' was added.
- Property 'l4accel/state_sync' was added.
- Property 'l4accel/tcp_msl' was added.
- Property 'l4accel/timeout' was added.
- Property 'l4accel/udp_count_requests' was added.
- Property 'ssl/max_key_size' was added.
- Property 'ssl/min_key_size' was added.

The following properties in 'Traffic Manager' (/api/tm/3.9/config/active/traffic_managers) have been added:

- Property 'appliance/managedpa' was added.
- Property 'basic/num_l4_children' was added.
- Property 'basic/num_l7_children' was added.
- Property 'iop/is_standalone' was added.
- Property 'iop/l4_event_driven_mode' was added.

- Property 'iop/l7_event_driven_mode' was added.
- Property 'iop/linux_interface' was added.
- Property 'iop/num_mbufs_per_mpool' was added.
- Property 'iop/send_to_linux' was added.

The following properties in 'Traffic IP Group' (/api/tm/3.9/config/active/traffic_ip_groups) have been added:

- Property 'basic/snat_ipaddresses' was added.

The following properties in 'Pool' (/api/tm/3.9/config/active/pools) have been added:

- Property 'l4accel/snat' was added.

The following properties in 'Global Settings' (/api/tm/3.9/config/active/global_settings) have been added:

- Property 'basic/data_plane_acceleration_cores' was added.
- Property 'basic/data_plane_acceleration_mode' was added.
- Property 'fault_tolerance/l4accel_child_timeout' was added.
- Property 'fault_tolerance/l4accel_sync_port' was added.
- Property 'source_nat/clist_locks' was added.
- Property 'source_nat/ip_limit' was added.
- Property 'source_nat/ip_local_port_range_high' was added.
- Property 'source_nat/portmaphashtable_locks' was added.
- Property 'source_nat/shared_pool_size' was added.

The following properties in 'Virtual servers' (/api/tm/3.9/status/local_tm/statistics/virtual_servers/*) have been added:

- Property 'statistics/total_tcp_reset' was added.
- Property 'statistics/total_udp_unreachables' was added.

The following properties in 'Session Persistence Class' (/api/tm/3.9/config/active/persistence) have been added:

- Property 'basic/subnet_mask_len_v4' was added.
- Property 'basic/subnet_mask_len_v6' was added.

The following properties for 'Pool' (/api/tm/3.9/config/active/pools) have had flags change:

- Property 'ssl/ssl_support_ssl2' has had the following flags added: [SOAP_IGNORE]

The following properties for 'Global Settings' (/api/tm/3.9/config/active/global_settings) have had flags change:

- Property 'admin/support_ssl2' has had the following flags added: [SOAP_IGNORE]
- Property 'ssl/support_ssl2' has had the following flags added: [SOAP_IGNORE]

The following properties for 'Virtual Server' (/api/tm/3.9/config/active/virtual_servers) have had flags change:

- Property 'ssl/ssl_support_ssl2' has had the following flags added: [SOAP_IGNORE]

Changes in Version 3.8

The following properties in 'Virtual Server' (/api/tm/3.8/config/active/virtual_servers) have been added:

- Property 'log/always_flush' was added.
- Property 'ssl/server_cert_alt_certificates' was added.
- Table property 'ssl/server_cert_host_mapping' field 'alt_certificates' was added.

The following properties in 'Pool' (/api/tm/3.8/config/active/pools) have been added:

- Property 'basic/lard_size' was added.
- Property 'udp/response_timeout' was added.

The following properties in 'Event Type' (/api/tm/3.8/config/active/event_types) have been added:

- Property 'basic/log2mainlog' was added.

The following properties in 'Traffic Manager' (/api/tm/3.8/config/active/traffic_managers) have been added:

- Property 'basic/use_mx' was added.
- Property 'remote_licensing/email_address' was added.
- Property 'remote_licensing/message' was added.

The following properties in 'Globals' (/api/tm/3.8/status/local_tm/statistics/globals) have been added:

- Property 'statistics/ssl_cipher_ecdsa_signs' was added.
- Property 'statistics/ssl_cipher_ecdsa_verifies' was added.

