ivanti

Pulse Secure Virtual Traffic Manager: Configuration System Guide

Copyright Notice

Contents

Preface

Document conventions

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

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 command 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.

Convention	Description
italic text	Identifies a variable.
[ ]	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.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Non-printing 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.

Notes and Warnings

Note, Attention, and Caution statements might be used in this document.

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

Attention

An Attention 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.

Requesting Technical Support

Technical product support is available through the Ivanti Support Center. If you have a support contract, file a ticket with support.

- Product warranties—For product warranty information, visit https://forums.ivanti.com/s/contactsupport?language=en_US

Self-Help Online Tools and Resources

For quick and easy problem resolution, Ivanti provides an online self-service portal called the Support Center that provides you with the following features:

- Find support offerings: https://forums.ivanti.com/s/contactsupport?language=en_US
- Search for known bugs: https://forums.ivanti.com/s/contactsupport?language=en_US
- Find product documentation: https://www.ivanti.com/support/product-documentation
- Download the latest versions of software and review release notes: https://forums.ivanti.com/s/contactsupport?language=en_US
- Open a case online in the support Case Management tool: https://forums.ivanti.com/s/contactsupport?language=en_US
- To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://forums.ivanti.com/s/contactsupport?language=en_US

For important product notices, technical articles, and to ask advice:

- Search the Knowledge Center for technical bulletins and security advisories: https://forums.ivanti.com/s/searchallcontent?language=en_US#t=KNOWLEDGE%20BASE&sort=relevancy
- Ask questions and find solutions at the Ivanti Community online forum: https://forums.ivanti.com/s/?language=en_US

Opening a Case with Support

You can open a case with Support on the Web or by telephone.

- Use the Case Management tool in the support at https://forums.ivanti.com/s/contactsupport?language=en_US.

For international or direct-dial options in countries without toll-free numbers, see https://forums.ivanti.com/s/contactsupport?language=en_US

Introduction

This chapter provides an introduction to the Pulse Secure Virtual Traffic Manager (Traffic Manager) configuration system. This chapter contains the following sections:

About this Guide

Ivanti intends this guide to be used by system administrators wanting to manually manipulate the configuration system of the Traffic Manager, and in particular, users of the zconf command line utility.

The Traffic Manager is available in a variety of software and appliance configurations. All configurations share the same core Traffic Manager software and configuration system, but different variants can provide different levels of functionality depending on the enabling license key.

See Configuration Sections for a complete list of all configuration sections present in the core software, irrespective of license key. Each section includes a brief summary of its purpose and a table of any available configuration keys. Each key is displayed with its description and a list of its attributes. Such attributes include the key type and default value.

Consider the following restrictions on each key:

- If no default value is given then a value MUST be specified for the key;
- Some values are picked/tuned at install so may not be the specified default;
- (file)names must not contain certain illegal characters: ._#-`!/* (or any control characters);
- Newlines cannot be used in key values;
- Some keys have certain dependencies. A “requires” attribute is displayed where this is the case.

This manual describes the configuration for this version.

The Traffic Manager Configuration File System

The Traffic Manager stores its configuration in a series of text files under a tree structure of directories, one per object type. As you make configuration changes through the UI or one of the product APIs, the Traffic Manager's "Admin Server" management component maintains and updates the files and directories accordingly.

ATTENTION
Unless instructed to do so by your support provider, Ivanti strongly recommends that you do not modify these files directly. They are maintained automatically by the Admin Server and as such your changes can be lost or cause unpredictable behavior in your Traffic Manager deployment. Always use the UI, programming APIs, or zconf utility to make changes.

The core Traffic Manager software reads the config when triggered by an observed update, or on a pre-defined chronological basis, and applies the appropriate logic. Additionally, the Traffic Manager ensures synchronization between itself and all other Traffic Manager instances in a cluster, and replicates out any changes that occur. The following diagram demonstrates the directory structure under the root config directory:

```
ZEUSHOME/zxtm/conf/
                        |
                        +-actionprogs/
                        |
                        +-actions/
                        |
                        +-activitymonitor/
                        |
                        +-auth/
                        |
                        +-bandwidth/
                        |
                        …
                        …
                        …
                        |
                        +-vservers/
                        |
                        +-zxtms/
```

Each of the sections listed in Configuration Sections typically have a file or directory entry under /conf in this way.

When you add a new object of a particular type, the Traffic Manager creates a new config text file with the same name and stores it under the corresponding sub-directory. For example, it you create a new virtual server called “myvirtualserver”, the Traffic Manager creates a new text file:

```
ZEUSHOME/zxtm/conf/vservers/myvirtualserver
```

Each config file consists of lines of key-value pairs, separated by new-line characters, in the following format:

```
<key> <value>[ <next value>…]
```

The key refers to some setting or feature, and the value is the item, or list of items, applied to that key. The key and value are separated by whitespace, and where the value is actually a list, each item is again separated by whitespace.

The key might be *simple* or *compound*. Compound keys are used to group related settings together, and consist of a common component and subsequent sub-components, separated by a ‘!’ character.

You can add comments into config files, pre-pended by the hash (#) character.

The following is an example virtual server config file (name on disk: ZEUSHOME/zxtm/conf/vservers/Intranet) that demonstrates all of the above features:

```
# This is an example config file for a
# virtual server named 'Intranet'
Address *
Enabled Yes
Pool Intranet-pool
Port 80
Protection servprot1
request_tracing!enabled Yes
request_tracing!trace_io Yes
responserules headeradjust test_rule
rules
slm slm-class1
timeout 40
webcache!enabled Yes
```

The Effect of Location Support when using Multi-Site Cluster Management

You can configure the Traffic Manager to provide support for management of multiple distributed physical, virtual or cloud-based Traffic Manager clusters. This is implemented in the configuration system by appending location names to the relevant keys in affected configuration files. The Traffic Manager uses the at (@) symbol followed by the location name, so key “foo” would become “foo@location”.

For example, a simple config key might be:

```
Enabled Yes
```

By adding location support, this key would become:

```
Enabled@cambridge Yes
Enabled@sanfrancisco Noyep
```

This convention ensures that config keys set as specific to a particular location are ignored by other locations.

For further information regarding location support, see the Multi-site Cluster Management chapter of the *Pulse Secure Virtual Traffic Manager: User’s Guide*.

Configuration Sections

This chapter provides a complete reference of the Traffic Manager configuration system. Each section refers to a specific configuration type, and lists all applicable configuration keys contained therein.

conf/actionprogs

The conf/actionprogs directory contains programs and scripts that can be used by actions of the program type. Action programs can be managed under the Catalogs > Extra Files > Action Programs section of the Admin Server UI or by using functions under the Alerting.Action section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/actions

The conf/actions directory contains configuration files for event handlers. The name of a file is the name of the action it defines. Actions can be configured under the System > Alerting section of the Admin Server UI or by using functions under the Alerting.Action section of the the SOAP API and CLI.

Key	Description
note	A description of the action. Value type: string Default value: <none>
type	The action type. Value type: enumeration Default value: <none> Permitted values: email: E-Mail log: Log to File syslog: Log to Syslog program: Program trap: SNMP Notify or Trap soap: SOAP Callback

Key	Description
Additional keys used when type is "email"	
from	The e-mail address from which messages will appear to originate. Requires: type is set to "email" Value type: string Default value: "vTM@%hostname%"
server	The SMTP server to which messages should be sent. This must be a valid IPv4 address or resolvable hostname (with optional port). Requires: type is set to "email" Value type: string Default value: <none>
to	A list of e-mail addresses to which messages will be sent. Requires: type is set to "email" Value type: string Default value: <none>
Additional keys used when type is "log"	
file	The full path of the file to log to. The text %zeushome% will be replaced with the location where the software is installed. Requires: type is set to "log" Value type: string Default value: <none>
Additional keys used when type is "program"	
arg!*	An argument to pass to the program. For example, to specify that the argument --foo=bar should be passed to the program executed by this action you would set a key arg!foo to the value bar. Requires: type is set to "program" Value type: string Default value: <none>
describe!*	A description for an argument provided to the program, this is used only for display purposes in the Admin Server UI. To give a description to the --foo example for arg!* above you would set the description text as the value for the key description!foo. Requires: type is set to "program"

Key	Description
	Value type: string Default value: <none>
program	The program to run. Requires: type is set to "program" Value type: string Default value: <none>
Additional keys used when type is "program", "email", or "soap"	
timeout	How long the action can run for before it is stopped automatically (set to 0 to disable timeouts). Requires: type is set to "program", "email", or "soap" Value type: seconds Default value: "60"
Additional keys used when type is "program", or "email"	
verbose	Enable or disable verbose logging for this action. Requires: type is set to "program", or "email" Value type: Yes / No Default value: "No"
Additional keys used when type is "soap"	
additional	Additional information to send with the SOAP call. Requires: type is set to "soap" Value type: string Default value: <none>
password	The password for HTTP basic authentication. Requires: type is set to "soap" Value type: password Default value: <none>
proxy	The address of the server implementing the SOAP interface (For example, https://example.com). Requires: type is set to "soap" Value type: string Default value: <none>

Key	Description
username	Username for HTTP basic authentication. Leave blank if you do not wish to use authentication. Requires: type is set to "soap" Value type: string Default value: <none>
Additional keys used when type is "syslog"	
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. Requires: type is set to "syslog" Value type: unsigned integer Default value: "2048"
sysloghost	The host and optional port to send syslog messages to (if empty, messages will be sent to localhost). Requires: type is set to "syslog" Value type: string Default value: <none>
Additional keys used when type is "trap"	
community	The community string to use when sending a Trap over SNMPv1 or a Notify over SNMPv2c. Requires: type is set to "trap" Value type: string Default value: <none>
snmp!auth_password	The authentication password for sending a Notify over SNMPv3. Blank to send unauthenticated traps. Requires: type is set to "trap" Value type: password Default value: <none>
snmp!hash_alg	The hash algorithm for SNMPv3 authentication. Requires: type is set to "trap" Value type: enumeration Default value: "md5" Permitted values: md5: MD5

Key	Description
	sha1: SHA-1
snmp!priv_password	The encryption password to encrypt a Notify message for SNMPv3. Requires that authentication also be configured. Blank to send unencrypted traps. Requires: type is set to "trap" Value type: password Default value: <none>
snmp!username	The SNMP username to use to send the Notify over SNMPv3. Requires: type is set to "trap" Value type: string Default value: <none>
snmp!version	The SNMP version to use to send the Trap/Notify. Requires: type is set to "trap" Value type: enumeration Default value: "snmpv1" Permitted values: snmpv1: SNMPv1 snmpv2c: SNMPv2c snmpv3: SNMPv3
traphost	The hostname or IPv4 address and optional port number that should receive traps. Requires: type is set to "trap" Value type: string Default value: <none>

conf/appliance/nat.cfg

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

Key	Description
many_to_one_ overload!*!pool	Pool of a "many to one overload" type NAT rule. Value type: string Default value: <none>

Key	Description
many_to_one_overload!*!tip	TIP Group of a "many to one overload" type NAT rule. Value type: string Default value: <none>
many_to_one_port_ locked!*!pool	Pool of a "many to one port locked" type NAT rule. Value type: string Default value: <none>
many_to_one_port_ locked!*!port	Port number of a "many to one port locked" type NAT rule. Value type: unsigned integer Default value: <none>
many_to_one_port_ locked!*!protocol	Protocol of a "many to one port locked" type NAT rule. Value type: enumeration Default value: <none> Permitted values: tcp: TCP udp: UDP udplite: UDPLITE sctp: SCTP icmp: ICMP
many_to_one_port_ locked!*!tip	TIP Group of a "many to one port locked" type NAT rule. Value type: string Default value: <none>
one_to_one!*!enable_ inbound	Enabling the inbound part of a "one to one" type NAT rule. Value type: Yes / No Default value: <none>
one_to_one!*!ip	IP Address of a "one to one" type NAT rule. Value type: string Default value: <none>
one_to_one!*!tip	TIP group of a "one to one" type NAT rule. Value type: string Default value: <none>
port_mapping!*!dport_first	First port of the dest. port range of a "port mapping" rule. Value type: unsigned integer

Key	Description
	Default value: <none>
port_mapping!*!dport_last	Last port of the dest. port range of a "port mapping" rule. Value type: unsigned integer Default value: <none>
port_mapping!*!virtual_server	Target Virtual Server of a "port mapping" rule. Value type: string Default value: <none>

conf/aptimizer/profiles

The conf/aptimize directory contains configuration files for Web Accelerator profiles. Web Accelerator profiles can be configured under the Catalogs > Web Accelerator > Web Accelerator Profiles section of the Admin Server UI or by using functions under the Catalog.Aptimizer.AptimizerProfiles section of the SOAP API and CLI.

Key	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. Value type: unsigned integer Default value: "0"
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. Value type: Yes / No Default value: "No"
built_in	If set to Yes this indicates that this configuration is built-in (provided as part of the software) and cannot be deleted or edited.

Key	Description
	Value type: Yes / No Default value: "No"
config	The configuration string for the Web Accelerator profile. Value type: string Default value: <none>
mode	Set the Web Accelerator mode to turn acceleration on or off. Value type: enumeration Default value: "active" Permitted values: idle: Off - Acceleration is disabled, but requests for Web Accelerator resources are served stealth: Stealth - Acceleration is controlled by a cookie active: On - Web Accelerator acceleration is enabled
show_info_bar	Show the Web Accelerator information bar on optimized web pages. This requires HTML optimization to be enabled in the acceleration settings. Value type: Yes / No Default value: "No"

conf/aptimizer/scopes

Priority should be higher than that of virtual servers

Key	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. Value type: string Default value: <none>
hostnames	The hostnames to limit acceleration to. Value type: list Default value: <none>
root	The root path of the application defined by this application scope.

Key	Description
	Value type: string Default value: "/"

conf/auth

The conf/auth directory contains configuration files for remote authentication services used to control access to the software. The name of a file is the name of the authenticator it defines. Authenticators can be configured under the System > Users section of the Admin Server UI.

Key	Description
auth!description	A description of the authenticator. Value type: string Default value: <none>
auth!enabled	Whether or not this authenticator is enabled. Value type: Yes / No Default value: "No"
auth!type	The type of the authenticator. Value type: enumeration Default value: <none> Permitted values: LDAP: LDAP RADIUS: RADIUS TACACSPlus: TACACS+
Additional keys used when type is "LDAP"	
ldap!basedn	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 Requires: auth!type is set to "LDAP" Value type: string Default value: <none>

Key	Description
ldap!binddn	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. Requires: auth!type is set to "LDAP" Value type: string Default value: <none>
ldap!dnmethod	The bind DN (Distinguished Name) for a user can either be searched for in the directory using the ldap!basedn and ldap!filter values, or it can be constructed from the username. Requires: auth!type is set to "LDAP" Value type: enumeration Default value: <none> Permitted values: construct: Construct search: Search
ldap!fallbackgroup	If ldap!groupattr 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. Requires: auth!type is set to "LDAP" Value type: string Default value: <none>
ldap!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 ldap!dnmethod is set to "Search", and to extract group information if ldap!groupfilter is not specified. Examples: sAMAccountName=%u for Active Directory, or uid=%u for some Unix LDAP schemas. Requires: auth!type is set to "LDAP" Value type: string Default value: <none>

Key	Description
ldap!groupattr	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. Requires: auth!type is set to "LDAP" Value type: string Default value: <none>
ldap!groupfield	The sub-field of the group attribute that gives a user's group. For example, if ldap!groupattr is memberOf and this retrieves values of the form CN=mygroup, OU=groups, OU=users, DC=mycompany, DC=local you would set groupfield to CN. If there are multiple matching fields only the first matching field will be used. Requires: auth!type is set to "LDAP" Value type: string Default value: <none>
ldap!groupfilter	If the user record returned by ldap!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)) Requires: auth!type is set to "LDAP" Value type: string Default value: <none>
ldap!port	The port to connect to the LDAP server on. Requires: auth!type is set to "LDAP" Value type: unsigned integer Default value: "389"
ldap!searchdn	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. Requires: auth!type is set to "LDAP" Value type: string

Key	Description
	Default value: <none>
ldap!searchpass	If binding to the LDAP server using ldap!searchdn requires a password, enter it here. Requires: auth!type is set to "LDAP" Value type: password Default value: <none>
ldap!server	The IP or hostname of the LDAP server. Requires: auth!type is set to "LDAP" Value type: string Default value: <none>
ldap!ssl	The type of TLS encryption, if any, to use. Usually STARTTLS will be used with port 389, and LDAPS with port 636. A Certificate Authority that the LDAP server's certificate chains back to must be present in the "Admin Certificate Authorities and Certificate Revocation Lists Catalog" under "SSL catalogs", otherwise the connection will fail. Requires: auth!type is set to "LDAP" Value type: enumeration Default value: "none" Permitted values: none: None starttls: STARTTLS ldaps: LDAPS
ldap!timeout	Connection timeout in seconds. Requires: auth!type is set to "LDAP" Value type: unsigned integer Default value: "30"
Additional keys used when type is "RADIUS"	
radius!fallbackgroup	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. Requires: auth!type is set to "RADIUS" Value type: string Default value: <none>

Key	Description
radius!groupattr	The RADIUS identifier for the attribute that specifies an account's group. May be left blank if radius!fallbackgroup is specified. Requires: auth!type is set to "RADIUS" Value type: unsigned integer Default value: "1"
radius!groupvendor	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 radius!groupattr to 11). Requires: auth!type is set to "RADIUS" Value type: unsigned integer Default value: "7146"
radius!nas-identifier	This value is sent to the RADIUS server. Requires: auth!type is set to "RADIUS" Value type: string Default value: <none>
radius!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. Requires: auth!type is set to "RADIUS" Value type: string Default value: <none>
radius!port	The port to connect to the RADIUS server on. Requires: auth!type is set to "RADIUS" Value type: unsigned integer Default value: "1812"
radius!secret	Secret key shared with the RADIUS server. Requires: auth!type is set to "RADIUS" Value type: password Default value: <none>
radius!server	The IP or hostname of the RADIUS server. Requires: auth!type is set to "RADIUS" Value type: string Default value: <none>
radius!timeout	Connection timeout in seconds.

Key	Description
	Requires: auth!type is set to "RADIUS" Value type: unsigned integer Default value: "30"
Additional keys used when type is "TACACSPlus"	
tacacsplus!authtype	Authentication type to use. Requires: auth!type is set to "TACACSPlus" Value type: enumeration Default value: "PAP" Permitted values: PAP: PAP ASCII: ASCII
tacacsplus!fallbackgroup	If tacacsplus!groupsvc is not defined above, 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. Requires: auth!type is set to "TACACSPlus" Value type: string Default value: <none>
tacacsplus!groupfield	The TACACS+ "service" field that provides each user's group. Requires: auth!type is set to "TACACSPlus" Value type: string Default value: "permission-group"
tacacsplus!groupsvc	The TACACS+ "service" that provides each user's group field. Requires: auth!type is set to "TACACSPlus" Value type: string Default value: "zeus"
tacacsplus!port	The port to connect to the TACACS+ server on. Requires: auth!type is set to "TACACSPlus" Value type: unsigned integer Default value: "49"
tacacsplus!secret	Secret key shared with the TACACS+ server. Requires: auth!type is set to "TACACSPlus" Value type: password

Key	Description
	Default value: <none>
tacacsplus!server	The IP or hostname of the TACACS+ server. Requires: auth!type is set to "TACACSPlus" Value type: string Default value: <none>
tacacsplus!timeout	Connection timeout in seconds. Requires: auth!type is set to "TACACSPlus" Value type: unsigned integer Default value: "30"

conf/authenticators

The conf/authenticators directory contains configuration files for external authenticators. The name of a file is the name of the authenticator it defines. Authenticators can be configured under the Catalogs > Authenticators section of the Admin Server UI or by using functions under the Catalog.Authenticators section of the SOAP API and CLI.

Key	Description
host	The hostname or IP address of the remote authenticator. Value type: string Default value: <none>
ldap!attr	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. Value type: list Default value: <none>
ldap!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. Value type: string Default value: <none>
ldap!bind!password	The password for the bind user. Value type: password

Key	Description
	Default value: <none>
ldap!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. Value type: string Default value: <none>
ldap!filter!basedn	The base distinguished name (DN) under which user records are located on the server. Value type: string Default value: <none>
ldap!ssl	Whether or not to enable SSL encryption to the LDAP server. Value type: Yes / No Default value: "No"
ldap!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. Value type: string Default value: <none>
ldap!ssl!type	The type of LDAP SSL encryption to use. Value type: enumeration Default value: "ldaps" Permitted values: ldaps: LDAPS starttls: Start TLS
note	A description of the authenticator. Value type: string Default value: <none>
port	The port on which the remote authenticator should be contacted. Value type: unsigned integer Default value: "389"

conf/bandwidth

The conf/bandwidth directory contains configuration files for bandwidth classes. The name of a file is the name of the bandwidth class it defines. Bandwidth classes can be configured under the Catalogs > Bandwidth section of the Admin Server UI or by using functions under the Catalog.Bandwidth section of the SOAP API and CLI.

Key	Description
maximum	The maximum bandwidth to allocate to connections that are associated with this bandwidth class (in kbits/second). Value type: unsigned integer Default value: "10000"
note	A description of this bandwidth class. Value type: string Default value: <none>
sharing	The scope of the bandwidth class. Value type: enumeration Default value: "cluster" Permitted values: connection: Each connection can use the maximum rate machine: Bandwidth is shared per traffic manager cluster: Bandwidth is shared across all traffic managers

conf/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.

Key	Description
address	The IP address of the BGP neighbor Value type: string Default value: <none>

Key	Description
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. Value type: seconds Default value: "5"
as_number	The AS number for the BGP neighbor Value type: unsigned integer Default value: "65534"
authentication_password	The password to be used for authentication of sessions with neighbors Value 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. Value type: seconds Default value: "90"
keepalive	The interval at which messages are sent to the BGP neighbor to keep the mutual BGP session established. Value type: seconds Default value: "30"
machines	The traffic managers that are to use this neighbor Value type: list Default value: <none>

conf/cloudcredentials

Configuration for cloud credentials used in cloud API calls.

Key	Description
api_server	The vCenter server hostname or IP address.

Key	Description
	Value type: string Default value: <none>
change_process_timeout	The amount of time a change process can take at most. The traffic manager creates and destroys nodes via API calls. This setting specifies how long to wait for such calls to complete. Value type: unsigned integer Default value: "200"
cred1	The first part of the credentials for the cloud user. Typically this is some variation on the username concept. Value 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. Value 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. Value type: password Default value: <none>
script	The script to call for communication with the cloud API. Value 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. Value type: unsigned integer Default value: "30"

conf/commkey

The conf/commkey file is for internal use only. You should never manually alter this file.

Key	Description
There are no items to display for this configuration type.	

conf/custom

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

Key	Description
stringlist!*	Named list of user-specified strings. Value type: list Default value: <none>

conf/dnsserver/zonefiles

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

Key	Description
There are no items to display for this configuration type.	

conf/dnsserver/zones

The conf/dnsserver/zones/ file contains zone metadata

Key	Description
origin	The domain origin of this Zone. Value type: string Default value: <none>
zonefile	The Zone File encapsulated by this Zone. Value type: string Default value: <none>

conf/events

The conf/events directory contains configuration files that tie actions to a set of events. In the web UI this functionality is controlled using the System > Alerting and System > Alerting > Event Types pages. The configuration files in conf/events represent the functionality configured on both these pages. The name of the configuration files are the "Event Type" names as shown in the UI. In the SOAP API and CLI this is managed in the Alerting.EventType section. The events subscribed to by a particular event type configuration are identified by an object type and a set of event tags using keys of the form "type!<object-type>!event_tags <tag-list>". For example: "type!vservers!event_tags vsstart vsstop". The events subscribed to can be further filtered to specific configuration objects using keys of the form "type!<object-type>!object_names <object-names>". The table below lists the object types and all the event tags that are available for them.

Key	Description
actions	The actions triggered by events matching this event type. (See the type!*!event_tags and type!*!object_names keys.) The value is a list of files to execute when a matching event occurs, these files must be located within the conf/actions directory. Refer to the documentation for the conf/actions configuration section for more information regarding how these files are executed. Value type: list Default value: <none>
built_in	If set to Yes this indicates that this configuration is built-in (provided as part of the software) and cannot be deleted or edited. Value type: Yes / No Default value: "No"
note	A description of this event type. Value type: string Default value: <none>

Key	Description
type!*!event_tags	This key is used to specify the object types and event tags that will trigger the configured actions. The object type is specified in place of the * (asterisk) in the key name. The key can be used multiple times in a configuration file to subscribe to events from multiple object types. The value can be * (asterisk) to subscribe to all events raised by the specified object type, or can be a list of specific event tags (refer to the table in the conf/events section documentation for a list of all object types and event tags). The following example sends an email alert when any virtual server starts or stops: actions E-Mail type!vservers!event_tags vsstart vsstop type!vservers!object_names * If this is in a file named conf/events/VSStartStop then on the System > Alerting UI page a mapping will be shown associating the event type "VSStartStop" with the action "E-Mail". See type!*!object_ names for additional information. Value type: list Default value: <none>
type!*!object_names	This key can be used to restrict the events that will trigger the configured actions to ones raised by objects with specific names (filenames). The * (asterisk) in the key must be replaced by an object type matching one that has also been used in a type!<object-type>!event_tags key. The value is a list containing the names of objects of the type specified in the key. (If this key is not specified then the default value of * (asterisk) is assumed, which means to subscribe to events from all objects of the given type). The following example sends email alert whenever the virtual server named "Very Important" starts or stops: actions E-Mail type!vservers!event_tags vsstart vsstop type!vservers!object_names "Very Important"

Key	Description
	If this is in a file named conf/events/VSStartStop then on the System > Alerting UI page a mapping will be shown associating the event type "VSStartStop" with the action "E-Mail". See type!*!event_tags for additional information. Value type: list Default value: <none>

Event tags by object type

Event Tag	Description
Event tags for object type: "cloudcredentials"	
apistatusprocesshanging	A cloud API process querying changes to cloud instances is hanging
autoscaleresponseparseerror	An API call made by the autoscaler process has returned a response that could not be parsed
autoscalestatusupdateerror	An API call made by the autoscaler process has reported an error
autoscalingprocesstimedout	A cloud API process has timed out
usedcredsdeleted	A Cloud Credentials object has been deleted but it was still in use (The configuration file containing cloud credentials was removed, but the credentials were still being used by one or more autoscaled pools.)
Event tags for object type: "config"	
confadd	Configuration file added
confdel	Configuration file deleted
confmod	Configuration file modified
confok	Configuration file now OK

Event Tag	Description
Event tags for object type: "faulttolerance"	
activatealldead	Activating this machine automatically because it is the only working machine in its Traffic IP Groups
activatedautomatically	Machine has recovered and been activated automatically because it would cause no service disruption
allmachinesok	All machines are working (All machines are working)
autofailbackafterdelay	Automatic failback after delay
autofailbacktimercancelled	Auto-failback delay timer cancelled
autofailbacktimerstarted	Auto-failback wait period started
autofailbacktimerstopped	Auto-failback delay timer stopped due to system failure
bgpneighborsdegraded	Some of the BGP neighbors do not have established sessions
bgpneighborsfailed	None of the BGP neighbors have an established session (None of the BGP neighbors have an established session)
bgpneighborsok	There are established sessions with all BGP neighbors (There are established sessions with all BGP neighbors)
bgpnoneighbors	There are no valid BGP neighbors defined (There are no valid BGP neighbors defined)
clockjump	The system clock jumped forwards or backwards by more than one second
clocknotmonotonic	The monotonic system clock went backwards
dropec2ipwarn	Removing EC2 IP Address from all machines; it is no longer a part of any Traffic IP Groups
dropgceipwarn	Removing GCE IP Address from all machines; it is no longer a part of any Traffic IP Groups

Event Tag	Description
dropipinfo	Dropping Traffic IP Address due to a configuration change or traffic manager recovery
dropipwarn	Dropping Traffic IP Address due to an error (The Traffic IP address was dropped due to a network failure)
ec2flipperraiselocalworking	Moving EC2 IP Address; local machine is working
ec2flipperraiseothersdead	Moving EC2 IP Address; other machines have failed
ec2iperr	Problem occurred when managing an EC2 IP address
ec2nopublicip	Cannot raise Elastic IP on this machine until EC2 provides it with a public IP address (An Elastic IP cannot currently be moved to this machine. This is usually because it has recently had its Elastic IP moved to another box, and EC2 has not yet returned its default public IP address.)
ec2nosecondaryprivateip	Cannot raise Elastic IP on this machine as no suitable secondary IP is available on the allowed network card(s) (An Elastic IP cannot currently be moved to this machine. This is usually because it doesn't have a secondary private address with either no EIP association assigned to network interface(s) or is used by a virtual server.)
flipperbackendsworking	Back-end nodes are now working (Back-end nodes are now working)
flipperdadreraise	Re-raising Traffic IP Address; Operating system did not fully raise the address (This address is being re-raised to circumvent the operating system's Duplicate Address Detection feature)
flipperfrontendsworking	Frontend machines are now working (The machines that your traffic manager is using to check network connectivity on the frontend (usually the default gateway) are now working)

Event Tag	Description
flipperipexists	Failed to raise Traffic IP Address; the address exists elsewhere on your network and cannot be raised
flipperraiselocalworking	Raising Traffic IP Address; local machine is working
flipperraiseosdrop	Raising Traffic IP Address; Operating System had dropped this IP address (Traffic IP Addresses are automatically managed by the traffic manager, and their configuration should only be altered from the vTM Admin Server.)
flipperraiseothersdead	Raising Traffic IP Address; other machines have failed (The Traffic IP Address will be raised as a result of the the death of another machine, or a config change.)
flipperraiseremotedropped	This Traffic Manager has re-raised traffic IP addresses as the remote machine which was hosting them has dropped them
flipperrecovered	Machine is ready to raise Traffic IP addresses
gceflipperraiselocalworking	Moving GCE IP Address; local machine is working
gceflipperraiseothersdead	Moving GCE IP Address; other machines have failed
gceiperr	Problem occurred when managing a GCE IP address
gcenofreenic	Cannot associate more External IP addresses with this instance, all interfaces are allocated (Cannot associate an external IP with this instance, all NICs are reserved)
machinefail	Remote machine has failed
machineok	Remote machine is now working (Remote machine is now working)
machinerecovered	Remote machine has recovered and can raise Traffic IP addresses
machinetimeout	Remote machine has timed out and been marked as failed

Event Tag	Description
multihostload	The amount of load handled by the local machine destined for this Traffic IP has changed
ospfneighborsdegraded	Some of the monitored OSPF neighbors are not peered (Some of the neighboring OSPF routers being monitored by flipper!ospfv2_neighbor_addrs are not peered)
ospfneighborsfailed	None of the monitored OSPF neighbors are peered (None of the neighboring OSPF routers being monitored by flipper!ospfv2_neighbor_addrs are peered)
ospfneighborsok	All monitored OSPF neighbors are peered (The neighboring OSPF routers being monitored by flipper!ospfv2_neighbor_addrs are all peered)
pingbackendfail	Failed to ping back-end nodes
pingfrontendfail	Failed to ping any of the machines used to check the front-end connectivity
pinggwfail	Failed to ping default gateway
pingsendfail	Failed to send ping packets
routingswfailed	Routing software had a major failure and will be restarted (The routing software stack used for Route Health Injection has had a major failure and will be restarted.)
routingswfailurelimitreached	Routing software has failed and reached its failure limit (The maximum number of failures in a set period has been reached by the routing software stack used for Route Health Injection.)
routingswoperational	Routing software is now operational (The routing software stack used for Route Health Injection has started.)
routingswstartfailed	Routing software failed to start

Event Tag	Description
	(The routing software stack used for Route Health Injection failed to start within the allowed time.)
statebaddata	Received an invalid response from another cluster member (An incorrectly formatted session persistence state message was received (for example version incompatibility between traffic managers).)
stateconnfail	Failed to connect to another cluster member for state sharing (The traffic manager failed to establish the connection used for session persistence state sharing.)
stateok	Successfully connected to another cluster member for state sharing
statereadfail	Reading state data from another cluster member failed (The traffic manager failed to read session persistence information from another traffic manager.)
statetimeout	Timeout while sending state data to another cluster member (Another traffic manager in the cluster failed to respond to a session persistence state message within (2 * 'state_sync_time').)
stateunexpected	Received unexpected state data from another cluster member (A session persistence state message was received when the traffic manager was not expecting it.)
statewritefail	Writing state data to another cluster member failed (The traffic manager failed to write session persistence state to another cluster member.)
zclustermoderr	An error occurred when using the zcluster Multi-Hosted IP kernel module
Event tags for object type: "general"	

Event Tag	Description
analyticsconnected	Analytics Transaction Metadata Export connected (A successful connection has been made to the transaction export endpoint)
analyticsconnectionerror	Analytics Transaction Metadata Export connection failure (Failed to make a connection to the transaction export endpoint)
analyticsdisconnected	Analytics Transaction Metadata Export disconnected (The connection to the transaction export endpoint was broken)
appfirewallcontrolerror	Application firewall control command failed
appfirewallcontrolrestarted	Application firewall restarted (Application firewall restarted)
appfirewallcontrolstarted	Application firewall started (Application firewall started)
appfirewallcontrolstopped	Application firewall stopped (Application firewall stopped)
appfirewallcontroltimeout	Application firewall control command timed out
appliance	Appliance notification
audit	An audit log event has occured
autherror	An error occurred during user authentication
autoscaleresolvefailure	A hostname used for DNS-derived Autoscaling doesn't resolve
autoscalinglicenseerror	Autoscaling not permitted by licence key
childcommsfail	There was an error communicating with a child process (A helper process did not properly acknowledge a control request.)
commchanneldied	SD Communications Channel Agent died (SD Communications Channel Agent died)

Event Tag	Description
commchannelstartfail	Failed to start SD Communications Channel Agent (Failed to start SD Communications Channel Agent)
commchannelterminatefail	SD Communications Channel Agent failed to terminate (The SD Communications Channel Agent failed to terminate promptly)
confrepfailed	Replication of configuration has failed
confreptimeout	Replication of configuration has timed out (Replication of configuration has timed out)
discopluginfailed	Traffic manager failed to get the required output from the Service Discovery plugin (Traffic manager failed to get the required output from the plugin)
discoplugininfo	Service Discovery plugin returned extra logging information (Plugin returned extra information which should be logged)
discopluginvalid	Service Discovery plugin returned invalid output (Plugin returned a node with missing information)
discopluginstartfail	Traffic manager failed to start the Service Discovery Plugin (Failed to run a plugin)
discopluginstartsuccess	Traffic manager has now successfully run the Service Discovery plugin (Traffic manager has now successfully run the plugin)
discopluginwarning	Service Discovery succeeded but provided a warning (Plugin succeeded but gave a warning in the error field.)
dnszonecreaterecord	The built-in DNS server has failed to create a DNS record
dnszoneparse	The built-in DNS server has failed to parse a DNS zone file

Event Tag	Description
dnszonevalidate	The built-in DNS server has failed to validate a DNS zone file
ec2dataretrievalfailed	Traffic manager failed to get the required data from Amazon servers (Traffic manager failed to get the required data from Amazon servers)
ec2dataretrievalsuccessful	Traffic manager has now successfully retrieved the required data from Amazon servers (Traffic manager has now successfully retrieved the required data from Amazon servers)
ec2initialized	The EC2 instance is now initialized (The EC2 instance is now initialized)
fewfreefds	Running out of free file descriptors (There are few free file descriptors remaining; this machine will soon become unable to establish new connections. See the manual for information on tuning to correct this.)
fipsfailinit	FIPS 140-2 cryptographic module initialization failed (A failure occurred when loading or during power-up testing of the FIPS 140-2 cryptographic module.)
fipsfailops	FIPS 140-2 cryptographic module operations failed (Unable to fully enable or retain the context for valid use of the FIPS 140-2 cryptographic module in the Traffic Manager.)
gcedataretrievalfailed	Traffic manager failed to get the required data from GCE instance
gcedataretrievalsuccessful	Traffic manager has now successfully retrieved the required data from GCE instance (Traffic manager has now successfully retrieved the required data from GCE instance)
geodataloadfail	Failed to load geolocation data

Event Tag	Description
licensetoomanylocations	A location has been disabled because you have exceeded the licence limit
logdiskfull	Log disk partition full (Log disk partition full)
logdiskoverload	Log disk partition usage has exceeded threshold (Log disk partition usage has exceeded threshold)
logdiskrecovered	Log disk partition usage has recovered (Log disk partition usage has recovered)
nameserveravailable	DNS-derived Autoscaling will resume updating, as the DNS server is now responding (DNS-derived Autoscaling will resume updating, as the DNS server is now responding)
nameserverunavailable	DNS-derived Autoscaling will not update, as the DNS server is unavailable (DNS-derived Autoscaling will not update, as the DNS server is unavailable)
numlocations-exceeded	Total number of locations exceeded the maximum limit
numnodes-exceeded	Total number of nodes exceeded the maximum number of nodes that can be monitored
numpools-exceeded	Total number of pools exceeded the maximum limit
numtipg-exceeded	Total number of traffic IP group exceeded the maximum limit
ocspstaplingfail	OCSP request (for OCSP stapling) failed (An OCSP request for a certificate, to be used for OCSP stapling has failed. The error log line contains the name of the certificate and the URL to which the request was made.)
ocspstaplingnomem	Insufficient memory for OCSP stapling

Event Tag	Description
	(The memory allocated for OCSP stapling was not large enough to store the responses for all configured certificates.)
ocspstaplingrevoked	An OCSP request (for OCSP stapling) reported that a certificate was revoked (An OCSP request for a certificate, to be used for OCSP stapling, was successful but reported that the certificate was revoked. The error log line contains the name of the certificate and the URL to which the request was made.)
ocspstaplingunknown	An OCSP request (for OCSP stapling) reported that a certificate was unknown (An OCSP request for a certificate, to be used for OCSP stapling, was successful but reported that the certificate was unknown. The error log line contains the name of the certificate and the URL to which the request was made.)
ocspstaplingunrevoked	An old but good OCSP response was returned for a revoked certificate (An OCSP request for a certificate previously indicated that a certificate had been revoked, but a recent response indicates that it is OK. This may indicate an OCSP replay attack. The error log line contains the name of the certificate and the URL to which the request was made.)
restartrequired	Software must be restarted to apply configuration changes
running	Software is running
sslcrltoobig	CRL does not fit in the configured amount of shared memory, increase ssl!crl_mem!size and restart software
sslticketencryptionkeyunavailable	No SSL ticket encryption key available

Event Tag	Description
	(No SSL ticket encryption key is available. The traffic manager will not be able to create new SSL session tickets until a new key has been provided. SSL connections to virtual servers will still be accepted while this error persists.)
sysctlreboot	Virtual Traffic Manager Appliance reboot required (A reboot is required)
timemovedback	Time has been moved back (This machine's clock has been set backwards by a significant amount; your traffic manager should be restarted to prevent problems with timeouts, fault tolerance and other areas.)
unspecifiedreboot	Virtual Traffic Manager restart/reboot required (A restart/reboot is required, cause unspecified)
upgradereboot	Virtual Traffic Manager Appliance reboot required (A reboot is required)
upgraderestart	Virtual Traffic Manager software restart required (A restart is required)
watchdog	Traffic Manager process stall detected by the watchdog (A traffic manager process stall was detected by the watchdog. This can be caused by overloaded virtual machine infrastructure, or by software defects. The program state has been logged for inclusion in a technical support request.)
zxtmcpustarvation	The number of simultaneously active connections has reached a level that the software cannot process in due time because of CPU starvation; there is a high risk of connections timing out
zxtmhighload	The number of simultaneously active connections has reached a level that the software cannot process in due time; there is a high risk of connections timing out

Event Tag	Description
zxtmswerror	Internal software error
Event tags for object type: "java"	
javadied	Java runner died
javanotfound	Cannot start Java runner, program not found
javastarted	Java runner started
javastartfail	Java runner failed to start
javastop	Java support has stopped (Java is now either unlicensed or disabled in Global Settings.)
javaterminatefail	Java runner failed to terminate (The process handling Java extensions failed to terminate promptly. Contact your support provider.)
servleterror	Servlet encountered an error
Event tags for object type: "licensekeys"	
analyticslicensedisabled	Realtime Analytics support has been disabled
analyticslicenseenabled	Realtime Analytics support has been enabled
autoscalinglicensedisabled	Autoscaling support has been disabled
autoscalinglicenseenabled	Autoscaling support has been enabled
bwlimited	License key bandwidth limit has been hit
cachesizereduced	Configured cache size exceeds license limit, only using amount allowed by license
communityeditionclustertoobig	Cluster size exceeds the Community Edition limit
expired	License expired (The license in use has expired.)
expiresoon	License key expires within 7 days
expiresoon15	License key expires within 15 days