The following properties in 'Global Settings' (/api/tm/3.8/config/active/global_settings) have been added:

- Property 'dns/checktime' was added.
- Property 'dns/hosts' was added.
- Property 'dns/hostsfirst' was added.
- Property 'dns/maxasynctries' was added.
- Property 'dns/resolv' was added.
- Property 'fault_tolerance/child_timeout' was added.
- Property 'http/max_chunk_header_length' was added.
- Property 'remote_licensing/registration_server' was added.
- Property 'remote_licensing/script_timeout' was added.
- Property 'remote_licensing/server_certificate' was added.
- Property 'ssl_hardware/nworkers' was added.
- Property 'ssl_hardware/queuelen' was added.
- Property 'web_cache/blocksize' was added.
- Property 'web_cache/max_byte_range_segments' was added.

- Property ‘web_cache/min_size_accept_range’ was added.

Changes in Version 3.7

The following resources were added:

- Resource 'State' (/api/tm/3.7/status/local_tm/state)

The following properties in 'Virtual Server' (/api/tm/3.7/config/active/virtual_servers) have been added:

- Property 'basic/mss' was added.

The following properties in 'Traffic IP Group' (/api/tm/3.7/config/active/traffic_ip_groups) have been added:

- Property 'basic/ip_assignment_mode' was added.

The following properties in 'Traffic Manager' (/api/tm/3.7/config/active/traffic_managers) have been added:

- Property 'appliance/ipmi_lan_access' was added.
- Property 'appliance/ipmi_lan_addr' was added.
- Property 'appliance/ipmi_lan_gateway' was added.
- Property 'appliance/ipmi_lan_ipsrc' was added.
- Property 'appliance/ipmi_lan_mask' was added.
- Property 'basic/appliance_card' was added.

Changes in Version 3.6

The following properties in 'Traffic Manager' (/api/tm/3.6/config/active/traffic_managers) have been added:

- Property 'appliance/ipv6_forwarding' was added.
- Property 'appliance/ssh_password_allowed' was added.

The following properties in 'Virtual Server' (/api/tm/3.6/config/active/virtual_servers) have been added:

- Property 'dns/edns_client_subnet' was added.
- Property 'http2/connect_timeout' was added.
- Property 'http2/data_frame_size' was added.
- Property 'http2/enabled' was added.
- Property 'http2/header_table_size' was added.
- Property 'http2/idle_timeout_no_streams' was added.
- Property 'http2/idle_timeout_open_streams' was added.
- Property 'http2/max_concurrent_streams' was added.
- Property 'http2/max_frame_size' was added.

- Property 'http2/max_header_padding' was added.
- Property 'http2/neverindex_blacklist' was added.
- Property 'http2/neverindex_default' was added.
- Property 'http2/neverindex_whitelist' was added.
- Property 'http2/stream_window_size' was added.

The following properties in 'Global Settings' (/api/tm/3.6/config/active/global_settings) have been added:

- Property 'auditlog/via_eventd' was added.
- Property 'basic/cluster_identifier' was added.
- Property 'basic/cpu_starvation_check_interval' was added.
- Property 'basic/cpu_starvation_check_tolerance' was added.
- Property 'basic/http2_no_cipher_blacklist_check' was added.
- Property 'web_cache/max_handles' was added.

Changes in Version 3.5

The following resources were added:

- Resource 'BGP Neighbor' (/api/tm/3.5/config/active/bgpneighbors)

The following properties in 'Virtual Server' (/api/tm/3.5/config/active/virtual_servers) have been added:

- Property 'basic/auto_upgrade_protocols' was added.
- Property 'basic/autodetect_upgrade_headers' was added.
- Property 'basic/transparent' was added.

The following properties in 'Traffic Manager' (/api/tm/3.5/config/active/traffic_managers) have been added:

- Property 'fault_tolerance/bgp_router_id' was added.
- Property 'iptables/config_enabled' was added.
- Property 'iptrans/chain' was added.
- Property 'iptrans/fwmark' was added.
- Property 'iptrans/iptables_enabled' was added.
- Property 'iptrans/routing_table' was added.