Event Tag	Description
expiresoon30	License key expires within 30 days
expiresoon60	License key expires within 60 days
expiresoon90	License key expires within 90 days
lessmemallowed	License allows less memory for caching
license-authorized	License key authorized (License key authorized)
license-authorized-ts	License key authorized by authorization code (License key authorized by authorization code)
license-explicitlydisabled-ts	License key explicitly disabled from authorization code
license-graceperiodexpired	Unable to authorize license key
license-graceperiodexpired-ts	Unable to authorize license key
license-rejected-authorized	License server rejected license key; key remains authorized
license-rejected-authorized-ts	License key rejected from authorization code; key remains authorized
license-rejected-unauthorized	License server rejected license key; key is not authorized (License server rejected license key; key is not authorized)
license-rejected-unauthorized-ts	License key rejected from authorization code
license-timedout-authorized	Unable to contact license server; license key remains authorized
license-timedout-authorized-ts	Unable to run authorization code to completion; key remains valid
license-timedout-unauthorized	Unable to contact license server; license key is not authorized (Unable to contact license server; license key is not authorized)

Event Tag	Description
license-timedout-unauthorized-ts	Unable to run authorization code to completion
license-unauthorized	License not authorized by remote server (The current license had not been authorized by the license server.)
licenseclustertoobig	Cluster size exceeds license key limit
licensecorrupt	Invalid License (This key is invalid and cannot be used; you should upload a valid key.)
licensestate-malformed	Error detected in LicenseStateFile format
licensestate-write-failed	Unable to preserve license state (The license state file could not be updated.)
morememallowed	License allows more memory for caching
ssltpslimited	License key SSL transactions-per-second limit has been hit
tpslimited	License key transactions-per-second limit has been hit
unlicensed	Started without a license
usinglicense	Using license key (This license key currently determines the available features, because it has more features than any other available keys.)
webcachelicensenewmax	A new limit on the maximum cache size is in place
Event tags for object type: "locations"	
locationavailable	Location is now available for GLB Service (Location is now available for GLB Service)
locationdisabled	Location has been disabled for GLB Service (Location has been disabled for GLB Service)
locationdraining	Location is being drained for GLB Service (Location is being drained for GLB Service)

Event Tag	Description
locationenabled	Location has just been enabled for GLB Service (Location has just been enabled for GLB Service)
locationfail	Location has failed for GLB Service (Location has failed for GLB Service)
locationmonitorfail	A monitor has detected a failure in this location
locationmonitorok	A monitor has indicated this location is now working
locationnotdraining	Location is not being drained for GLB Service (Location is not being drained for GLB Service)
locationok	Location is now healthy for GLB Service (Location is now healthy for GLB Service)
locationsoapfail	An external SOAP agent has detected a failure in this location (An external SOAP agent has detected a failure in this location)
locationsoapok	An external SOAP agent indicates this location is now working (An external SOAP agent indicates this location is now working)
locationunavailable	Location has become unavailable for GLB Service (Location has become unavailable for GLB Service)
locempty	Location no longer contains any machines
locmovemachine	Machine now in location
Event tags for object type: "monitors"	
monitorfail	Monitor has detected a failure
monitorok	Monitor is working
Event tags for object type: "pools"	
apichangeprocesshanging	API change process still running after refractory period is over

Event Tag	Description
autonodecreationcomplete	The creation of a new node requested by an autoscaled pool is now complete
autonodecreationstarted	Creation of new node instigated
autonodedestroyed	A cloud API call to destroy a node has been started
autonodedestructioncomplete	The destruction of a node in an autoscaled pool is now complete
autonodedisappeared	A node in an autoscaled pool has disappeared from the cloud
autonodeexisted	IP address of newly created instance already existed in pool's node list (The autoscaler has been informed about the completion of an instance creation in the cloud, but unexpectedly a node with the same ip address already existed in the pool's node list.)
autonodenopublicip	Node has no public IP address (We want the public IP but it is unset)
autonoderemoved	A node in an DNS-derived autoscaled pool has been removed
autonodestatuschange	The status of a node in an autoscaled pool has changed (The status of a node in an autoscaled pool has changed. This can be, for instance, a node moving from the 'pending' state, when it is still being created/finalized in the cloud environment, to the 'active' state, when it can be fully used.)
autoscalednodecontested	Two pools are trying to use the same instance
autoscaledpoolrefractory	An autoscaled pool is now refractory (An autoscaled pool's size has recently changed, so no further changes are made until it has settled down)
autoscaledpooltoobig	Over maximum size - shrinking

Event Tag	Description
autoscaledpooltoosmall	Minimum size undercut - growing
autoscaleinvalidargforcreatenode	The 'imageid' was empty when attempting to create a node in an autoscaled pool
autoscaleinvalidargfordeletenode	'unique id' was empty when attempting to destroy a node in an autoscaled pool
autoscalepoolconfupdate	A pool config file has been updated by the autoscaler process
autoscalewrongimageid	A node created by the autoscaler has the wrong imageid
autoscalewrongname	A node created by the autoscaler has a non-matching name
autoscalewrongsizeid	A node created by the autoscaler has the wrong sizeid
autoscalingchangeprocessfailure	An API process that should have created or destroyed a node has failed to produce the expected result
autoscalingdisabled	Autoscaling for a pool has been disabled due to errors communicating with the cloud API
autoscalinghitfloor	Minimum size reached, cannot shrink further
autoscalinghitroof	Maximum size reached by autoscaled pool, cannot grow further
autoscalinghysteresiscantgrow	An autoscaled pool is waiting to grow (An autoscaled pool should grow according to its response statistics, but the hysteresis setting demands that the growth condition persist for a longer time before the pool actually creates a new node.)
autoscalinghysteresiscantshrink	An autoscaled pool is waiting to shrink (An autoscaled pool should shrink according to its response statistics, but the hysteresis setting demands that the shrink condition persist for a longer time before the pool actually creates a new node.)
autoscalingpoolstatechange	An autoscaled pool's state has changed

Event Tag	Description
autoscalingresuscitatepool	An autoscaled pool has failed completely
badcontentlen	HTTP response contained an invalid Content-Length header (The HTTP response contained an invalid 'Content-Length' header. The traffic manager will not be able to detect the end of the response and the response cannot make use of keep-alives.)
cannotshrinkemptypool	Attempt to scale down a pool that only had pending nodes or none at all
ehloinvalid	Node returned invalid EHLO response
nodedrainingtodelete	Removed node is in use and will be drained (Removed node is in use and will be drained)
nodedrainingtodeletetimeout	Draining to delete period for node has expired (Draining to delete period has timed out for node.)
nodefail	Node has failed
noderesolvefailure	Failed to resolve node address
noderesolvemultiple	Node resolves to multiple IP addresses
nodeworking	Node is working again
nostarttls	Node doesn't provide STARTTLS support
pooldied	Pool has no back-end nodes responding
poolnonodes	Pool configuration contains no valid backend nodes
poolok	Pool now has working nodes (One or more nodes are now available for this pool.)
starttlsinvalid	Node returned invalid STARTTLS response
Event tags for object type: "protection"	
triggersummary	Summary of recent service protection events

Event Tag	Description
	(Service protection has generated a summary of recent events. The frequency of these messages is configured by log_time on each service protection class.)
Event tags for object type: "rules"	
aptimizedisabled	Rule attempted to use Web Accelerator but it is not enabled
aptimizeuseunknownprofile	Rule selected an unknown Web Accelerator profile
aptimizeuseunknownscope	Rule selected an unknown Web Accelerator scope
datalocalstorefull	data.local.set() has run out of space
datastorefull	data.set() has run out of space (data.set() operations will continue to fail until data.remove() or data.reset() is used)
forwardproxybadhost	Rule selected an unresolvable host (A rule selected a host which could not be resolved to an IP address)
invalidemit	Rule used event.emit() with an invalid custom event (The event ID was empty or contained invalid characters.)
norate	Rule selected an unknown rate shaping class
poolactivenodesunknown	Rule references an unknown pool via pool.activenodes
pooluseunknown	Rule selected an unknown pool
ruleabort	Rule aborted during execution
rulebodycomperror	Rule encountered invalid data while uncompressing response (Rule could not decompress a compressed HTTP response body)
rulebufferlarge	Rule has buffered more data than expected

Event Tag	Description
	(A rule is using more data than expected according to the configuration setting trafficscript!memory_warning in the Global Settings page. This is a warning only; this connection will continue to be handled. If many connections exceed the limit at the same time, your traffic manager might slow down or run out of memory. Consider re-writing the rule to reduce its memory usage or changing the limit.)
rulelogmsginfo	Rule logged an info message using log.info
rulelogmsgserious	Rule logged an error message using log.error
rulelogmsgwarn	Rule logged a warning message using log.warn
rulenopersistence	Rule selected an unknown session persistence class
ruleoverrun	Rule exceeded execution time warning threshold
rulesinvalidrequestbody	Client sent invalid HTTP request body (Invalid request body data encountered by rule)
rulestreamerrorgetresponse	Attempt to use http.getResponse or http.getResponseBody after http.stream.startResponse (Attempt to use http.getResponse or http.getResponseBody after http.stream.startResponse.)
rulestreamerrorinternal	Internal error while processing HTTP stream
rulestreamerrornotenough	Rule did not supply enough data in HTTP stream (Rule had specified a content length but then supplied less data than advertised. Correct Content-Length header in rule/Java Extension or remove it altogether.)
rulestreamerrornotfinished	Attempt to initialize HTTP stream before previous stream had finished (Either a rule called http.stream.startResponse() twice or a rule failed to call http.stream.finishResponse() and its connection was kept-alive. Check the use of the http.stream.* TrafficScript * functions in your rules.)

Event Tag	Description
rulestreamerrornotstarted	Attempt to stream data or finish a stream before streaming had been initialized (A rule called http.stream.writeResponse() or http.stream.finishResponse() before calling http.stream.startResponse(). Check the use of the http.stream.* TrafficScript functions in your rules.)
rulestreamerrorprocessfailure	Data supplied to HTTP stream could not be processed (The data provided by a rule for streaming could not be processed successfully. Check the use of the http.stream.* TrafficScript functions in your rules.)
rulestreamerrortoomuch	Rule supplied too much data in HTTP stream (Rule had specified a content length but then supplied more data than advertised. Correct Content-Length header in rule/Java Extension or remove it altogether.)
rulexmlerr	Rule encountered an XML error
serviceruleabort	GLB service rule aborted during execution
servicerulelocdead	GLB service rule specified a location that has either failed or been marked as draining in the service configuration
servicerulelocnotconfigured	GLB service rule specified a location that is not configured for the service
servicerulelocunknown	GLB service rule specified an unknown location
Event tags for object type: "services"	
glbactivedcmismatch	Active datacentre mismatches among cluster members
glbdeadlocmissingips	A DNS Query returned IP addresses that are not configured for any location that is currently alive
glbfailalter	Failed to alter DNS packet for global load balancing (The DNS packet could not be altered. This usually occurs when the record is signed using DNSSEC, and there is no private key configured to re-sign it.)

Event Tag	Description
glblogwritefail	Failed to write log file for GLB service
glbmanualfailback	Manual failback triggered (Manual failback triggered)
glbmissingips	A DNS Query returned IP addresses that are not configured in any location
glbnewmaster	A location has been set as active for a GLB service
glbnolocations	No valid location could be chosen for Global Load Balancing
glbservicedied	GLB Service has failed (GLB Service has failed)
glbserviceok	GLB Service has recovered (GLB Service has recovered)
glbtoomanylocations	There are too many Data Centers configured and the Global Load Balancing feature is not guaranteed to work reliably with more than 255 Data Centres
Event tags for object type: "slm"	
slmclasslimitexceeded	SLM shared memory limit exceeded (The number of SLM classes configured requires more shared memory than is currently reserved for them. SLM classes will continue to work, but with reduced accuracy. For full accuracy, please increase the configuration key slm_class_limit on the Global Settings page and restart your traffic manager.)
slmfallenbelowserious	SLM has fallen below serious threshold (The percentage of requests meeting the monitor's criteria has fallen below the serious threshold.)
slmfallenbelowwarn	SLM has fallen below warning threshold (A lower percentage of requests meet this monitor's criteria than desired, and this was not the case at the previous check.)

Event Tag	Description
slmnodeinfo	Node information when SLM is non-conforming (no SNMP trap) (A summary of the nodes that have contributed to the SLM failure when it falls below the serious threshold. This event will not trigger an SNMP trap.)
slmrecoveredserious	SLM has risen above the serious threshold (The percentage of requests that meet this monitor's criteria has risen above the serious threshold. The percentage was below the serious threshold at the previous check.)
slmrecoveredwarn	SLM has recovered (The percentage of requests that meet this monitor's criteria has risen above the warning threshold. The percentage was below the warning threshold at the previous check.)
Event tags for object type: "sslhw"	
sslhwfail	SSL hardware support failed (SSL hardware support has stopped with an error)
sslhwrestart	SSL hardware support restarted
sslhwstart	SSL hardware support started
Event tags for object type: "vservers"	
connerror	A protocol error has occurred
connfail	A socket connection failure has occurred
dnsaddzone	The built-in DNS server has successfully added a DNS zone
dnserroraddzone	The built-in DNS server has failed to add a DNS zone
dnserrordeletezone	The built-in DNS server has failed to delete a DNS zone
dnssecexpired	DNSSEC zone contains expired signatures (DNSSEC zone contains expired signatures)

Event Tag	Description
dnssecexpires	DNSSEC zone contains signatures that are about to expire (DNSSEC zone contains signatures that are about to expire)
dnszonedelete	DNS zone has been deleted
idpcertexpired	IDP certificate expired (IDP certificate expired)
idpcerttoexpire	IDP certificate will expire within seven days (IDP certificate to expire)
logfiledeleted	A virtual server request log file was deleted (appliances only)
maxclientbufferdrop	Dropped connection, request exceeded max_client_ buffer limit (The traffic manager is still reading the request, but the amount of data read in is larger than max_client_buffer - abandon the connection)
poolpersistencemismatch	Pool uses a session persistence class that does not work with this virtual server's protocol
privkeyok	Private key now OK (hardware available) (The private key for this virtual server is now available, because some required hardware is available again.)
respcompfail	Error compressing HTTP response
responsetoolarge	Response headers from webserver too large (The response headers from the webserver were bigger than max_server_buffer; the request will be rejected with an error.)
rtspstreamnoports	No suitable ports available for streaming data connection (Consider changing the tuneables 'streaming_ portrange_low' and 'streaming_portrange_high'.)

Event Tag	Description
samlauthnrequestcompressionfailure	Failed to compress SAML authentication request
samlauthnrequestfailure	Failed to create SAML authentication request
samlcookiedecryptfailure	Failed to decrypt SAML session cookie (Failed to decrypt SAML session cookie)
samlcookieencryptfailure	Failed to encrypt cookie content (Failed to encrypt cookie content)
samlnouserinresponse	No user specified in SAML response (No user specified in SAML response)
samlrelaystatedecryptfailure	Failed to decrypt SAML RelayState (Failed to decrypt SAML RelayState)
samlrelaystateencryptfailure	Failed to encrypt SAML RelayState (Failed to encrypt SAML RelayState)
samlrelaystateinvalid	Failed to extract information from SAML RelayState
samlresponseparsefailure	SAML response parse failure
samlresponsevalidationfailure	SAML response validation failed
samlsigverificationfailure	SAML response signature verification error
sipstreamnoports	No suitable ports available for streaming data connection (Consider changing the tuneables 'streaming_ portrange_low' and 'streaming_portrange_high'.)
ssldrop	Request(s) received while SSL configuration invalid, connection closed
sslfail	One or more SSL connections from clients failed recently (One or more SSL connections from clients failed recently)
sslhandshakemsgsizelimit	SSL handshake messages have exceeded the size permitted by configuration (SSL handshake messages have exceeded the size permitted by configuration.)

Event Tag	Description
sslrehandshakemininterval	SSL re-handshake requests have exceeded the frequency permitted by configuration (SSL re-handshake requests have exceeded the frequency permitted by configuration.)
vscacertexpired	Certificate Authority certificate expired (Certificate Authority certificate expired)
vscacerttoexpire	Certificate Authority certificate will expire within seven days (Certificate Authority certificate to expire)
vscrloutofdate	CRL for a Certificate Authority is out of date
vslogwritefail	Failed to write log file for virtual server
vssslcertexpired	Public SSL certificate expired (Public SSL certificate expired)
vssslcerttoexpire	Public SSL certificate will expire within seven days (Public SSL certificate to expire)
vsstart	Virtual server started
vsstop	Virtual server stopped
Event tags for object type: "zxtms"	
versionmismatch	Configuration update refused: traffic manager version mismatch

conf/extra

The conf/extra directory contains miscellaneous user-uploaded files. These files can be used in TrafficScript code using the resource.get function. The files can be managed under the Catalogs > Extra Files > Miscellaneous Files section of the Admin Server UI or by using functions under the Conf.Extra section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/flipper

The conf/flipper directory contains configuration files for traffic IP groups. The name of a file is the name of the traffic IP group it defines. Traffic IP groups can be managed under the Services > Traffic IP Groups section of the Admin Server UI or by using functions under the TrafficIPGroups section of the SOAP API and CLI. \gui_only \regex .* \errortext no error \soap_ignore

Key	Description
enabled	If set to No, the traffic IP group will be disabled and none of the traffic IP addresses will be raised. Value type: Yes / No Default value: "Yes"
hash_srcport	Whether or not the source port should be taken into account when deciding which traffic manager should handle a request. Requires: mode is set to "multihosted" Value type: Yes / No Default value: "No"
ip_assignment_mode	Configure how traffic IPs are assigned to traffic managers in Single-Hosted mode Value type: enumeration Default value: "balanced" Permitted values: alphabetic: Alphabetical order of traffic manager hostnames balanced: Approximately balanced between traffic managers
ipaddress!*!machine	Assigns a traffic IP address to a specific traffic manager, while the traffic manager is operating correctly it will host the address. The IP address must be one from the ipaddresses list and takes the place of the * in the key name, the key value is the name of the traffic manager that should host the IP address. If this is not specified for an IP address the address is automatically assigned to a machine. Value type: string Default value: <none>
ipaddresses	The IP addresses that belong to the Traffic IP group. Value type: list Default value: <none>

Key	Description
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. Value type: Yes / No Default value: "No"
location	The location in which the Traffic IP group is based. Value type: int Default value: "0"
machines	The traffic managers that can host the traffic IP group's IP addresses. Value type: list 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. Value type: enumeration Default value: "singlehosted" Permitted values: singlehosted: Raise each address on a single machine (Single-Hosted mode) 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 ec2elastic: Use an EC2-Classic Elastic IP address. ec2vpcelastic: Use an EC2-VPC Elastic IP address. ec2vpcprivate: Use an EC2-VPC Private IP address. gceexternal: Use GCE External IP addresses.
multicast	The multicast IP address used to duplicate traffic to all traffic managers in the group. Requires: mode is set to "multihosted" Value type: string Default value: <none>
note	A note, used to describe this Traffic IP Group

Key	Description
	Value 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. Requires: mode is set to "rhi" Value type: unsigned integer 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. Requires: mode is set to "rhi" Value type: unsigned integer 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. Requires: mode is set to "rhi" Value type: unsigned integer 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. Requires: mode is set to "rhi" Value type: unsigned integer 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. Requires: mode is set to "rhi" Value type: string Default value: "ospf"

Key	Description
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. Value type: list Default value: <none>

conf/groups

Files in the conf/groups directory define the permission groups configured for administrative access to the software. The name of a file is the name of the group it defines. Permission groups can be managed under the System > Users section of the Admin Server UI. Each group will contain a list of configuration keys with names that mostly correspond to pages in the Admin Server UI. These 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 an exclamation mark (!) followed by the sub-permission name. The built-in admin group has a special permission key of all with the value full, this cannot be altered for the admin group but can be used in other group configuration files to change the default permission level for the group.

Key	Description
description	A description for the group. Value 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. Value type: unsigned integer Default value: "0"
timeout	Inactive UI sessions will timeout after this number of seconds. To disable inactivity timeouts for the group set this to 0 (zero). Value type: unsigned integer Default value: "30"

Key	Description
Permission keys by section	
Section: "Activity"	
Connections	"Connections" Permitted values: none, ro, or full
Connections!Details	"Connections > Details" Permitted values: none, ro, or full
Web_Cache	"Content Cache" Permitted values: none, ro, or full
Web_Cache!Clear	"Content Cache > Clear" Permitted values: none, ro, or full
Monitoring	"Current Activity" Permitted values: none, ro, or full
Monitoring!Edit	"Current Activity > Edit" Permitted values: none, ro, or full
Request_Logs	"Download Logs" Permitted values: none, ro, or full
Draining	"Draining Nodes" Permitted values: none, ro, or full
Statd	"Historical Activity" Permitted values: none, ro, or full
Map	"Map" Permitted values: none, ro, or full
Log_Viewer	"View Logs" Permitted values: none, ro, or full
Log_Viewer!View	"View Logs > View" Permitted values: none, ro, or full
Section: "Advanced Management"	
Appliance_Console	"Appliance Console" Permitted values: none or full

Key	Description
	Requires feature: Appliance
Custom	"Custom Configuration Sets" Permitted values: none, ro, or full
SOAP_API	"SOAP Control API" Permitted values: none or full Requires feature: SOAP
Section: "Catalogs"	
Authenticators	"Authenticators" Permitted values: none, ro, or full
Authenticators!Edit	"Authenticators > Edit" Permitted values: none, ro, or full
Bandwidth	"Bandwidth" Permitted values: none, ro, or full Requires feature: Bandwidth
Bandwidth!Edit	"Bandwidth > Edit" Permitted values: none, ro, or full Requires feature: Bandwidth
Bandwidth!Edit!CopyClass	"Bandwidth > Edit > Copy Class" Permitted values: none, ro, or full Requires feature: Bandwidth
Cloud_Credentials	"Cloud Credentials" Permitted values: none, ro, or full
Cloud_Credentials!Edit	"Cloud Credentials > Edit" Permitted values: none, ro, or full
DNS_Server	"DNS Server" Permitted values: none, ro, or full
DNS_Server!Zonefiles	"DNS Server > Zone Files" Permitted values: none, ro, or full
DNS_Server!Zones	"DNS Server > Zones" Permitted values: none, ro, or full

Key	Description
DNS_Server!Zones!Edit	"DNS Server > Zones > Edit" Permitted values: none, ro, or full
Extra_Files	"Extra Files" Permitted values: none, ro, or full
Extra_Files!Action_Programs	"Extra Files > Action Programs" Permitted values: none, ro, or full
Extra_Files!Miscellaneous_Files	"Extra Files > Miscellaneous" Permitted values: none, ro, or full
Extra_Files!ExternProgMonitors	"Extra Files > Monitor Programs" Permitted values: none, ro, or full
GLB_Services	"GLB Services" Permitted values: none, ro, or full
GLB_Services!Edit	"GLB Services > Edit" Permitted values: none, ro, or full
GLB_Services!Edit!DNS_Settings	"GLB Services > Edit > DNS Settings" Permitted values: none, ro, or full
GLB_Services!Edit!DNSSEC	"GLB Services > Edit > DNSSEC" Permitted values: none, ro, or full
GLB_Services!Edit!Load_Balancing	"GLB Services > Edit > Load Balancing" Permitted values: none, ro, or full
GLB_Services!Edit!Locations	"GLB Services > Edit > Locations" Permitted values: none, ro, or full
GLB_Services!Edit!Request_Logging	"GLB Services > Edit > Request Logging" Permitted values: none, ro, or full
GLB_Services!Edit!Rules	"GLB Services > Edit > Rules" Permitted values: none, ro, or full
Java	"Java" Permitted values: none, ro, or full
Java!Edit	"Java > Edit"

Key	Description
	Permitted values: none, ro, or full
Kerberos	"Kerberos" Permitted values: none, ro, or full
Kerberos!krb5confs	"Kerberos > Kerberos Configuration Files" Permitted values: none, ro, or full
Kerberos!Kerberos_Keytabs	"Kerberos > Kerberos Keytabs" Permitted values: none, ro, or full
Kerberos!Kerberos_Principals	"Kerberos > Kerberos Principals" Permitted values: none, ro, or full
Kerberos!Kerberos_Principals!Edit	"Kerberos > Kerberos Principals > Edit" Permitted values: none, ro, or full
Locations	"Locations" Permitted values: none, ro, or full
Locations!Edit	"Locations > Edit" Permitted values: none, ro, or full
Monitors	"Monitors" Permitted values: none, ro, or full
Monitors!Edit	"Monitors > Edit" Permitted values: none, ro, or full
Monitors!Edit!CopyMonitor	"Monitors > Edit > Copy Monitor" Permitted values: none, ro, or full
Catalog	"Overview" Permitted values: none, ro, or full
Persistence	"Persistence" Permitted values: none, ro, or full
Persistence!Edit	"Persistence > Edit" Permitted values: none, ro, or full
Persistence!Edit!CopyClass	"Persistence > Edit > Copy Class" Permitted values: none, ro, or full

Key	Description
Service_Protection	"Protection" Permitted values: none, ro, or full
Service_Protection!Edit	"Protection > Edit" Permitted values: none, ro, or full
Service_Protection!Edit!CopyClass	"Protection > Edit > Copy Class" Permitted values: none, ro, or full
Rate	"Rate" Permitted values: none, ro, or full Requires feature: Rules
Rate!Edit	"Rate > Edit" Permitted values: none, ro, or full Requires feature: Rules
Rules	"Rules" Permitted values: none, ro, or full Requires feature: Rules
Rules!GEdit	"Rules > Build" Permitted values: none, ro, or full Requires feature: Rules
Rules!GEdit!AddAction	"Rules > Build > Add Action" Permitted values: none, ro, or full Requires feature: Rules
Rules!GEdit!AddCondition	"Rules > Build > Add Condition" Permitted values: none, ro, or full Requires feature: Rules
Rules!GEdit!Convert	"Rules > Build > Convert" Permitted values: none, ro, or full Requires feature: Rules
Rules!Edit	"Rules > Edit" Permitted values: none, ro, or full Requires feature: Rules
Rules!Edit!CheckSyntax	"Rules > Edit > Check Syntax"

Key	Description
	Permitted values: none, ro, or full Requires feature: Rules
Rules!Edit!SaveAs	"Rules > Edit > Save As" Permitted values: none, ro, or full Requires feature: Rules
SAML	"SAML" Permitted values: none, ro, or full
SAML!Trusted_Identity_Providers	"SAML > Trusted Identity Providers" Permitted values: none, ro, or full
SAML!Trusted_Identity_Providers!Edit	"SAML > Trusted Identity Providers > Edit" Permitted values: none, ro, or full
Service_Discovery	"Service Discovery" Permitted values: none, ro, or full
Service_Discovery!Builtin_Plugins	"Service Discovery > Built-in Plugins" Permitted values: none, ro, or full
Service_Discovery!User_Plugins	"Service Discovery > User Plugins" Permitted values: none, ro, or full
SLM	"SLM" Permitted values: none, ro, or full Requires feature: SLM
SLM!Edit	"SLM > Edit" Permitted values: none, ro, or full Requires feature: SLM
SLM!Edit!CopyClass	"SLM > Edit > Copy Class" Permitted values: none, ro, or full Requires feature: SLM
SSL	"SSL" Permitted values: none, ro, or full
SSL!AdminCAs	"SSL > Admin CAs and CRLs" Permitted values: none, ro, or full

Key	Description
SSL!AdminCAs!Edit	"SSL > Admin CAs and CRLs > Edit" Permitted values: none, ro, or full
SSL!AdminCAs!Import	"SSL > Admin CAs and CRLs > Import" Permitted values: none, ro, or full
SSL!CAs	"SSL > CAs and CRLs" Permitted values: none, ro, or full
SSL!CAs!Edit	"SSL > CAs and CRLs > Edit" Permitted values: none, ro, or full
SSL!CAs!Import	"SSL > CAs and CRLs > Import" Permitted values: none, ro, or full
SSL!Client_Certs	"SSL > Client Certs" Permitted values: none, ro, or full
SSL!Client_Certs!Edit	"SSL > Client Certs > Edit" Permitted values: none, ro, or full
SSL!Client_Certs!Edit!Chain	"SSL > Client Certs > Edit > Chain" Permitted values: none, ro, or full
SSL!Client_Certs!Edit!CopyCert	"SSL > Client Certs > Edit > Copy Certificate" Permitted values: none, ro, or full
SSL!Client_Certs!Edit!Sign	"SSL > Client Certs > Edit > Sign" Permitted values: none, ro, or full
SSL!Client_Certs!Import	"SSL > Client Certs > Import" Permitted values: none, ro, or full
SSL!Client_Certs!New	"SSL > Client Certs > New" Permitted values: none, ro, or full
SSL!DNSSEC_Keys	"SSL > DNSSEC Keys" Permitted values: none, ro, or full
SSL!SSL_Certs	"SSL > Server Certs" Permitted values: none, ro, or full
SSL!SSL_Certs!Edit	"SSL > Server Certs > Edit"

Key	Description
	Permitted values: none, ro, or full
SSL!SSL_Certs!Edit!Chain	"SSL > Server Certs > Edit > Chain" Permitted values: none, ro, or full
SSL!SSL_Certs!Edit!CopyCert	"SSL > Server Certs > Edit > Copy Certificate" Permitted values: none, ro, or full
SSL!SSL_Certs!Edit!Sign	"SSL > Server Certs > Edit > Sign" Permitted values: none, ro, or full
SSL!SSL_Certs!Import	"SSL > Server Certs > Import" Permitted values: none, ro, or full
SSL!SSL_Certs!New	"SSL > Server Certs > New" Permitted values: none, ro, or full
SSL!Ticket_Keys	"SSL > SSL ticket keys" Permitted values: none, ro, or full
Aptimizer	"Web Accelerator" Permitted values: none, ro, or full
Aptimizer!URL_Sets	"Web Accelerator > Application Scopes" Permitted values: none, ro, or full
Aptimizer!URL_Sets!Edit	"Web Accelerator > Application Scopes > Edit" Permitted values: none, ro, or full
Aptimizer!Aptimizer_Profiles	"Web Accelerator > Web Accelerator Profiles" Permitted values: none, ro, or full
Aptimizer!Aptimizer_Profiles!Edit	"Web Accelerator > Web Accelerator Profiles > Edit" Permitted values: none, ro, or full
Section: "Configuring"	
Config_Summary	"Config Summary" Permitted values: none, ro, or full
Config_Summary!Export	"Config Summary > Export configuration" Permitted values: none, ro, or full

Key	Description
Config_Summary!Export_Secrets	"Config Summary > Export secrets" Permitted values: none, ro, or full
Pools	"Pools" Permitted values: none, ro, or full
Pools!Edit	"Pools > Edit" Permitted values: none, ro, or full
Pools!Edit!Bandwidth	"Pools > Edit > Bandwidth" Permitted values: none, ro, or full Requires feature: Bandwidth
Pools!Edit!DNSAutoscaling	"Pools > Edit > DNS-derived autoscaling" Permitted values: none, ro, or full
Pools!Edit!IP_Transparency	"Pools > Edit > IP Transparency" Permitted values: none, ro, or full
Pools!Edit!Kerberos_Protocol_Transition	"Pools > Edit > Kerberos Protocol Transition" Permitted values: none, ro, or full
Pools!Edit!Load_Balancing	"Pools > Edit > Load Balancing" Permitted values: none, ro, or full
Pools!Edit!Monitors	"Pools > Edit > Monitors" Permitted values: none, ro, or full
Pools!Edit!Protocol_Settings	"Pools > Edit > Protocol Settings" Permitted values: none, ro, or full
Pools!Edit!Service_Discovery	"Pools > Edit > Service Discovery" Permitted values: none, ro, or full
Pools!Edit!Persistence	"Pools > Edit > Session Persistence" Permitted values: none, ro, or full
Pools!Edit!SSL	"Pools > Edit > SSL" Permitted values: none, ro, or full
Pools!Edit!Autoscaling	"Pools > Edit > vTM Autoscaler" Permitted values: none, ro, or full

Key	Description
Traffic_IP_Groups	"Traffic IP Groups" Permitted values: none, ro, or full
Traffic_IP_Groups!Edit	"Traffic IP Groups > Edit" Permitted values: none, ro, or full
Traffic_IP_Groups!Networking	"Traffic IP Groups > Networking" Permitted values: none, ro, or full
Virtual_Servers	"Virtual Servers" Permitted values: none, ro, or full
Virtual_Servers!Edit	"Virtual Servers > Edit" Permitted values: none, ro, or full
Virtual_Servers!Edit!Authentication	"Virtual Servers > Edit > Authentication" Permitted values: none, ro, or full
Virtual_Servers!Edit!Classes	"Virtual Servers > Edit > Classes" Permitted values: none, ro, or full
Virtual_Servers!Edit!Request_Tracing	"Virtual Servers > Edit > Connection Analytics" Permitted values: none, ro, or full
Virtual_Servers!Edit!Content_Caching	"Virtual Servers > Edit > Content Caching" Permitted values: none, ro, or full
Virtual_Servers!Edit!Content_Compression	"Virtual Servers > Edit > Content Compression" Permitted values: none, ro, or full
Virtual_Servers!Edit!DNS_Server	"Virtual Servers > Edit > DNS Server" Permitted values: none, ro, or full
Virtual_Servers!Edit!Error_Logging	"Virtual Servers > Edit > Error Logging" Permitted values: none, ro, or full
Virtual_Servers!Edit!Kerberos_Protocol_Transition	"Virtual Servers > Edit > Kerberos Protocol Transition" Permitted values: none, ro, or full
Virtual_Servers!Edit!Protocol_Settings	"Virtual Servers > Edit > Protocol Settings" Permitted values: none, ro, or full

Key	Description
Virtual_Servers!Edit!Request_Logging	"Virtual Servers > Edit > Request Logging" Permitted values: none, ro, or full
Virtual_Servers!Edit!Rules	"Virtual Servers > Edit > Rules" Permitted values: none, ro, or full Requires feature: Rules
Virtual_Servers!Edit!Rules!EnableDisable	"Virtual Servers > Edit > Rules > Enable / Disable" Permitted values: none, ro, or full Requires feature: Rules
Virtual_Servers!Edit!Rules!Move	"Virtual Servers > Edit > Rules > Move" Permitted values: none, ro, or full Requires feature: Rules
Virtual_Servers!Edit!Rules!OnceEvery	"Virtual Servers > Edit > Rules > Once / Every" Permitted values: none, ro, or full Requires feature: Rules
Virtual_Servers!Edit!Rules!Remove	"Virtual Servers > Edit > Rules > Remove" Permitted values: none, ro, or full Requires feature: Rules
Virtual_Servers!Edit!GLB_Services	"Virtual Servers > Edit > Service" Permitted values: none, ro, or full
Virtual_Servers!Edit!SSL_Decryption	"Virtual Servers > Edit > SSL Decryption" Permitted values: none, ro, or full
Virtual_Servers!Edit!Aptimizer_Settings	"Virtual Servers > Edit > Web Accelerator" Permitted values: none, ro, or full
Section: "Diagnosing"	
Audit_Log	"Audit Log" Permitted values: none, ro, or full
Audit_Log!Audit_Archive	"Audit Log > Audit Archive" Permitted values: none, ro, or full
Diagnose	"Cluster Diagnosis" Permitted values: none, ro, or full

Key	Description
Diagnose!Replicate	"Cluster Diagnosis > Replicate" Permitted values: none, ro, or full
Event_Log	"Event Log" Permitted values: none, ro, or full
Event_Log!Clear	"Event Log > Clear" Permitted values: none, ro, or full
Event_Log!Event_Archive	"Event Log > Event Archive" Permitted values: none, ro, or full
Hardware	"Hardware" Permitted values: none, ro, or full Requires feature: Appliance
Routing	"Routing" Permitted values: none, ro, or full
Support_Files	"Support Files" Permitted values: none, ro, or full
Support	"Technical Support" Permitted values: none, ro, or full
Support!TSR	"Technical Support > TSR" Permitted values: none, ro, or full
Section: "Main Pages"	
Help	"Help" Permitted values: none, ro, or full
MainIndex	"Main Index" Permitted values: none, ro, or full
Reboot	"Reboot" Permitted values: none, ro, or full
Restart	"Restart" Permitted values: none, ro, or full
Shutdown	"Shutdown"

Key	Description
	Permitted values: none, ro, or full
Section: "System"	
Alerting	"Alerting" Permitted values: none, ro, or full
Alerting!Actions	"Alerting > Actions" Permitted values: none, ro, or full
Alerting!Actions!Edit	"Alerting > Actions > Edit" Permitted values: none, ro, or full
Alerting!Event_Types	"Alerting > Event Types" Permitted values: none, ro, or full
Alerting!Event_Types!Edit	"Alerting > Event Types > Edit" Permitted values: none, ro, or full
Analytics_Export	"Analytics Export" Permitted values: none, ro, or full
Analytics_Export!Log_Files	"Analytics Export > Log Files" Permitted values: none, ro, or full
Analytics_Export!Log_Files!Edit	"Analytics Export > Log Files > Edit" Permitted values: none, ro, or full
Backup	"Backups" Permitted values: none, ro, or full
Backup!Config_Difference	"Backups > Compare" Permitted values: none, ro, or full
Backup!Edit	"Backups > Edit" Permitted values: none, ro, or full
Backup!Partial	"Backups > Partial" Permitted values: none, ro, or full
Fault_Tolerance	"Fault Tolerance" Permitted values: none, ro, or full
Fault_Tolerance!BGP_Neighbors	"Fault Tolerance > BGP Neighbors"

Key	Description
	Permitted values: none, ro, or full
Fault_Tolerance!BGP_Neighbors!Edit	"Fault Tolerance > BGP Neighbors > Edit" Permitted values: none, ro, or full
Global_Settings	"Global Settings" Permitted values: none, ro, or full
Global_Settings!Restore_Defaults	"Global Settings > Restore Defaults" Permitted values: none, ro, or full
License_Keys	"Licenses" Permitted values: none, ro, or full
License_Keys!InstallNew	"Licenses > Install New" Permitted values: none, ro, or full
License_Keys!Remove	"Licenses > Remove" Permitted values: none, ro, or full
License_Keys!Register	"Licenses > Self-register" Permitted values: none, ro, or full
Networking	"Networking" Permitted values: none, ro, or full Requires feature: Appliance
Networking!NAT	"Networking > NAT" Permitted values: none, ro, or full Requires feature: Appliance
Security	"Security" Permitted values: none, ro, or full
SNMP	"SNMP" Permitted values: none, ro, or full
Sysctl	"Sysctl" Permitted values: none, ro, or full Requires feature: Appliance
DateTime	"Time" Permitted values: none, ro, or full

Key	Description
	Requires feature: Appliance
Traffic_Managers	"Traffic Managers" Permitted values: none, ro, or full
Traffic_Managers!AddRemove	"Traffic Managers > AddRemove" Permitted values: none, ro, or full
Traffic_Managers!Rollback	"Traffic Managers > Rollback" Permitted values: none, ro, or full
Traffic_Managers!Upgrade	"Traffic Managers > Upgrade" Permitted values: none, ro, or full
Access_Management	"Users" Permitted values: none, ro, or full
Access_ Management!AuthenticationMethods	"Users > Authenticators" Permitted values: none, ro, or full
Access_ Management!AuthenticationMethods!Edit	"Users > Authenticators > Edit" Permitted values: none, ro, or full
Access_Management!Groups	"Users > Groups" Permitted values: none, ro, or full
Access_Management!Groups!Edit	"Users > Groups > Edit" Permitted values: none, ro, or full
Access_Management!LocalUsers	"Users > Local" Permitted values: none, ro, or full
Access_Management!LocalUsers!Edit	"Users > Local > Edit" Permitted values: none, ro, or full
Access_ Management!LocalUsers!EditOtherUsers	"Users > Local > Other Users" Permitted values: none, ro, or full
Access_ Management!LocalUsers!PasswordPolicy	"Users > Local > Password Policy" Permitted values: none, ro, or full
Access_Management!Suspended_Users	"Users > Suspended Users" Permitted values: none, ro, or full

Key	Description
AFM	"Web Application Firewall" Permitted values: none, ro, or full
AFM!Admin	"Web Application Firewall > Administration" Permitted values: none or full
Section: "Wizards"	
Wizard!Backup	"Wizard > Backup my configuration" Permitted values: none, ro, or full
Wizard!AzureKeyVault	"Wizard > Connect to Microsoft Azure Key Vault" Permitted values: none, ro, or full
Wizard!DisableNode	"Wizard > Disable a node" Permitted values: none, ro, or full
Wizard!DrainNode	"Wizard > Drain a node" Permitted values: none, ro, or full
Wizard!EnableRule	"Wizard > Enable/Disable a rule" Permitted values: none, ro, or full Requires feature: Rules
Wizard!FreeDiskSpace	"Wizard > Free up some disk space" Permitted values: none, ro, or full
Wizard!ClusterJoin	"Wizard > Join a cluster" Permitted values: none, ro, or full
Wizard!PulseConnectSecure	"Wizard > Load-balance Pulse Connect Secure" Permitted values: none, ro, or full
Wizard!NewService	"Wizard > Manage a new service" Permitted values: none, ro, or full
Wizard!OptimalGatewaySelection	"Wizard > Optimal Gateway Selection" Permitted values: none, ro, or full
Wizard!AptimizeService	"Wizard > Optimize a web application" Permitted values: none, ro, or full
Wizard!ReactivateNode	"Wizard > Reactivate a node"

Key	Description
	Permitted values: none, ro, or full
Wizard!RemoveNode	"Wizard > Remove a node" Permitted values: none, ro, or full
Wizard!Restore	"Wizard > Restore from a backup" Permitted values: none, ro, or full
Wizard!SSLDecryptService	"Wizard > SSL Decrypt a service" Permitted values: none, ro, or full

conf/jars

The conf/jars directory contains files for TrafficScript Java Extensions. This includes items such as jar and class files that provide servlets and their dependencies, as well as data files for general use by Java Extensions. The files in this directory can be managed using the Catalogs > Java section of the Admin Server UI and the Catalog.JavaExtension section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/kerberos/keytabs

The conf/kerberos/keytabs directory contains Kerberos keytab files that contain credentials for Kerberos principals the traffic manager will use to perform Kerberos operations. Kerberos keytabs can be managed under the Catalogs > Kerberos > Kerberos Keytabs section of the administrative interface or by using functions under the Catalog.Kerberos.Keytabs section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/kerberos/krb5confs

The conf/kerberos/krb5confs directory contains krb5.conf configuration files for Kerberos principals the traffic manager can use to perform Kerberos operations. These are provided to provide raw control of the kerberos library should it be neccessary to enable operations the standard configuration cannot achieve. krb5.conf files can be managed under the Catalogs > Kerberos > Kerberos Configuration Files section of the administrative interface or by using functions under the Catalog.Kerberos.KRB5confs section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/kerberos/principals

The conf/kerberos/principals directory contains configuration files for Kerberos principals the traffic manager can use to perform Kerberos operations. The name of a file is the name of the Kerberos principal it defines. Kerberos principals can be configured under the Catalogs > Kerberos Principals section of the administrative interface or by using functions under the Catalog.Kerberos.Principals section of the SOAP API and CLI.