The following properties in 'Information' (/api/tm/3.5/status/local_tm/information) have been added:

- Property 'information/uuid' was added.

The following properties in 'Pool' (/api/tm/3.5/config/active/pools) have been added:

- Property 'basic/node_delete_behavior' was added.
- Property 'basic/node_drain_to_delete_timeout' was added.

The following properties in 'Monitor' (/api/tm/3.5/config/active/monitors) have been added:

- Property 'basic/can_edit_ssl' was added.
- Property 'basic/can_use_ssl' was added.
- Property 'basic/editable_keys' was added.

The following properties in 'Traffic IP Group' (/api/tm/3.5/config/active/traffic_ip_groups) have been added:

- Property 'basic/rhi_bgp_metric_base' was added.
- Property 'basic/rhi_bgp_passive_metric_offset' was added.
- Property 'basic/rhi_protocols' was added.

The following properties in 'Global Settings' (/api/tm/3.5/config/active/global_settings) have been added:

- Property 'basic/license_servers' was added.
- Property 'bgp/as_number' was added.
- Property 'bgp/enabled' was added.
- Property 'ssl/ssl3_diffie_hellman_client_min_key_length' was added.

The following properties in 'Security Settings' (/api/tm/3.5/config/active/security) have been added:

- Property 'ssh_intrusion/bantime' was added.
- Property 'ssh_intrusion/blacklist' was added.
- Property 'ssh_intrusion/enabled' was added.
- Property 'ssh_intrusion/findtime' was added.
- Property 'ssh_intrusion/maxretry' was added.
- Property 'ssh_intrusion/whitelist' was added.

The following properties for 'Virtual Server' (/api/tm/3.5/config/active/virtual_servers) have had flags change:

- Property 'basic/glb_services' has had the following flags removed: [SOAP_IGNORE]

Changes in Version 3.4

The following properties in 'Web cache' (/api/tm/3.4/status/local_tm/statistics/cache/web_cache) have been added:

- Property 'statistics/url_store_allocated' was added.
- Property 'statistics/url_store_free' was added.
- Property 'statistics/url_store_size' was added.
- Property 'statistics/url_store_total_allocations' was added.
- Property 'statistics/url_store_total_failures' was added.
- Property 'statistics/url_store_total_frees' was added.

The following properties in 'Pool' (/api/tm/3.4/config/active/pools) have been added:

- Property 'auto_scaling/addnode_delaytime' was added.

- Property 'ssl/common_name_match' was added.
- Property 'ssl/elliptic_curves' was added.

The following properties in 'GLB Service' (/api/tm/3.4/config/active/glb_services) have been added:

- Property 'basic/peer_health_timeout' was added.

The following properties in 'Virtual Server' (/api/tm/3.4/config/active/virtual_servers) have been added:

- Property 'dns/rrset_order' was added.
- Property 'gzip/etag_rewrite' was added.
- Property 'ssl/elliptic_curves' was added.

The following properties in 'Globals' (/api/tm/3.4/status/local_tm/statistics/globals) have been added:

- Property 'statistics/ssl_cipher_ecdh_agreements' was added.
- Property 'statistics/ssl_cipher_ecdh_generates' was added.

The following properties in 'Global Settings' (/api/tm/3.4/config/active/global_settings) have been added:

- Property 'admin/ssl_elliptic_curves' was added.
- Property 'auditlog/via_syslog' was added.
- Property 'ssl/elliptic_curves' was added.
- Property 'ssl_hardware/azure_api_version' was added.
- Property 'ssl_hardware/azure_client_id' was added.
- Property 'ssl_hardware/azure_client_secret' was added.
- Property 'ssl_hardware/azure_connect_timeout' was added.
- Property 'ssl_hardware/azure_idle_timeout' was added.
- Property 'ssl_hardware/azure_vault_url' was added.
- Property 'ssl_hardware/azure_verify_rest_api_cert' was added.
- Property 'web_cache/url_store_keep_free' was added.
- Property 'web_cache/url_store_max_mallocs' was added.
- Property 'web_cache/url_store_num_bins' was added.