Key	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. Requires: krb5conf is set to "" Value type: list Default value: <none>
keytab	The name of the Kerberos keytab file containing suitable credentials to authenticate as the specified Kerberos principal. Value type: string Default value: <none>
krb5conf	The name of an optional Kerberos configuration file (krb5.conf). Value type: string

Key	Description
	Default value: <none>
realm	The Kerberos realm where the principal belongs. Value type: string Default value: <none>
service	The service name part of the Kerberos principal name the traffic manager should use to authenticate itself. Value type: string Default value: <none>

conf/licensekeys

The conf/licensekeys directory is where license key files for the software are stored. License keys can be managed under the System > Licenses section of the Admin Server UI or by using the System.LicenseKeys section of the SOAP API or CLI.

Key	Description
There are no items to display for this configuration type.	

conf/locations

Configuration for locations. Must be higher priority than global.cfg.

Key	Description
based_on	Used by the UI to store where we got the lat/long coords from (a preset value, user entered etc.). Value type: string Default value: "none"
id	The identifier of this location. Value type: unsigned integer Default value: <none>

Key	Description
latitude	The latitude of this location. Value type: double Default value: "0.0"
longitude	The longitude of this location. Value type: double Default value: "0.0"
note	A note, used to describe this location. Value type: string Default value: <none>
type	Does this location contain traffic managers and configuration or is it a recipient of GLB requests? Value type: enumeration Default value: "config" Permitted values: config: Configuration glb: GLB

conf/locations.cfg

The locations.cfg file contains custom geolocation data. This is a text file that must be managed manually, comments in the file describe the data format.

Key	Description
There are no items to display for this configuration type.	

conf/log_export

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

Key	Description
appliance_only	Whether entries from the specified log files should be exported only from appliances. Value type: Yes / No Default value: "No"
built_in	Whether this configuration is built-in. Editing and deletion of built-in configurations is restricted. Value type: Yes / No Default value: "No"
enabled	Export entries from the log files included in this category. Value type: Yes / No Default value: "No"
files	The set of files to export as part of this category, specified as a list of glob patterns. Value type: list Default value: <none>
history	How much historic log activity should be exported. Value type: enumeration Default value: "none" Permitted values: none: Do not export any historic entries all: Export all 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. Value type: unsigned integer Default value: "10"
metadata!*	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. Value type: string Default value: <none>
note	A description of this category of log files.

Key	Description
	Value type: string Default value: <none>

conf/monitors

The conf/monitors directory contains configuration files for backend node monitors. The name of a file is the name of the monitor it defines. Monitors can be configured under the Catalogs > Monitors section of the Admin Server UI or by using functions under the Catalog.Monitor section of the SOAP API and CLI.

Key	Description
back_off	Should the monitor slowly increase the delay after it has failed? Value type: Yes / No Default value: "Yes"
delay	The minimum time between calls to a monitor. Value type: seconds 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. Value type: unsigned integer Default value: "3"
health_only	Should this monitor only report health (ignore load)? Value type: Yes / No Default value: "No"
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. Requires: scope is set to "poolwide" Value type: string Default value: <none>
max_response_len	The maximum amount of data to read back from a server, use 0 for unlimited.

Key	Description
	Value type: bytes Default value: "2048"
note	A description of the monitor. Value type: string Default value: <none>
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. Value type: enumeration 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. Value type: seconds Default value: "3"
type	The internal monitor implementation of this monitor. Value type: enumeration Default value: "ping" Permitted values: ping: Ping monitor connect: TCP Connect monitor http: HTTP monitor tcp_transaction: TCP transaction monitor program: External program monitor sip: SIP monitor rtsp: RTSP monitor
udp_accept_all	If this monitor uses UDP, should it accept responses from any IP and port? Value type: Yes / No Default value: "No"
use_ssl	Whether or not the monitor should connect using SSL.

Key	Description
	Requires: can_use_ssl is set to "Yes" Value type: Yes / No Default value: "No"
verbose	Whether or not the monitor should emit verbose logging. This is useful for diagnosing problems. Value type: Yes / No Default value: "No"
Additional keys used when type is "http"	
authentication	The HTTP basic-auth <user>:<password> to use for the test HTTP request. Requires: type is set to "http" Value 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). Requires: type is set to "http" Value type: string Default value: <none>
host_header	The host header to use in the test HTTP request. Requires: type is set to "http" Value type: string Default value: <none>
path	The path to use in the test HTTP request. This must be a string beginning with a / (forward slash). Requires: type is set to "http" Value 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). Requires: type is set to "http" Value type: string Default value: "^[234][0-9][0-9]$"

Key	Description
Additional keys used when type is "program"	
arg!*	The arguments that will be passed to the program. For example, to specify the argument --foo=bar as part of the program's command-line you set the key arg!foo to the value bar. Requires: type is set to "program" Value type: string Default value: <none>
describe!*	A description for the argument specified in place of the * character. For example, to describe the argument in the example for arg!* you could specify the description as the value for the key describe!foo. Requires: type is set to "program" Value type: string Default value: <none>
program	The program to run. This must be an executable file, either within the conf/scripts directory or specified as an absolute path to some other location on the filesystem. Requires: type is set to "program" Value type: string Default value: <none>
Additional keys used when type is "rtsp"	
rtsp_body_regex	The regular expression that the RTSP response body must match. Requires: type is set to "rtsp" Value type: string Default value: <none>
rtsp_path	The path to use in the RTSP request (some servers will return 500 Internal Server Error unless this is a valid media file). Requires: type is set to "rtsp" Value type: string Default value: "/"
rtsp_status_regex	The regular expression that the RTSP response status code must match. Requires: type is set to "rtsp" Value type: string

Key	Description
	Default value: "^[234][0-9][0-9]$"
Additional keys used when type is "sip"	
sip_body_regex	The regular expression that the SIP response body must match. Requires: type is set to "sip" Value type: string Default value: <none>
sip_status_regex	The regular expression that the SIP response status code must match. Requires: type is set to "sip" Value type: string Default value: "^[234][0-9][0-9]$"
sip_transport	Which transport protocol the SIP monitor will use to query the server. Requires: type is set to "sip" Value type: enumeration Default value: "udp" Permitted values: udp: UDP tcp: TCP
Additional keys used when type is "tcp_transaction"	
close_string	An optional string to write to the server before closing the connection. Requires: type is set to "tcp_transaction" Value type: string Default value: <none>
response_regex	A regular expression to match against the response from the server. Requires: type is set to "tcp_transaction" Value type: string Default value: ".+"
write_string	The string to write down the TCP connection. Requires: type is set to "tcp_transaction" Value type: string Default value: <none>

conf/persistence

The conf/persistence directory contains configuration files for persistence classes. The name of a file is the name of the persistence class it defines. Persistence classes can be configured under the Catalogs > Persistence section of the Admin Server UI or by using functions under the Catalog.Persistence section of the SOAP API and CLI.

Key	Description
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.) Value type: Yes / No Default value: "Yes"
failuremode	The action the pool should take if the session data is invalid or it cannot contact the node specified by the session. Value type: enumeration Default value: "newnode" Permitted values: newnode: Choose a new node to use url: Redirect the user to a given URL close: Close the connection (using error_file on Virtual Servers > Edit > Protocol Settings)
note	A description of the session persistence class. Value type: string Default value: <none>
type	The type of session persistence to use. Value type: enumeration Default value: "ip" Permitted values: ip: IP-based persistence universal: Universal session persistence named: Named Node session persistence sardine: Transparent session affinity kipper: Monitor application cookies j2ee: J2EE session persistence asp: ASP and ASP.NET session persistence

Key	Description
	x-zeus: X-Zeus-Backend cookies ssl: SSL Session ID persistence
url	The redirect URL to send clients to if the session persistence is configured to redirect users when a node dies. Requires: failuremode is set to "url" (case insensitive) Value type: string Default value: <none>
Additional keys used when type is "ip"	
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. Requires: type is set to "ip" Value type: int Default value: "0"
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. Requires: type is set to "ip" Value type: int Default value: "0"
Additional keys used when type is "kipper"	
cookie	The cookie name to use for tracking session persistence. Requires: type is set to "kipper" Value type: string Default value: <none>
Additional keys used when type is "sardine"	
transparent_always_set_cookie	Whether or not the cookie should be inserted in every response sent to the client when using transparent session affinity. If set to No then the cookie is inserted only if the corresponding request did not already contain a matching cookie.

Key	Description
	Requires: type is set to "sardine" Value type: Yes / No Default value: "No"
transparent_directives	The cookie directives to include in the cookie sent when using transparent session affinity. If more than one directive is included, the semi-colon separator between them must be included in this string. The semi-colon separator between the cookie value and the first directive should not be included in this string. Requires: type is set to "sardine" Value type: string Default value: <none>

conf/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.

Key	Description
autoscale!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. autoscale!enabled' is set to "yes"' Value type: seconds Default value: "0"
autoscale!cloudcredentials	The Cloud Credentials object containing authentication credentials to use in cloud API calls. autoscale!enabled' is set to "yes"' Value type: string Default value: <none>
autoscale!cluster	The ESX host or ESX cluster name to put the new virtual machine instances on.

Key	Description
	autoscale!enabled' is set to "yes"' Value type: string Default value: <none>
autoscale!datacenter	The name of the logical datacenter on the vCenter server. Virtual machines will be scaled up and down under the datacenter root folder. autoscale!enabled' is set to "yes"' Value type: string Default value: <none>
autoscale!datastore	The name of the datastore to be used by the newly created virtual machine. autoscale!enabled' is set to "yes"' Value type: string Default value: <none>
autoscale!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. Value type: Yes / No Default value: "No"
autoscale!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. autoscale!enabled' is set to "yes"' Value type: Yes / No Default value: "Yes"
autoscale!extraargs	Any extra arguments to the autoscaling API. Each argument can be separated by comma. E.g in case of EC2, it can take extra parameters to the Amazon's RunInstance API say DisableApiTermination=false,Placement.Tenancy=default. autoscale!enabled' is set to "yes"'

Key	Description
	Value type: string Default value: <none>
autoscale!hysteresis	The time period in seconds for which a change condition must persist before the change is actually instigated. Value type: unsigned integer Default value: "20"
autoscale!imageid	The identifier for the image of the instances to create. autoscale!enabled' is set to "yes"' Value type: string Default value: <none>
autoscale!ipstouse	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. autoscale!enabled' is set to "yes"' Value type: enumeration Default value: "publicips" Permitted values: publicips: Public IP addresses privateips: Private IP addresses
autoscale!lastnode_idletime	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. Value type: unsigned integer Default value: "3600"
autoscale!max_nodes	The maximum number of nodes in this autoscaled pool. autoscale!enabled' is set to "yes"' Value type: unsigned integer Default value: "4"
autoscale!min_nodes	The minimum number of nodes in this autoscaled pool. autoscale!enabled' is set to "yes"' Value type: unsigned integer Default value: "1"

Key	Description
autoscale!name	The beginning of the name of nodes in the cloud that are part of this autoscaled pool. autoscale!enabled' is set to "yes"' Value type: string Default value: <none>
autoscale!port	The port number to use for each node in this autoscaled pool. autoscale!enabled' is set to "yes"' Value type: unsigned integer Default value: "80"
autoscale!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. autoscale!enabled' is set to "yes"' Value type: unsigned integer Default value: "180"
autoscale!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. autoscale!enabled' is set to "yes"' Value type: unsigned integer Default value: "1000"
autoscale!scaledown_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. autoscale!enabled' is set to "yes"' Value type: unsigned integer Default value: "95"
autoscale!scaleup_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. autoscale!enabled' is set to "yes"' Value type: unsigned integer Default value: "40"

Key	Description
autoscale!securitygroupids	List of security group IDs to associate to the new EC2 instance. autoscale!enabled' is set to "yes"' Value type: list Default value: <none>
autoscale!sizeid	The identifier for the size of the instances to create. autoscale!enabled' is set to "yes"' Value type: string Default value: <none>
autoscale!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. autoscale!enabled' is set to "yes"' Value type: list Default value: <none>
bandwidth_class	The Bandwidth Management Class this pool uses, if any. Value type: string Default value: <none>
disabled	A list of nodes in the pool that are in the 'disabled' state. Value type: list Default value: <none>
dns_autoscale!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. Value type: Yes / No Default value: "No"
dns_autoscale!hostnames	A list of hostnames which will be used for DNS-derived autoscaling dns_autoscale!enabled' is set to "yes"' Value type: list Default value: <none>
dns_autoscale!port	The port number to use for each node when using DNS-derived autoscaling Value type: unsigned integer

Key	Description
	Default value: "80"
draining	A list of nodes in the pool that are in the 'draining' state. Value type: list Default value: <none>
failpool	If all of the nodes in this pool have failed, then requests can be diverted to another pool. Value type: string Default value: <none>
ftp_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. Value type: Yes / No Default value: "No"
keepalive	Whether or not the pool should maintain HTTP keepalive connections to the nodes. Value type: Yes / No Default value: "Yes"
keepalive!non_idempotent	Whether or not the pool should maintain HTTP keepalive connections to the nodes for non-idempotent requests. Value type: Yes / No Default value: "No"
kerberos_protocol_transition!principal	The Kerberos principal the traffic manager should use when performing Kerberos Protocol Transition. Value type: string Default value: <none>
kerberos_protocol_transition!target	The Kerberos principal name of the service this pool targets. Value type: string Default value: <none>
load_balancing!algorithm	The load balancing algorithm that this pool uses. Value type: enumeration Default value: "roundrobin" Permitted values:

Key	Description
	roundrobin: Round Robin wroundrobin: Weighted Round Robin cells: Perceptive connections: Least Connections wconnections: Weighted Least Connections responsetimes: Fastest Response Time random: Random Node
load_balancing!weighting!*	Weights for each node in the pool. The actual values in isolation do not matter, as long as they are valid integers, the per-node weightings are calculated on their relative values between the nodes. The key should be specified once-per-node with the node identifier (<ip>:<port>) replacing the * in each instance. Value type: int Default value: <none>
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. Value type: seconds Default value: "4"
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. Value type: unsigned integer Default value: "0"
max_connections_pernode	The maximum number of concurrent connections allowed to each back-end node in this pool per machine. A value of 0 means unlimited connections. Value type: unsigned integer Default value: "0"
max_idle_connections_ pernode	The maximum number of unused HTTP keepalive connections that should be maintained to an individual node. Zero signifies no limit. Value type: unsigned integer Default value: "50"

Key	Description
max_queue_size	The maximum number of connections that can be queued due to connections limits. A value of 0 means unlimited queue size. Value type: unsigned integer Default value: "0"
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). Value type: seconds Default value: "30"
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. Value type: unsigned integer Default value: "2"
max_transactions_per_node	The maximum number of concurrent transactions allowed to each back-end node in this pool per machine. A value of 0 means unlimited transactions. Idle connections kept alive for reuse do not count against this limit. A transaction begins by allocating a connection for sending the request, and ends (for the purposes of queuing) after a complete response has been received from the node. Value type: unsigned integer Default value: "0"
monitors	A list of monitors assigned to this pool. Value type: list 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. Value type: Yes / No Default value: "No"
node_connclose	Close all connections to a node once we detect that it has failed.

Key	Description
	Value type: Yes / No Default value: "No"
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. Value type: unsigned integer Default value: "3"
node_delete_behavior	Specify the deletion behavior for nodes in this pool. Value type: enumeration Default value: "immediate" Permitted values: immediate: All connections to the node are closed immediately. drain: Allow existing connections to the node to finish before deletion.
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. Value type: seconds Default value: "0"
node_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. Value type: seconds Default value: "60"
node_so_nagle	Whether or not Nagle's algorithm should be used for TCP connections to the back-end nodes. Value type: Yes / No Default value: "Yes"
nodes	A list of nodes in this pool. A node should be specified as a <ip>:<port> pair. Value type: list Default value: <none>
note	A description of the pool.

Key	Description
	Value 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. Value type: Yes / No Default value: "Yes"
persistence	The default Session Persistence class this pool uses, if any. Value type: string Default value: <none>
priority!enabled	Enable priority lists. Value type: Yes / No Default value: "No"
priority!nodes	Minimum number of highest-priority active nodes. Value type: unsigned integer Default value: "1"
priority!values	A list of node priorities, higher values signify higher priority. Priorities are specified using the format <ip>:<port>:<priority>, if a priority is not specified for a node it is assumed to be 1. Value type: list Default value: <none>
queue_timeout	The maximum time to keep a connection queued in seconds. Value type: seconds Default value: "10"
service_discovery!enabled	Are the nodes of this pool determined by a Service Discovery plugin? If yes, nodes will be automatically added and removed from the pool by the traffic manager. Value type: Yes / No

Key	Description
	Default value: "No"
service_discovery!interval	The minimum time before rerunning the Service Discovery plugin Value type: unsigned integer Default value: "10"
service_discovery!plugin	The plugin script a Service Discovery autoscaled pool should use to retrieve the list of nodes. Value type: string Default value: <none>
service_discovery!plugin_args	The arguments for the script specified in "service_ discovery!plugin", e.g. a common instance tag, or name of a managed group of cloud instances. Value type: string Default value: <none>
service_discovery!timeout	The maximum time a plugin should be allowed to run before timing out. Set to 0 for no timeout. Value type: unsigned integer Default value: "0"
smtp!send_starttls	If we are encrypting traffic for an SMTP connection, should we upgrade to SSL using STARTTLS. Value type: Yes / No Default value: "Yes"
ssl_cipher_suites	The SSL/TLS cipher suites to allow for connections to a back-end node. Leaving this empty will make the pool use the globally configured cipher suites, see configuration key ssl!cipher_suites in the Global Settings section of the System tab. See there for how to specify SSL/TLS cipher suites. Value type: string Default value: <none>
ssl_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. Value type: Yes / No

Key	Description
	Default value: "No"
ssl_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. Value type: list Default value: <none>
ssl_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, ssl!elliptic_ curves in the Global Settings section of the System tab. See there for how to specify SSL elliptic curves. Value type: string Default value: <none>
ssl_encrypt	Whether or not the pool should encrypt data before sending it to a back-end node. Value type: Yes / No Default value: "No"
ssl_enhance	SSL protocol enhancements allow your traffic manager to prefix each new SSL connection with information about the client. This enables Pulse Secure Virtual Traffic Manager 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 Pulse Secure vTMs, whose virtual servers have the ssl_trust_magic setting enabled. Value type: Yes / No Default value: "No"
ssl_fixed_client_certificate	An entry in the SSL client certificates catalog, containing a certificate and private key to be used whenever client authentication is requested. If set, this overrides server request parameters. Value type: string Default value: <none>

Key	Description
ssl_middlebox_compatibility	Whether or not TLS 1.3 middlebox compatibility mode as described in RFC 8446 appendix D.4 will be used in connections to pool nodes. Choosing the global setting means the value of configuration key ssl!middlebox_compatibility from the Global Settings section of the System tab will be enforced. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for use of middlebox compatibility enabled: Enable use of middlebox compatibility disabled: Disable use of middlebox compatibility
ssl_send_close_alerts	Whether or not to send an SSL/TLS "close alert" when initiating a socket disconnection. Value type: Yes / No Default value: "Yes"
ssl_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. Value type: Yes / No Default value: "No"
ssl_session_cache_enabled	Whether or not the SSL client cache will be used for this pool. Choosing the global setting means the value of the configuration key ssl!client_cache!enabled from the Global Settings section of the System tab will be enforced. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for use of the session cache enabled: Enable use of the session cache disabled: Disable use of the session cache

Key	Description
ssl_session_tickets_enabled	Whether or not SSL session tickets, including TLS >= 1.3 PSKs, will be used for this pool if the session cache is also enabled. Choosing the global setting means the value of the configuration key ssl!client_cache!tickets_enabled from the Global Settings section of the System tab will be enforced. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for use of session tickets enabled: Enable use of session tickets disabled: Disable use of session tickets
ssl_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, ssl!signature_algorithms in the Global Settings section of the System tab. See there for how to specify SSL signature algorithms. Value type: string Default value: <none>
ssl_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. Value type: Yes / No Default value: "No"
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. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for SSLv3 enabled: Enable SSLv3 disabled: Disable SSLv3

Key	Description
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. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for TLSv1.0 enabled: Enable TLSv1.0 disabled: Disable 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. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for TLSv1.1 enabled: Enable TLSv1.1 disabled: Disable TLSv1.1
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. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for TLSv1.2 enabled: Enable TLSv1.2 disabled: Disable TLSv1.2
ssl_support_tls1_3	Whether or not TLSv1.3 is enabled for this pool. Choosing the global setting means the value of the configuration key ssl!support_tls1_3 from the Global Settings section of the System tab will be enforced. Value type: enumeration

Key	Description
	Default value: "use_default" Permitted values: use_default: Use the global setting for TLSv1.3 enabled: Enable TLSv1.3 disabled: Disable TLSv1.3
transparent	Whether or not connections to the back-ends appear to originate from the source client IP address. Value type: Yes / No Default value: "No"
udp_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). Value type: enumeration Default value: "dest_only" Permitted values: dest_only: Only the IP address and port to which the request was sent. dest_ip_only: Only the IP address to which the request was sent, but from any port. ip_mask: Only a specific set of IP addresses, but from any port. all: Any IP address and any port.
udp_accept_from_mask	The CIDR mask that matches IPs we want to receive responses from. Requires: udp_accept_from is set to "ip_mask" Value type: string Default value: <none>
udp_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. Value type: seconds Default value: "0"

conf/protection

The conf/protection directory contains configuration files for service protection classes. The name of a file is the name of the protection class it defines. Service protection classes can be configured under the Catalogs > Protection section of the Admin Server UI or by using functions under the Catalog.Protection section of the SOAP API and CLI.

Key	Description
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. Value type: list Default value: <none>
banned	Disallow access to these IP addresses. Value type: list Default value: <none>
debug	Whether or not to output verbose logging. Value type: Yes / No Default value: "No"
enabled	Enable or disable this service protection class. Value type: Yes / No Default value: "Yes"
http!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. Value type: Yes / No Default value: "No"
http!max_body_length	Maximum permitted length of HTTP request body data, set to 0 to disable the limit. Value type: bytes Default value: "0"
http!max_header_length	Maximum permitted length of a single HTTP request header (key and value), set to 0 to disable the limit.

Key	Description
	Value type: bytes Default value: "0"
http!max_request_length	Maximum permitted size of all the HTTP request headers, set to 0 to disable the limit. Value type: bytes Default value: "0"
http!max_url_length	Maximum permitted URL length, set to 0 to disable the limit. Value type: bytes Default value: "0"
http!reject_binary	Whether or not URLs and HTTP request headers that contain binary data (after decoding) should be rejected. Value type: Yes / No Default value: "No"
http!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. Value type: Yes / No Default value: "Yes"
log_time	Log service protection messages at these intervals. If set to 0 no messages will be logged and no alerts will be sent. Value type: seconds Default value: "60"
max_10_connections	Additional limit on maximum concurrent 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.) Value type: unsigned integer Default value: "200"
max_1_connections	Maximum concurrent connections each connecting IP address is allowed. Set to 0 to disable this limit. Value type: unsigned integer

Key	Description
	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). Value type: unsigned integer Default value: "0"
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 concurrent 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.) Value type: unsigned integer Default value: "4"
note	A description of the service protection class. Value type: string Default value: <none>
per_process_connection_ count	Whether concurrent 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. Value type: Yes / No Default value: "Yes"

Key	Description
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. Value type: seconds Default value: "60"
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. Value type: string 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). Value type: Yes / No Default value: "No"

conf/rate

The conf/rate directory contains configuration files for request rate shaping classes. The name of a file is the name of the rate shaping class it defines. Request rate shaping classes can be configured under the Catalogs > Rate section of the Admin Server UI or by using functions under the Catalog.Rate section of the SOAP API and CLI.

Key	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. Value type: unsigned integer Default value: "0"

Key	Description
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. Value type: unsigned integer Default value: "0"
note	A description of the rate class. Value type: string Default value: <none>

conf/rules

The conf/rules directory contains plain text and compiled TrafficScript rule files. The name of a file is the name of the rule it defines. Rules are managed under the Catalogs > Rules section of the Admin Server UI or by using functions under the Catalog.Rule section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/saml/trustedidps

Configuration for SAML IDP trust relationships.

Key	Description
add_zlib_header	Whether or not to add the zlib header when compressing the AuthnRequest Value type: Yes / No Default value: "No"
certificate	The certificate used to verify Assertions signed by the identity provider Value type: string

Key	Description
	Default value: <none>
entity_id	The entity id of the IDP Value type: string Default value: <none>
strict_verify	Whether or not SAML responses will be verified strictly Value type: Yes / No Default value: "Yes"
url	The IDP URL to which Authentication Requests should be sent Value type: string Default value: <none>

conf/scripts

The conf/scripts directory contains programs and scripts that may be run by monitors of the program type. Monitor programs can be managed under the Catalogs > Extra Files > Monitor Programs section of the Admin Server UI or by using functions under the Catalog.Monitor section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/security

The conf/security file contains the security configuration of the software. Settings in this classes can be configured under the System > Security section of the Admin Server UI.

Key	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. Value type: list Default value: <none>
ssh_intrusion!bantime	The amount of time in seconds to ban an offending host for. Value type: unsigned integer Default value: "600"
ssh_intrusion!blacklist	The list of hosts to permanently ban, identified by IP address or DNS hostname in a space-separated list. Value type: list Default value: <none>
ssh_intrusion!enabled	Whether or not the SSH Intrusion Prevention tool is enabled. Value type: Yes / No Default value: "Yes"
ssh_intrusion!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. Value type: unsigned integer Default value: "600"
ssh_intrusion!maxretry	The number of failed connection attempts a host can make before being banned. Value type: unsigned integer Default value: "6"
ssh_intrusion!whitelist	The list of hosts to never ban, identified by IP address, DNS hostname or subnet mask, in a space-separated list. Value type: list Default value: <none>

conf/servicediscovery

The conf/servicediscovery directory contains plugins for use with Service Discovery for pool nodes.

Key	Description
There are no items to display for this configuration type.	

conf/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.

Key	Description
algorithm	Defines the global load balancing algorithm to be used. Value type: enumeration Default value: "hybrid" Permitted values: load: Load geo: Geographic hybrid: Adaptive roundrobin: Round Robin weightedrandom: Weighted Random chained: Primary/Backup
all_monitors_needed	Do all monitors assigned to a location need to report success in order for it to be considered healthy? Value type: Yes / No Default value: "Yes"
autofail	Enable/Disable automatic failback mode. Value type: Yes / No Default value: "No"
autorecovery	The last location to fail will be available as soon as it recovers. Value type: Yes / No Default value: "Yes"

Key	Description
dc!weight!*	Assign weights for each location. Value type: unsigned integer Default value: <none>
disable_on_failure	Locations recovering from a failure will become disabled. Value type: Yes / No Default value: "No"
dnssec!*	The domain this private key authenticates. Value type: list Default value: <none>
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. Value type: list Default value: <none>
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. Value type: list Default value: <none>
enabled	Enable/Disable our response manipulation of DNS. Value type: Yes / No Default value: "No"
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. Value type: unsigned integer Default value: "50"

Key	Description
last_resort_response	The response to be sent in case there are no locations available. Value type: list Default value: <none>
localips!*	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. Value type: list Default value: <none>
location_order	The locations this service operates for and defines the order in which locations fail. Value type: list Default value: <none>
log!enabled	Log connections to this GLB service? Value type: Yes / No Default value: "No"
log!filename	The filename the verbose query information should be logged to. Appliances will ignore this. Requires: log!enabled is set to "Yes" Value type: string Default value: "%zeushome%/zxtm/log/services/%g.log"
log!format	The format of the log lines. Requires: log!enabled is set to "Yes" Value type: string Default value: "%t, %s, %l, %q, %g, %n, %d, %a"
monitors!*	The monitors that are present in a location. Value type: list Default value: <none>
return_ips_on_fail	Return all or none of the IPs under complete failure. Value type: Yes / No Default value: "Yes"
rules	Response rules to be applied in the context of the service, in order, comma separated.

Key	Description
	Value type: list Default value: <none>
ttl	The TTL for the DNS resource records handled by the GLB service. Value type: int Default value: "-1"

conf/servlets

The conf/servlets directory contains configuration files for Java Extension servlets. If there are any parameters configured for a Java servlet, this configuration is stored in this directory in a file with the same name as the full name of the servlet. Servlet parameters are configured by clicking on the servlet name in the Catalogs > Java section of the Admin Server UI of by using the various "Properties" functions in the Catalog.JavaExtensions section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/settings.cfg

The conf/settings.cfg file contains general global settings that are used across a cluster. These setting are managed under the System > Global Settings section of the Admin Server UI or by using functions under the GlobalSettings section of the SOAP API and CLI.

Key	Description
admin!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. Value type: Yes / No Default value: "Yes"

Key	Description
admin!insert_extra_fragment	Whether or not admin server SSL3 and TLS1 use one-byte fragments as a BEAST countermeasure for admin server and internal connections. Value type: Yes / No Default value: "No"
admin!ssl3_allow_ rehandshake	Whether or not SSL3/TLS re-handshakes should be supported for admin server and internal connections. Value type: enumeration Default value: "rfc5746" Permitted values: always: Always allow safe: Allow safe re-handshakes rfc5746: Only if client uses RFC 5746 (Secure Renegotiation Extension) never: Never allow
admin!ssl3_ciphers	The SSL ciphers to use for admin server and internal connections. For information on supported ciphers see the online help. Value type: string Default value: <none>
admin!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. Value type: enumeration Default value: "2048" Permitted values: 1024: 1024 2048: 2048 3072: 3072 4096: 4096
admin!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.

Key	Description
	Value type: unsigned integer Default value: "1000"
admin!ssl_elliptic_curves	The SSL elliptic curve preference list for admin and internal connections. For information on supported curves see the online help. Value type: string Default value: <none>
admin!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. Value type: bytes Default value: "10240"
admin!ssl_prevent_timing_side_channels	This configuration is now obsolete and has no effect whether set or unset. Value type: Yes / No Default value: "No"
admin!ssl_signature_algorithms	The SSL signature algorithms preference list for admin and internal connections. For information on supported algorithms see the online help. Value type: string Default value: <none>
admin!support_ssl3	Whether or not SSL3 support is enabled for admin server and internal connections. Value type: Yes / No Default value: "No"
admin!support_tls1	Whether or not TLS1.0 support is enabled for admin server and internal connections. Value type: Yes / No Default value: "Yes"
admin!support_tls1_1	Whether or not TLS1.1 support is enabled for admin server and internal connections. Value type: Yes / No Default value: "Yes"

Key	Description
admin!support_tls1_2	Whether or not TLS1.2 support is enabled for admin server and internal connections. Value type: Yes / No Default value: "Yes"
admin!support_tls1_3	Whether or not TLS1.3 support is enabled for admin server and internal connections. Value type: Yes / No Default value: "Yes"
afm_enabled	Is the application firewall enabled. Value type: Yes / No Default value: "No"
allow_consecutive_chars	Whether or not to allow the same character to appear consecutively in passwords. Value type: Yes / No Default value: "Yes"
appliance!bootloader_ password	The password used to protect the bootloader. An empty string means there will be no protection. Value type: password Default value: <none>
appliance!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. Value type: Yes / No Default value: "No"
appliance!returnpath!*!ipv4	The MAC address/network interface to IPv4 address mapping of a router the software is connected to. The value is the IPv4 address, the * (asterisk) in the key name is the MAC address and an optional network interface name, for example, 00:50:56:a6:24:3d or 00:50:56:a6:24:3d#eth0. Value type: string

Key	Description
	Default value: <none>
appliance!returnpath!*!ipv6	The MAC address/network interface to IPv6 address mapping of a router the software is connected to. The value is the IPv6 address, the * (asterisk) in the key name is the MAC address and an optional network interface name, for example, 00:50:56:a6:24:3d or 00:50:56:a6:24:3d#eth0. Value type: string Default value: <none>
aptimizer!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. Value type: string Default value: "2MB"
aptimizer!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. Value type: string Default value: "2MB"
aptimizer!watchdog_interval	The period of time (in seconds) after which a previous failure will no longer count towards the watchdog limit. Value type: seconds Default value: "300"
aptimizer!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. Value type: unsigned integer

Key	Description
	Default value: "3"
asp_cache_size	The maximum number of entries in the ASP session persistence cache. This is used for storing session mappings for ASP session persistence. Approximately 100 bytes will be pre-allocated per entry. Value type: unsigned integer Default value: "32768"
auditlog!via_eventd	Whether to mirror the audit log to EventD. Value type: Yes / No Default value: "No"
auditlog!via_syslog	Whether to output audit log message to the syslog. Value type: Yes / No Default value: "No"
auth!saml!key_lifetime	Lifetime in seconds of cryptographic keys used to decrypt SAML SP sessions stored externally (client-side). Value type: seconds Default value: "86400"
auth!saml!key_rotation_ interval	Rotation interval in seconds for cryptographic keys used to encrypt SAML SP sessions stored externally (client-side). Value type: seconds Default value: "14400"
autoscaler!verbose	Whether or not detailed messages about the autoscaler's activity are written to the error log. Value type: Yes / No Default value: "No"
banner_accept	Whether or not users must explicitly agree to the displayed login_ banner text before logging in to the Admin Server. Value type: Yes / No Default value: "No"
bgp!as_number	The number of the BGP AS in which the traffic manager will operate. Must be entered in decimal. Value type: unsigned integer

Key	Description
	Default value: "65534"
bgp!enabled	Whether BGP Route Health Injection is enabled Value type: Yes / No Default value: "No"
chunk_size	The default chunk size for reading/writing requests. Value type: bytes 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. Value type: Yes / No Default value: "No"
cluster_identifier	Cluster identifier. Generally supplied by Services Director. Value type: string Default value: <none>
control!canupdate!default	The default value of control!canupdate for new cluster members. If you have cluster members joining from less trusted locations (such as cloud instances) this can be set to No in order to make them effectively "read-only" cluster members. Value type: Yes / No Default value: "Yes"
controlallow	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. Value type: string Default value: "all"
dns!max_ttl	Maximum Time To Live (expiry time) for entries in the DNS cache. Value type: seconds Default value: "86400"
dns!min_ttl	Minimum Time To Live (expiry time) for entries in the DNS cache. Value type: seconds

Key	Description
	Default value: "86400"
dns!negative_expiry	Expiry time for failed lookups in the DNS cache. Value type: seconds Default value: "60"
dns!size	Maximum number of entries in the DNS cache. Value type: unsigned integer Default value: "10867"
dns!timeout	Timeout for receiving a response from a DNS server. Value type: seconds Default value: "12"
ec2!access_key_id	Deprecated: This key is unused. Amazon authentication credentials are now extracted from IAM Roles assigned to an EC2 instance. Value type: string Default value: <none>
ec2!awstool_timeout	The maximum amount of time requests to the AWS Query API can take before timing out. Value type: unsigned integer Default value: "10"
ec2!metadata_server	URL for the EC2 metadata server, http://169.254.169.254/latest/meta-data for example. Value type: string Default value: <none>
ec2!query_server	URL for the Amazon EC2 endpoint, https://ec2.amazonaws.com/ for example. Value type: string Default value: <none>
ec2!secret_access_key	Deprecated: This key is unused. Amazon authentication credentials are now extracted from IAM Roles assigned to an EC2 instance. Value type: password Default value: <none>

Key	Description
ec2!verify_query_server_cert	Whether to verify Amazon EC2 endpoint's certificate using CA(s) present in SSL Certificate Authorities Catalog. Value type: Yes / No Default value: "No"
errlevel	The minimum severity of events/alerts that should be logged to disk. ERR_INFO will log all events; a higher severity setting will log fewer events. More fine-grained control can be achieved using events and actions in the Alerting section of the UI. Value type: enumeration Default value: "6" Permitted values: 1: ERR_FATAL 2: ERR_SERIOUS 5: ERR_WARN 6: ERR_INFO
errlog	The file to log event messages to. Value type: string Default value: "%zeushome%/zxtm/log/errors"
fips!enabled	Enable FIPS Mode (requires software restart). Value type: Yes / No Default value: "No"
flipper!arp_count	The number of ARP packets a traffic manager should send when an IP address is raised. Value type: unsigned integer Default value: "10"
flipper!autofailback	Whether or not traffic IPs automatically move back to machines that have recovered from a failure and have dropped their traffic IPs. Value type: Yes / No Default value: "Yes"
flipper!autofailback_delay	Configure the delay of automatic failback after a previous failover event. This setting has no effect if autofailback is disabled. Value type: seconds

Key	Description
	Default value: "10"
flipper!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. Value type: seconds Default value: "5"
flipper!frontend_check_addrs	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. Value type: list Default value: "%gateway%"
flipper!heartbeat_method	The method traffic managers should use to exchange cluster heartbeat messages. Value type: enumeration Default value: "unicast" Permitted values: multicast: multicast unicast: unicast
flipper!igmp_interval	The interval between unsolicited periodic IGMP Membership Report messages for Multi-Hosted Traffic IP Groups. Value type: seconds Default value: "30"
flipper!monitor_interval	The frequency, in milliseconds, that each traffic manager machine should check and announce its connectivity. Value type: unsigned integer Default value: "500"
flipper!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. Value type: seconds Default value: "5"

Key	Description
flipper!multicast_address	The multicast address and port to use to exchange cluster heartbeat messages. Requires: flipper!heartbeat_method is set to "multicast" Value type: string Default value: "239.100.1.1:9090"
flipper!unicast_port	The unicast UDP port to use to exchange cluster heartbeat messages. Requires: flipper!heartbeat_method is set to "unicast" Value type: unsigned integer Default value: "9090"
flipper!use_bindip	Whether or not cluster heartbeat messages should only be sent and received over the management network. Value type: Yes / No Default value: "No"
flipper!verbose	Whether or not a traffic manager should log all connectivity tests. This is very verbose, and should only be used for diagnostic purposes. Value type: Yes / No Default value: "No"
ftp_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. Value type: Yes / No Default value: "No"
gslb!verbose	Write a message to the logs for every DNS query that is load balanced, showing the source IP address and the chosen datacenter. Value type: Yes / No Default value: "No"
idle_connection_timeout	How long an unused HTTP keepalive connection should be kept before it is discarded. Value type: seconds

Key	Description
	Default value: "10"
ip_cache_expiry	IP session persistence cache expiry time in seconds. A session will not be reused if the time since it was last used exceeds this value. 0 indicates no expiry timeout. Value type: unsigned integer Default value: "0"
ip_cache_size	The maximum number of entries in the IP session persistence cache. This is used to provide session persistence based on the source IP address. Approximately 100 bytes will be pre-allocated per entry. Value type: unsigned integer Default value: "32768"
j2ee_cache_expiry	J2EE session persistence cache expiry time in seconds. A session will not be reused if the time since it was last used exceeds this value. 0 indicates no expiry timeout. Value type: unsigned integer Default value: "0"
j2ee_cache_size	The maximum number of entries in the J2EE session persistence cache. This is used for storing session mappings for J2EE session persistence. Approximately 100 bytes will be pre-allocated per entry. Value type: unsigned integer Default value: "32768"
java!classpath	CLASSPATH to use when starting the Java runner. Value type: string Default value: <none>
java!command	Java command to use when starting the Java runner, including any additional options. Value type: string Default value: "java -server"