The following properties in 'Traffic Manager' (/api/tm/3.4/config/active/traffic_managers) have been added:

- Property 'appliance/manageazureroutes' was added.

Changes in Version 3.3

The following resources were added:

- Resource 'DNS Zone File' (/api/tm/3.3/config/active/dns_server/zone_files)
- Resource 'DNS Zone' (/api/tm/3.3/config/active/dns_server/zones)

The following properties in 'Global Settings' (/api/tm/3.3/config/active/global_settings) have been added:

- Property 'admin/honor_fallback_scsv' was added.
- Property 'ssl/honor_fallback_scsv' was added.

The following properties in 'Globals' (/api/tm/3.3/status/local_tm/statistics/globals) have been added:

- Property 'statistics/ssl_cipher_aes_gcm_decrypts' was added.
- Property 'statistics/ssl_cipher_aes_gcm_encrypts' was added.

The following properties in 'Traffic Manager' (/api/tm/3.3/config/active/traffic_managers) have been added:

- Property 'autodiscover/product_id' was added.
- Property 'basic/kmod_policy' was added.
- Property 'basic/start_sysd' was added.

The following properties in 'Virtual Server' (/api/tm/3.3/config/active/virtual_servers) have been added:

- Property 'basic/ssl_honor_fallback_scsv' was added.
- Property 'dns/verbose' was added.
- Property 'dns/zones' was added.

The following properties in 'GLB Service' (/api/tm/3.3/config/active/glb_services) have been added:

- Property 'basic/autorecovery' was added.
- Property 'basic/disable_on_failure' was added.
- Property 'basic/last_resort_response' was added.

The following properties for 'Virtual Server' (/api/tm/3.3/config/active/virtual_servers) have had flags change:

- Property 'dns/edns_udpsize' has had the following flags removed: [SOAP_IGNORE]
- Property 'dns/max_udpsize' has had the following flags removed: [SOAP_IGNORE]
- Property 'ssl/prefer_sslv3' has had the following flags added: [SOAP_IGNORE]

Changes in Version 3.2

The following properties in "Pool" (/api/tm/3.2/config/active/pools) have been added:

- "Property "ssl/signature_algorithms" was added.
- "Property "ssl/ssl_support_tls1_2" was added.

The following properties in "Virtual servers" (/api/tm/3.2/status/local_tm/statistics/virtual_servers/*) have been added:

- "Property "statistics/max_duration_timed_out" was added.
- "Property "statistics/processing_timed_out" was added.

The following properties in "Globals" (/api/tm/3.2/status/local_tm/statistics/globals) have been added:

- "Property "statistics/ssl_cipher_dh_agreements" was added.

- "Property "statistics/ssl_cipher_dh_generates" was added.
- "Property "statistics/ssl_cipher_dsa_signs" was added.
- "Property "statistics/ssl_cipher_dsa_verifies" was added.
- "Property "statistics/ssl_handshake_t_l_sv12" was added.

The following properties in "Traffic Manager" (/api/tm/3.2/config/active/traffic_managers) have been added:

- "Property "basic/adminMasterXMLIP" was added.
- "Property "basic/adminSlaveXMLIP" was added.
- "Property "basic/authenticationServerIP" was added.
- "Property "basic/updaterIP" was added.
- "Property "ec2/trafficips_public_enis" was added.

The following properties in "Global Settings" (/api/tm/3.2/config/active/global_settings) have been added:

- "Property "admin/ssl_signature_algorithms" was added.
- "Property "admin/support_tls12" was added.
- "Property "aptimizer/max_concurrent_jobs" was added.
- "Property "ec2/action_timeout" was added.
- "Property "ssl/signature_algorithms" was added.
- "Property "ssl/support_tls1_2" was added.

The following properties in "Virtual Server" (/api/tm/3.2/config/active/virtual_servers) have been added:

- "Property "dns/edns_udpsize" was added.
- "Property "dns/max_udpsize" was added.
- "Property "ssl/signature_algorithms" was added.
- "Property "ssl/ssl_support_tls1_2" was added.