Key	Description
java!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. Value type: Yes / No Default value: "No"
java!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. Value type: string Default value: <none>
java!max_conns	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. Value type: unsigned integer Default value: "256"
java!session_age	Default time to keep a Java session. Value type: seconds Default value: "86400"
kerberos!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. Value type: Yes / No Default value: "No"
license_servers	A list of license servers for FLA licensing. A license server should be specified as a <ip/host>:<port> pair. Value type: list Default value: <none>

Key	Description
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. Value type: unsigned integer Default value: "0"
load_change_limit	The maximum change to load per second, when monitored by GLB. This limit does not apply to external setting of the load by a SOAP agent. Value type: unsigned integer Default value: "800"
log!flushtime	How long to wait before flushing the request log files for each virtual server. Value type: seconds Default value: "5"
log!rate	The maximum number of connection errors logged per second when connection error reporting is enabled. Value type: unsigned integer Default value: "50"
log!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. Value type: seconds Default value: "30"
log!time	The minimum time between log messages for log intensive features such as SLM. Value type: seconds Default value: "60"
log_export!auth!hec_token	The HTTP Event Collector token to use for HTTP authentication with a Splunk server. Value type: string Default value: <none>

Key	Description
log_export!auth!http	The HTTP authentication method to use when exporting log entries. Value type: enumeration Default value: "none" Permitted values: none: None basic: Basic (Username and Password) splunk: Splunk (HEC token)
log_export!auth!password	The password to use for HTTP basic authentication. Value type: password Default value: <none>
log_export!auth!username	The username to use for HTTP basic authentication. Value type: string Default value: <none>
log_export!enabled	Monitor log files and export entries to the configured endpoint. Value type: Yes / No Default value: "No"
log_export!endpoint	The URL to which log entries should be sent. Entries are sent using HTTP(S) POST requests. Value type: string Default value: <none>
log_export!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. Value type: seconds Default value: "30"
log_export!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. Value type: Yes / No Default value: "Yes"

Key	Description
login_banner	Banner text displayed on the Admin Server login page and before logging in to appliance SSH servers. Value type: string Default value: <none>
login_delay	The number of seconds before another login attempt can be made after a failed attempt. Value type: seconds Default value: "4"
max_idle_connections	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. Value type: unsigned integer Default value: "0"
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. Value type: unsigned integer Default value: "0"
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). Value type: Yes / No Default value: "No"
max_login_suspension_time	The number of minutes to suspend users who have exceeded the max_login_attempts limit. Value type: unsigned integer Default value: "15"

Key	Description
max_tcp_buff_mem	The maximum amount of memory allowed to be used to buffer network data in user space for all TCP connections. The TCP data buffered are either received from clients but before sending to pool nodes, or recevied from pool nodes but before sending to clients. This is specified as either a percentage of system RAM, 5% for example, or an absolute size such as 1024MB and 2GB. A numeric value without suffix MB, GB or % defaults to MB. A value of 800 means 800MB. A value of 0 means unlimited. Value type: string Default value: "0"
maxfds	The maximum number of file descriptors that your traffic manager will allocate. Value type: unsigned integer Default value: "1048576"
min_alpha_chars	Minimum number of alphabetic characters a password must contain. Set to 0 to disable this restriction. Value type: unsigned integer Default value: "0"
min_numeric_chars	Minimum number of numeric characters a password must contain. Set to 0 to disable this restriction. Value type: unsigned integer Default value: "0"
min_password_length	Minimum number of characters a password must contain. Set to 0 to disable this restriction. Value type: unsigned integer Default value: "0"
min_special_chars	Minimum number of special (non-alphanumeric) characters a password must contain. Set to 0 to disable this restriction. Value type: unsigned integer Default value: "0"
min_uppercase_chars	Minimum number of uppercase characters a password must contain. Set to 0 to disable this restriction. Value type: unsigned integer

Key	Description
	Default value: "0"
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. Value type: unsigned integer Default value: "4096"
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 'No'. Value type: Yes / No Default value: "No"
notify!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. Value type: seconds Default value: "30"
notify!max_attempts	The number of times to attempt to send an alert email before giving up. Value type: unsigned integer Default value: "10"
ospfv2!area	The OSPF area in which the traffic manager will operate. May be entered in decimal or IPv4 address format. Value type: string Default value: "0.0.0.1"
ospfv2!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. Value type: enumeration Default value: "normal" Permitted values:

Key	Description
	normal: Normal area stub: Stub area nssa: Not So Stubby Area (RFC3101)
ospfv2!authentication_key_id_a	OSPFv2 authentication key ID. If set to 0, which is the default value, the key is disabled. Value type: unsigned integer Default value: "0"
ospfv2!authentication_key_id_b	OSPFv2 authentication key ID. If set to 0, which is the default value, the key is disabled. Value type: unsigned integer Default value: "0"
ospfv2!authentication_shared_secret_a	OSPFv2 authentication shared secret (MD5). If set to blank, which is the default value, the key is disabled. Value type: string Default value: <none>
ospfv2!authentication_shared_secret_b	OSPFv2 authentication shared secret (MD5). If set to blank, which is the default value, the key is disabled. Value type: string Default value: <none>
ospfv2!dead_interval	The number of seconds before declaring a silent router down. Value type: seconds Default value: "40"
ospfv2!enabled	Whether OSPFv2 Route Health Injection is enabled Value type: Yes / No Default value: "No"
ospfv2!hello_interval	The interval at which OSPF "hello" packets are sent to the network. Value type: seconds Default value: "10"
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. Value type: unsigned integer Default value: "0"

Key	Description
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. Value type: unsigned integer Default value: "0"
post_login_banner	Banner text to be displayed on the appliance console after login. Value type: string Default value: <none>
protection!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. Value type: string Default value: "20MB"
rate_class_limit	The maximum number of Rate classes that can be created. Approximately 100 bytes will be pre-allocated per Rate class. Value type: unsigned integer Default value: "25000"
recent_conns	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. Value type: unsigned integer Default value: "500"
recent_conns_retain_time	The amount of time for which snapshots will be retained on the Connections page. Value type: seconds Default value: "60"
recent_conns_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.

Key	Description
	Value type: unsigned integer Default value: "500"
remote_licensing!comm_channel_enabled	Whether to create a Communications Channel agent to send and receive messages from the Services Director Registration Server. This will be disabled when performing self-registration with a Services Director which does not support this feature. Value type: Yes / No Default value: "Yes"
remote_licensing!comm_channel_port	The port number the Services Director instance is using for access to the traffic manager Communications Channel. Value type: unsigned integer Default value: "8102"
remote_licensing!owner	The Owner of a Services Director instance, used for self-registration. Value type: string Default value: <none>
remote_licensing!owner_secret	The secret associated with the Owner. Value type: string Default value: <none>
remote_licensing!policy_id	The auto-accept Policy ID that this instance should attempt to use. Value type: string Default value: <none>
remote_licensing!registration_server	A Services Director address for self-registration. A registration server should be specified as a <ip/host>:<port> pair. Value type: string Default value: <none>
remote_licensing!server_certificate	The certificate of a Services Director instance, used for self-registration. Value type: string Default value: <none>

Key	Description
rest!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. Value type: seconds Default value: "120"
rest!enabled	Whether or not the REST service is enabled. Value type: Yes / No Default value: "Yes"
rest!max_http_header_len	The maximum allowed length in bytes of a HTTP request's headers. Value type: unsigned integer Default value: "4096"
rest!maxfds	Maximum number of file descriptors that the REST API will allocate. The REST API must be restarted for a change to this setting to take effect. Value type: unsigned integer Default value: "1048576"
rest!repabstime	Configuration changes will be replicated across the cluster after this period of time, regardless of whether additional API requests are being made. Value type: seconds Default value: "20"
rest!replulltime	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. Value type: seconds Default value: "5"
rest!reptimeout	The period of time after which configuration replication across the cluster will be cancelled if it has not completed. Value type: seconds Default value: "10"

Key	Description
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. Value 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. Value type: unsigned integer Default value: "1024"
snmp_user_counters	The number of user defined SNMP counters. Approximately 100 bytes will be pre-allocated at start-up per user defined SNMP counter. Value type: unsigned integer Default value: "10"
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. Value type: bytes Default value: "0"
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. Value type: bytes Default value: "0"
soap!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. Value type: unsigned integer Default value: "10"

Key	Description
socket_opt	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. Value type: enumeration Default value: "auto" Permitted values: auto: auto Yes: Yes No: No
ssl!allow_rehandshake	Whether or not SSL/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. Value type: enumeration Default value: "safe" Permitted values: always: Always allow safe: Allow safe re-handshakes rfc5746: Only if client uses RFC 5746 (Secure Renegotiation Extension) never: Never allow
ssl!cache!enabled	Whether or not the SSL server session cache is enabled, unless overridden by virtual server settings. Value type: Yes / No Default value: "Yes"
ssl!cache!expiry	How long the SSL session IDs for SSL decryption should be stored for. Value type: seconds Default value: "1800"
ssl!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. Value type: Yes / No Default value: "Yes"

Key	Description
ssl!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. Each entry will allocate approximately 1.75kB of metadata. Value type: unsigned integer Default value: "6151"
ssl!cipher_suites	The SSL/TLS cipher suites preference list for SSL/TLS connections, unless overridden by virtual server or pool settings. For information on supported cipher suites see the online help. Value type: string Default value: <none>
ssl!client_cache!enabled	Whether or the SSL client cache will be used, unless overridden by pool settings. Value type: Yes / No Default value: "Yes"
ssl!client_cache!expiry	How long in seconds SSL sessions should be stored in the client cache for, by default. Servers returning session tickets may also provide a lifetime hint, which will be used if it is less than this value. Value type: seconds Default value: "14400"
ssl!client_cache!size	How many entries the SSL client session cache should hold, per child. This cache is used to cache SSL sessions to help speed up SSL handshakes when performing SSL encryption. Each entry will require approx 100 bytes of memory plus space for either an SSL session id or an SSL session ticket, which may be as small as 16 bytes or may be as large as a few kilobytes, depending upon the server behavior. Value type: unsigned integer Default value: "1024"
ssl!client_cache!tickets_enabled	Whether or not session tickets, including TLS >= 1.3 PSKs, may be requested and stored in the SSL client cache. Value type: Yes / No

Key	Description
	Default value: "Yes"
ssl!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. Value type: string Default value: "5MB"
ssl!diffie_hellman_modulus_size	The size in bits of the modulus for the domain parameters used for cipher suites that use finite field Diffie-Hellman key agreement. Value type: enumeration Default value: "2048" Permitted values: 1024: 1024 2048: 2048 3072: 3072 4096: 4096
ssl!elliptic_curves	The SSL/TLS elliptic curve preference list for SSL/TLS connections using TLS version 1.0 or higher, unless overridden by virtual server or pool settings. For information on supported curves see the online help. Value type: string Default value: <none>
ssl!honor_fallback_scsv	Whether or not ssl-decrypting Virtual Servers honor the Fallback SCSV to protect connections against downgrade attacks. Value type: Yes / No Default value: "Yes"
ssl!insert_extra_fragment	Whether or not SSL3 and TLS1 use one-byte fragments as a BEAST countermeasure. Value type: Yes / No Default value: "No"
ssl!log_keys	Whether SSL connection key logging should be available via the ssl.sslkeylogline() TrafficScript function. If this setting is disabled then ssl.sslkeylogline() will always return the empty string.

Key	Description
	Value type: Yes / No Default value: "No"
ssl!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. Value type: bytes Default value: "10240"
ssl!middlebox_compatibility	Whether or not TLS 1.3 middlebox compatibility mode as described in RFC 8446 appendix D.4 will be used in connections to pool nodes, unless overridden by pool settings. Value type: Yes / No Default value: "Yes"
ssl!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. Value type: unsigned integer Default value: "1000"
ssl!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. Value type: unsigned integer Default value: "2048"
ssl!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. Value type: seconds Default value: "60"
ssl!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.)

Key	Description
	Value type: seconds Default value: "864000"
ssl!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. Value type: string Default value: "1MB"
ssl!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. Value type: seconds Default value: "30"
ssl!ocsp_stapling!verify_ response	Whether the OCSP response signature should be verified before the OCSP response is cached. Value type: Yes / No Default value: "No"
ssl!prevent_timing_side_ channels	This configuration is now obsolete and has no effect whether set or unset. Value type: Yes / No Default value: "No"
ssl!signature_algorithms	The SSL/TLS signature algorithms preference list for SSL/TLS connections using TLS version 1.2 or higher, unless overridden by virtual server or pool settings. For information on supported algorithms see the online help. Value type: string Default value: <none>
ssl!support_ssl3	Whether or not SSL3 support is enabled. Requires: fips!enabled is set to "Yes" Value type: Yes / No Default value: "No"
ssl!support_tls1	Whether or not TLS1.0 support is enabled. Value type: Yes / No

Key	Description
	Default value: "Yes"
ssl!support_tls1_1	Whether or not TLS1.1 support is enabled. Value type: Yes / No Default value: "Yes"
ssl!support_tls1_2	Whether or not TLS1.2 support is enabled. Value type: Yes / No Default value: "Yes"
ssl!support_tls1_3	Whether or not TLS1.3 support is enabled. Value type: Yes / No Default value: "Yes"
ssl!tickets!enabled	Whether or not session tickets will be issued to and accepted from clients that support them, unless overridden by virtual server settings. Value type: Yes / No Default value: "Yes"
ssl!tickets!reissue_policy	When an SSL session ticket will be reissued (ie when a new ticket will be generated for the same SSL session). Value type: enumeration Default value: "never" Permitted values: always: always never: never
ssl!tickets!ticket_expiry	The length of time for which an SSL session ticket will be accepted by a virtual server after the ticket is created. If a ticket is reissued (if ssl!tickets!reissue_policy is set to 'always') this time starts at the time when the ticket was reissued. Value type: seconds Default value: "14400"
ssl!tickets!ticket_key_expiry	The length of time for which an auto-generated SSL ticket key will be used to decrypt old session ticket, before being deleted from memory. This setting is ignored if there are any entries in the (REST-only) SSL ticket keys catalog. Value type: seconds

Key	Description
	Default value: "86400"
ssl!tickets!ticket_key_rotation	The length of time for which an auto-generated SSL ticket key will be used to encrypt new session tickets, before a new SSL ticket key is generated. The ticket encryption key will be held in memory for ssl!tickets!ticket_key_expiry, so that tickets encrypted using the key can still be decrypted and used. This setting is ignored if there are any entries in the (REST-only) SSL ticket keys catalog. Value type: seconds Default value: "14400"
ssl!tickets!time_tolerance	How many seconds to allow the current time to be outside the validity time of an SSL ticket before considering it invalid. Value type: seconds Default value: "30"
ssl!validate_server_certificates_catalog	Whether the traffic manager should validate that SSL server certificates form a matching key pair before the certificate gets used on an SSL decrypting virtual server. Value type: Yes / No Default value: "Yes"
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. Value type: unsigned integer Default value: "32768"
ssld!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. Value type: Yes / No Default value: "No"

Key	Description
ssld!azure!client_id	The client identifier used when accessing the Microsoft Azure Key Vault. Value type: string Default value: <none>
ssld!azure!client_secret	The client secret used when accessing the Microsoft Azure Key Vault. Value type: password Default value: <none>
ssld!azure!vault_url	The URL for the REST API of the Microsoft Azure Key Vault. Value type: string Default value: <none>
ssld!azure!verify_rest_api_cert	Whether or not the Azure Key Vault REST API certificate should be verified. Value type: Yes / No Default value: "Yes"
ssld!driver!pkcs11_debug	Print verbose information about the PKCS11 hardware security module to the event log. Value type: Yes / No Default value: "No"
ssld!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. Value type: string Default value: <none>
ssld!driver!pkcs11_slot_desc	The label of the SSL Hardware slot to use. Only required if you have multiple HW accelerator slots. Value type: string Default value: <none>
ssld!driver!pkcs11_slot_type	The type of SSL hardware slot to use. Value type: enumeration Default value: "operator" Permitted values: operator: Operator Card Set

Key	Description
	softcard: Soft Card module: Module Protected
ssld!driver!pkcs11_user_pin	The User PIN for the PKCS token (PKCS#11 devices only). Value type: password Default value: <none>
ssld!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. Value type: unsigned integer Default value: "5"
ssld!library	The type of SSL hardware to use. The drivers for the SSL hardware should be installed and accessible to the traffic manager software. Value type: enumeration Default value: "none" Permitted values: none: None pkcs11: PKCS#11 azure: Microsoft Azure Key Vault
statd!days	Number of days to store historical traffic information, if set to 0 the data will be kept indefinitely. Value type: unsigned integer Default value: "90"
state_sync_time	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. Value type: seconds Default value: "3"

Key	Description
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. Value type: seconds Default value: "6"
telemetry!enabled	Allow the reporting of anonymized usage data for product improvement and customer support purposes. Value type: Yes / No Default value: "Yes"
tip_class_limit	The maximum number of Traffic IP Groups that can be created. Value type: unsigned integer Default value: "10000"
track_unknown_users	Whether to remember past login attempts from usernames that are not known to exist (should be set to No for an Admin Server accessible from the public Internet). This does not affect the audit log. Value type: Yes / No Default value: "No"
trafficscript!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. Value type: string Default value: "5%"
trafficscript!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. Value type: string Default value: "5%"

Key	Description
trafficscript!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. Value type: unsigned integer Default value: "500"
trafficscript!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. Value type: unsigned integer Default value: "100000"
trafficscript!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 (). Value type: bytes Default value: "1048576"
trafficscript!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. Value type: unsigned integer Default value: "57"
trafficscript!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. Value type: unsigned integer Default value: "10000000"
trafficscript!regex_match_warn_perc	The percentage of trafficscript!regex_match_limit at which TrafficScript reports a performance warning.

Key	Description
	Value type: unsigned integer Default value: "5"
trafficscript!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 effects: Your traffic manager may no longer be able to know whether a pool is in use. Errors for pools that aren't in use will not be hidden. Some settings displayed for a Pool may not be appropriate for the type of traffic being managed. Pool usage information on the pool edit pages and config summary may not be accurate. Monitors will run for all pools (with this option disabled monitors will only run for Pools that are used). Value type: Yes / No Default value: "No"
transaction_export!enabled	Export metadata about transactions processed by the traffic manager to an external location. Value type: Yes / No Default value: "No"
transaction_export!endpoint	The endpoint to which transaction metadata should be exported. The endpoint is specified as a hostname or IP address with a port. Value type: string Default value: <none>
transaction_export!tls	Whether the connection to the specified endpoint should be encrypted. Value type: Yes / No Default value: "Yes"
transaction_export!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. Value type: Yes / No Default value: "Yes"

Key	Description
udp_read_multiple	Whether or not the traffic manager should try to read multiple UDP packets from clients each time the kernel reports data received from clients. This can improve performance for the situation with high UDP traffic throughput from clients to the traffic manager. Therefore, in general it is recommended that this be set to 'Yes'. Value type: Yes / No Default value: "Yes"
uipage_banner	Banner text to be displayed on all Admin Server pages. Value type: string Default value: <none>
universal_cache_expiry	Universal session persistence cache expiry time in seconds. A session will not be reused if the time since it was last used exceeds this value. 0 indicates no expiry timeout. Value type: unsigned integer Default value: "0"
universal_cache_size	The maximum number of entries in the global universal session persistence cache. This is used for storing session mappings for universal session persistence. Approximately 100 bytes will be pre-allocated per entry. Value type: unsigned integer Default value: "32768"
watchdog!timeout	The maximum time in seconds a process can fail to update its heartbeat, before the watchdog considers it to have stalled. Value type: seconds Default value: "5"
webcache!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. Value type: unsigned integer Default value: "512"

Key	Description
webcache!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 webcache!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. Value type: Yes / No Default value: "No"
webcache!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. Value type: string Default value: "%zeushome%/zxtm/internal"
webcache!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. Value type: unsigned integer Default value: "10000"
webcache!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. Value type: string Default value: "2%"
webcache!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. Value type: unsigned integer Default value: "2048"
webcache!normalize_query	Enable normalization (lexical ordering of the parameter-assignments) of the query string. Value type: Yes / No

Key	Description
	Default value: "Yes"
webcache!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. Value type: string Default value: "20%"
webcache!verbose	Add an X-Cache-Info header to every HTTP response, showing whether the request and/or the response was cacheable. Value type: Yes / No Default value: "No"

conf/slm

The conf/slm directory contains configuration files for service level monitoring (SLM) classes. The name of a file is the name of the SLM class it defines. SLM classes can be configured under the Catalogs > SLM section of the Admin Server UI or by using functions under the Catalog.SLM section of the SOAP API and CLI.

Key	Description
note	A description for the SLM class. Value type: string Default value: <none>
response_time	Responses that start being sent to the client within this time limit, expressed in milliseconds, are treated as conforming. Value type: unsigned integer Default value: "1000"
serious_threshold	When the percentage of conforming responses drops below this level, a serious error level message will be emitted. Value type: unsigned integer Default value: "0"
warning_threshold	When the percentage of conforming responses drops below this level, a warning message will be emitted.