The following properties in "Rules" (/api/tm/3.2/status/local_tm/statistics/rules/*) have been added:

- "Property "statistics/execution_time_warnings" was added.

Changes in Version 3.1

The following resources were added:

- Resource "User counters 32" (/api/tm/3.1/status/local_tm/statistics/extras/user_counters_32)
- Resource "User counters 64" (/api/tm/3.1/status/local_tm/statistics/extras/user_counters_64)

The following properties in "Global Settings" (/api/tm/3.1/config/active/global_settings) have been added:

- Property "ec2/verify_query_server_cert" was added.
- Property "fault_tolerance/igmp_interval" was added.

- Property "ssl/cache_per_virtualserver" was added.
- Property "ssl/ocsp_max_response_size" was added.
- Property "trafficscript/execution_time_warning" was added.

The following properties in "Virtual Server" (/api/tm/3.1/config/active/virtual_servers) have been added:

- Property "basic/close_with_rst" was added.
- Property "log/session_persistence_verbose" was added.

The following properties in "Pool" (/api/tm/3.1/config/active/pools) have been added:

- Property "basic/node_close_with_rst" was added.

The following properties in "Traffic Manager" (/api/tm/3.1/config/active/traffic_managers) have been added:

- Property "basic/cloud_platform" was added.
- Property "basic/disk_serious" was added.
- Property "basic/disk_warn" was added

Changes in Version 3.0

Note: The Traffic Manager REST representation of pool nodes has changed for this release. All nodes and related properties are now contained within a single table (`nodes_table`). For more information, see the **Pool** configuration resource in the "Resource Model Reference" chapter of the *REST API Guide* for Stingray Traffic Manager release 9.6.

The following resources were added:

- Resource 'NAT Configuration' (/api/tm/3.0/config/active/appliance/nat)

The following properties in 'Global Settings' (/api/tm/3.0/config/active/global_settings) have been added:

- Property 'autoscaler/slm_interval' was added.
- Property 'autoscaler/verbose' was added.
- Property 'fault_tolerance/tipv6_raise_deprecated' was added.
- Property 'ssl/disable_stitched_cbc_hmac' was added.
- Property 'ssl/ocsp_stapling_maximum_refresh_interval' was added.
- Property 'ssl/ocsp_stapling_time_tolerance' was added.
- Property 'ssl/ocsp_stapling_verify_response' was added.

The following properties in 'Pool' (/api/tm/3.0/config/active/pools) have been added:

- Property 'auto_scaling/extraargs' was added.
- Property 'auto_scaling/securitygroupids' was added.
- Property 'auto_scaling/subnetids' was added.
- Property 'basic/nodes_table' was added.

- Property 'ssl/ssl_ciphers' was added.
- Property 'ssl/ssl_support_ssl2' was added.
- Property 'ssl/ssl_support_ssl3' was added.
- Property 'ssl/ssl_support_tls1' was added.
- Property 'ssl/ssl_support_tls1_1' was added.

The following properties in 'Traffic Manager' (/api/tm/3.0/config/active/traffic_managers) have been added:

- Property 'appliance/ipv4_forwarding' was added.

The following properties in 'Virtual Server' (/api/tm/3.0/config/active/virtual_servers) have been added:

- Property 'basic/completionrules' was added.
- Property 'connection/max_transaction_duration' was added.
- Property 'log/save_all' was added.
- Property 'recent_connections/enabled' was added.
- Property 'recent_connections/save_all' was added.
- Property 'ssl/ssl_ciphers' was added.
- Property 'ssl/ssl_support_ssl2' was added.
- Property 'ssl/ssl_support_ssl3' was added.
- Property 'ssl/ssl_support_tls1' was added.
- Property 'ssl/ssl_support_tls1_1' was added.

The following properties in 'Pool' (/api/tm/3.0/config/active/pools) have been removed:

- Property 'basic/disabled' was removed.
- Property 'basic/draining' was removed.
- Property 'basic/nodes' was removed.
- Property 'load_balancing/node_weighting' was removed.
- Property 'load_balancing/priority_values' was removed.