Key	Description
	Value type: unsigned integer Default value: "50"

conf/ssl/admin_cas

The conf/ssl/admin_cas directory contains SSL certificate authority certificates (CAs) and certificate revocation lists (CRLs) which can be used when validating connections made by the admin server for user authentication. CAs and CRLs can be managed under the Catalogs > SSL > Admin CAs and CRLs section of the Admin Server UI or by using functions under the Catalog.SSL.AdminCertificateAuthorities section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/ssl/cas

The conf/ssl/cas directory contains SSL certificate authority certificates (CAs) and certificate revocation lists (CRLs) which can be used when validating server and client certificates. CAs and CRLs can be managed under the Catalogs > SSL > CAs and CRLs section of the Admin Server UI or by using functions under the Catalog.SSL.CertificateAuthorities section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/ssl/client_keys

The conf/ssl/client_keys directory contains SSL public and private key files for use when connecting to backend nodes that require clients certificate authentication. For each key managed by the software there will be two files, the file names give the name of the SSL keypair followed by .public or .private depending on which key is in the file. Client keys keys can be managed under the Catalogs > SSL > Client Certs section of the Admin Server UI or by using functions under the Catalog.SSL.ClientCertificates section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/ssl/dnssec_keys

Config for DNSSEC private keys. Contains the keys id and algorithm followed by a RSA key block. Other key types can be converted using our cert tool.

Key	Description
There are no items to display for this configuration type.	

conf/ssl/server_keys

The conf/ssl/server_keys directory contains SSL public and private key files for use with virtual servers that have ssl_decrypt enabled. For each key managed by the software there will be two files, the file names give the name of the SSL keypair followed by .public or .private depending on which key is in the file. If a keypair was generated by the software there will also be a corresponding .request file (which can be used to have your key signed by a CA). Server keys can be managed under the Catalogs > SSL > Server Certs section of the Admin Server UI or by using functions under the Catalog.SSL.Certificates section of the SOAP API and CLI.

Key	Description
There are no items to display for this configuration type.	

conf/ssl/ticket_keys

Configuration for SSL ticket encryption keys when managed externally via the ssl/ticket_keys REST API endpoints.

Key	Description
algorithm	The algorithm used to encrypt session tickets. The algorithm determines the length of the key that must be provided. Value type: enumeration Default value: "aes_256_cbc_hmac_sha256" Permitted values: aes_256_cbc_hmac_sha256: AES-256 CBC with HMAC-SHA256. Requires a total of 64 bytes of key material.
id	A 16-byte key identifier, with each byte encoded as two hexadecimal digits. Key identifiers are transmitted in plaintext at the beginning of a TLS session ticket, and are used to identify the ticket encryption key that was used to encrypt a ticket. (They correspond to the 'key_name' field in RFC 5077.) They are required to be unique across the set of SSL ticket encryption keys. Value type: string Default value: <none>
key	The session ticket encryption key, with each byte encoded as two hexadecimal digits. The required key length is determined by the chosen key algorithm. See the documentation for the 'algorithm' field for more details. Value type: password Default value: <none>
validity_end	The latest time at which this key may be used to encrypt new session tickets. Given as number of seconds since the epoch (1970-01-01T00:00:00Z). Value type: seconds Default value: <none>
validity_start	The earliest time at which this key may be used to encrypt new session tickets. Given as number of seconds since the epoch (1970-01-01T00:00:00Z). Value type: seconds Default value: <none>

conf/users

The conf/users file defines login details for users with access to the software. This is a single file containing details for all locally managed users of the software. The asterisk (*) in the keys represents the name of the user the key applies to, for example the key to store the applet width for the 'admin' user is user!admin!appletwidth. Users are managed under the System > Users section of the web UI. In the SOAP API and CLI users are managed using functions in the Users section.

Key	Description
user!*!applet_max_vs	The maximum number of virtual server traffic bars to show in the applet. Value type: unsigned integer Default value: "5"
user!*!group	The user's Permission Group. Value type: string Default value: <none>
user!*!old_password!*	A salted MD5 hash of the user's nth most recent password. This config key is used to track older passwords set by an user to implement password policy settings. This key's value is updated by the software only. The config key is of the form 'user!<username>!old_password!<n>' where <username> is the name of the user and <n> takes integer values starting from 0 and signifies the nth most recent password. Value type: password Default value: <none>
user!*!old_password_ timestamp!*	The timestamp of the nth most recent password. This key is of the form user!<username>!old_password_timestamp!<n>, and stores the timestamp when the user!<username>!old_password!<n> was recorded in the file. See 'user!*!old_password!*' config key description for more details. Value type: string Default value: <none>
user!*!password	A salted MD5 hash of the user's password. User records from older versions of the software may use a crypt()-style hash. Value type: password

Key	Description
	Default value: <none>
user!*!password!timestamp	Timestamp representing the time that the current password was created. This is used internally by the software to track password expiry. Value type: string Default value: <none>
user!*!status	The user's status. Value type: enumeration Default value: "1" Permitted values: 1: Active 2: Suspended
user!*!trafficscript_editor	Use the advanced TrafficScript editor when modifying rules. This adds automatic line numbering, syntax highlighting and indentation. Value type: Yes / No Default value: "Yes"
user!*!use_applet	Enable the Admin Server UI traffic monitoring applet. Value type: Yes / No Default value: "Yes"

conf/vservers

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.

Key	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.

Key	Description
	Value type: Yes / No Default value: "Yes"
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. Value type: Yes / No Default value: "No"
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. Value type: Yes / No Default value: "No"
address	The addresses on which to listen for incoming connections. Value type: list Default value: "*"
alt_certificates	The SSL certificates and corresponding private keys. Requires: ssl_decrypt is set to "Yes" Value type: list Default value: <none>
aptimizer!enabled	Whether the virtual server should optimize web content. Value type: Yes / No Default value: "No"
aptimizer!profile!*!urls	The application scopes for which to apply a particular acceleration profile. Value type: list Default value: <none>
auth!saml!idp	Name of the Trusted Identity Provider configuration to use. To create Identity Providers, please visit section Trusted Identity Providers Value type: string Default value: <none>

Key	Description
auth!saml!nameid_format	The NameID format to request and expect from the identity provider. Value type: enumeration Default value: "none" Permitted values: none: none unspecified: unspecified emailAddress: emailAddress
auth!saml!sp_acs_url	The 'Assertion Consumer Service' endpoint for the SAML service provider on this virtual server, ie the endpoint to which the identity provider will cause the user agent to send SAML assertions. This should be an HTTPS URL, must be in the same cookie domain as all hostnames used by the end user to access the virtual server (see cookie configuration) and the port must be the port on which this virtual server is listening. It must match the URI placed by the identity provider in the 'Recipient' attribute in the SAML assertion, if present. Value type: string Default value: <none>
auth!saml!sp_entity_id	The entity ID to be used by the SAML service provider function on this virtual server. This should usually be a URL, or a URN, however it may be any string. It must match the entity ID placed by the identity provider in the 'Audience' field in the SAML assertion. Value type: string Default value: <none>
auth!saml!time_tolerance	Time tolerance on authentication checks. When checking time-stamps and expiry dates against the current time on the system, allow a tolerance of this many seconds. For example, if a SAML response contains a 'NotOnOrAfter' that is 4 seconds in the past according to the local time, and the tolerance is set to 5 seconds, it will still be accepted. This is to prevent a lack of clock synchronization from resulting in rejection of SAML responses. Value type: seconds Default value: "5"

Key	Description
auth!session!cookie_ attributes	Attributes of cookie used for authentication session. Value type: string Default value: "HttpOnly; SameSite=Strict"
auth!session!cookie_name	Name of cookie used for authentication session. Value type: string Default value: "VS_SamlSP_Auth"
auth!session!log_external_ state	Whether or not to include state of authentication sessions stored encrypted on the client as plaintext in the logs. Value type: Yes / No Default value: "No"
auth!session!timeout	Timeout on authentication session. Value type: seconds Default value: "7200"
auth!type	Type of authentication to apply to requests to the virtual server. Value type: enumeration Default value: "none" Permitted values: none: None saml_sp: SAML Service Provider
auth!verbose	Whether or not detailed messages about virtual server authentication should be written to the error log. Value type: Yes / No Default value: "No"
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. Value type: Yes / No Default value: "Yes"
bandwidth_class	The bandwidth management class that this server should use, if any. Value type: string

Key	Description
	Default value: <none>
ca_sites!*!cert_headers	Which parts of the client certificate, if any, should be inserted into requests to a back-end node, as header fields. The same fields as for ssl_client_cert_headers are made available, and optionally the base64 encoded certificate itself. Value type: enumeration Default value: <none> Permitted values: none: None simple: Fields all: Fields and PEM
ca_sites!*!client_cas	The certificate authorities used to verify client certificates for a particular destination site IP or SNI hostname. The specific site replaces the * (asterisk) in the key name, the value must be a valid file name in the conf/ssl/cas directory. The key can be specified multiple times to cover multiple IP addresses or SNI hostnames. Requires: ssl_decrypt is set to "Yes" Value type: list Default value: <none>
ca_sites!*!request_cert	Whether or not the virtual server should request an identifying certificate from each client connecting to particular destination IP address or SNI hostname. If a client certificate is requested this setting also determines whether the TLS handshake can continue successfully if the client does not present a certificate. Value type: enumeration Default value: <none> Permitted values: 0: No 1: Yes, allow if absent 2: Yes, deny if absent
client_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.

Key	Description
	Requires: ssl_decrypt is set to "Yes" Value type: list Default value: <none>
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. Value type: Yes / No Default value: "No"
completionrules	Rules that are run at the end of a transaction, in order, comma separated. Value type: list Default value: <none>
connect_timeout	The time, in seconds, for which an established connection can remain idle waiting for some initial data to be received from the client. The initial data is defined as a complete set of request headers for HTTP, SIP and RTSP services, or the first byte of data for all other services. A value of 0 will disable the timeout. Value type: seconds Default value: "10"
cookie!domain	The way in which the traffic manager should rewrite the domain portion of any cookies set by a back-end web server. Value type: enumeration Default value: "0" Permitted values: 0: Do not rewrite the domain 1: Rewrite the domain to the host header of the request 2: Rewrite the domain to the named domain value
cookie!newdomain	The domain to use when rewriting a cookie's domain to a named value. Requires: cookie!domain is set to "2" Value type: string Default value: <none>

Key	Description
cookie!pathregex	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: Value type: string Default value: <none>
cookie!pathreplace	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. Requires: cookie!pathregex is set to a regular expression Value type: string Default value: <none>
cookie!secure	Whether or not the traffic manager should modify the "secure" tag of any cookies set by a back-end web server. Value type: enumeration Default value: "0" Permitted values: 0: Do not modify the 'secure' tag 1: Set the 'secure' tag 2: Unset the 'secure' tag
dns!edns_client_subnet	Enable/Disable use of EDNS client subnet option Value type: Yes / No Default value: "Yes"
dns!edns_udpsize	EDNS UDP size advertised in responses. Value type: unsigned integer Default value: "4096"
dns!max_udpsize	Maximum UDP answer size. Value type: unsigned integer Default value: "4096"
dns!rrset_order	Response record ordering. Value type: enumeration Default value: "fixed" Permitted values: fixed: Fixed cyclic: Cyclic

Key	Description
dns!verbose	Whether or not the DNS Server should emit verbose logging. This is useful for diagnosing problems. Value type: Yes / No Default value: "No"
dns!zones	The DNS zones Value type: list Default value: <none>
enabled	Whether the virtual server is enabled. Value type: Yes / No Default value: "No"
error_file	Specify how the traffic manager should respond to the client when an internal or backend error is detected. In addition to sending custom or default error pages, the traffic manager can be instructed to close the connection without returning a response. Custom error pages can be uploaded via the Extra Files catalog page. Value type: string Default value: "Default"
ftp!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). Requires: ssl_decrypt is set to "Yes" Value type: Yes / No Default value: "Yes"
ftp_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 ftp_data_bind_low global setting. Value type: unsigned integer Default value: "0"
ftp_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. Value type: Yes / No

Key	Description
	Default value: "Yes"
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. Value type: Yes / No Default value: "Yes"
ftp_portrange_high	If non-zero, then this controls the upper bound of the port range to use for FTP data connections. Value type: unsigned integer Default value: "0"
ftp_portrange_low	If non-zero, then this controls the lower bound of the port range to use for FTP data connections. Value type: unsigned integer Default value: "0"
glb_services	The associated GLB services for this DNS virtual server. Value type: list Default value: <none>
gzip!compresslevel	Compression level (1-9, 1=low, 9=high). Value type: unsigned integer Default value: "1"
gzip!enabled	Compress web pages sent back by the server. Value type: Yes / No Default value: "No"
gzip!etag_rewrite	How the ETag header should be manipulated when compressing content. Value type: enumeration Default value: "wrap" Permitted values: ignore: Leave the ETag unchanged delete: Delete the ETag header weaken: Change the ETag header to specify a weak match wrap: Wrap the ETag, and attempt to unwrap safe conditional requests

Key	Description
gzip!include_mime	MIME types to compress. Complete MIME types can be used, or a type can end in a '*' to match multiple types. Value type: list Default value: "text/html text/plain"
gzip!maxsize	Maximum document size to compress (0 means unlimited). Value type: bytes Default value: "10000000"
gzip!minsize	Minimum document size to compress. Value type: bytes Default value: "1000"
gzip!nosize	Compress documents with no given size. Value type: Yes / No Default value: "Yes"
http2!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. Value type: unsigned integer Default value: "0"
http2!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. Value type: bytes Default value: "4096"
http2!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. Value type: Yes / No Default value: "Yes"

Key	Description
http2!header_table_size	This setting controls the amount of memory allowed for header compression on each HTTP/2 connection. Value type: bytes Default value: "4096"
http2!headers_index_blacklist	A list of header names that should never be compressed using indexing. Value type: list Default value: <none>
http2!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. Value type: Yes / No Default value: "Yes"
http2!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. Value type: list Default value: <none>
http2!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. Value type: unsigned integer Default value: "262144"

Key	Description
http2!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. Value type: unsigned integer Default value: "120"
http2!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. Value type: unsigned integer Default value: "600"
http2!max_concurrent_ streams	This setting controls the number of streams a client is permitted to open concurrently on a single connection. Value type: unsigned integer Default value: "200"
http2!max_frame_size	This setting controls the maximum HTTP/2 frame size clients are permitted to send to the traffic manager. Value type: bytes Default value: "16384"
http2!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. Value type: bytes Default value: "0"
http2!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. Value type: Yes / No Default value: "Yes"

Key	Description
http2!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. Value type: bytes Default value: "65535"
http2_client_buffer_multiplier	The amount of memory, in multiples of the value specified by max_client_buffer, that the virtual server should use to store data sent by a client through a HTTP/2 connection. The value specified can be between 0 and 200. The value of 0 means unlimited. This setting limits buffer size for a HTTP/2 connection and does not affect buffer size for HTTP/1 connections or TCP stream connections. The number of HTTP/2 streams that can be opened in a single HTTP/2 connection is given by the http2!max_concurrent_streams. An overall cap to the amount of memory allocated for buffers for all TCP connections is given by the global max_tcp_buff_mem setting. Value type: unsigned integer Default value: "0"
http2_server_buffer_multiplier	The amount of memory, in multiples of the value specified by max_server_buffer, that the virtual server should use to store data sent to a client through HTTP/2 connection. The value specified can be between 0 and 200. The value of 0 means unlimited. This setting limits buffer size for a HTTP/2 connection and does not affect buffer size for HTTP/1 connections or TCP stream connections. The number of HTTP/2 streams that can be opened in a single HTTP/2 connection is given by the http2!max_concurrent_streams. An overall cap to the amount of memory allocated for buffers for all TCP connections is given by the global max_tcp_buff_mem setting. Value type: unsigned integer Default value: "0"

Key	Description
http_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. Value type: enumeration Default value: "lazy" Permitted values: lazy: lazy eager: eager
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). Requires: ssl_decrypt is set to "Yes" Value type: list 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. Value type: unsigned integer Default value: "1"
keepalive	Whether or not the virtual server should use keepalive connections with the remote clients. Value type: Yes / No Default value: "Yes"

Key	Description
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. Value type: seconds Default value: "10"
kerberos_protocol_transition!enabled	Whether or not the virtual server should use Kerberos Protocol Transition. Value type: Yes / No Default value: "No"
kerberos_protocol_transition!principal	The Kerberos principal this virtual server should use to perform Kerberos Protocol Transition. Value type: string Default value: <none>
kerberos_protocol_transition!target	The Kerberos principal name of the service this virtual server targets. Value type: string Default value: <none>
location!regex	If the 'Location' header matches this regular expression, rewrite the header using the 'location!replace' pattern: Value 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): Requires: location!regex is set to a regular expression Value 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. Value type: enumeration Default value: "1" Permitted values: 0: Nothing;

Key	Description
	2: Rewrite the hostname to the request's "Host" header, and rewrite the protocol and port if necessary; 1: Do not rewrite the hostname. Rewrite the protocol and port if the hostname matches the request's "Host" header.
log!client_connection_failures	Should the virtual server log failures occurring on connections to clients. Value type: Yes / No Default value: "No"
log!enabled	Whether or not to log connections to the virtual server to a disk on the file system. Value type: Yes / No Default value: "No"
log!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. Requires: log!enabled is set to "Yes" Value type: string Default value: "%zeushome%/zxtm/log/%v.log"
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. Requires: log!enabled is set to "Yes" Value type: string Default value: "%h %l %u %t "%r" %s %b "%{Referer}i" "%{User-agent}i""
log!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(). Value type: Yes / No Default value: "Yes"

Key	Description
log!server_connection_ failures	Should the virtual server log failures occurring on connections to nodes. Value type: Yes / No Default value: "No"
log!session_persistence_ verbose	Should the virtual server log session persistence events. Value type: Yes / No Default value: "No"
log!ssl_failures	Should the virtual server log failures occurring on SSL secure negotiation. Value type: Yes / No Default value: "No"
log!ssl_resumption_failures	Should the virtual server log messages when attempts to resume SSL sessions (either from the session cache or a session ticket) fail. Note that failure to resume an SSL session does not result in the SSL connection being closed, but it does cause a full SSL handshake to take place. Value type: Yes / No Default value: "No"
max_client_buffer	The amount of memory, in bytes, that the virtual server should use to store data sent by the client through one TCP connection or HTTP/2 stream. Larger values will use more memory, but will minimise the number of read() and write() system calls that the traffic manager must perform. Value type: bytes Default value: "65536"
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. Value type: unsigned integer

Key	Description
	Default value: "0"
max_server_buffer	The amount of memory, in bytes, that the virtual server should use to store data returned by the server through one TCP connection. Larger values will use more memory, but will minimise the number of read() and write() system calls that the traffic manager must perform. Value type: bytes Default value: "65536"
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. Value type: seconds Default value: "0"
mime!default	Auto-correct MIME types if the server sends the "default" MIME type for files. Value type: string Default value: "text/plain"
mime!detect	Auto-detect MIME types if the server does not provide them. Value type: Yes / No Default value: "No"
note	A description for the virtual server. Value type: string Default value: <none>
pool	The default pool to use for traffic. Value type: string Default value: <none>
port	The port on which to listen for incoming connections.

Key	Description
	Value type: unsigned integer Default value: <none>
private_key	The SSL private key. Requires: ssl_decrypt is set to "Yes" Value type: string Default value: <none>
protection	The service protection class that should be used to protect this server, if any. Value type: string Default value: <none>
protocol	The protocol that the virtual server is using. Value type: enumeration Default value: "http" Permitted values: http: HTTP ftp: FTP imapv2: IMAPv2 imapv3: IMAPv3 imapv4: IMAPv4 pop3: POP3 smtp: SMTP ldap: LDAP telnet: Telnet ssl: SSL https: SSL (HTTPS) imaps: SSL (IMAPS) pop3s: SSL (POP3S) ldaps: SSL (LDAPS) udpstreaming: UDP - Streaming udp: UDP dns: DNS (UDP) dns_tcp: DNS (TCP) sipudp: SIP (UDP) siptcp: SIP (TCP) rtsp: RTSP

Key	Description
	server_first: Generic server first client_first: Generic client first stream: Generic streaming
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. Value type: Yes / No Default value: "No"
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. Value type: Yes / No Default value: "No"
public_cert	The SSL public certificate. Requires: ssl_decrypt is set to "Yes" Value type: string Default value: <none>
recent_conns!enabled	Whether or not connections handled by this virtual server should be shown on the Activity > Connections page. Value type: Yes / No Default value: "Yes"
recent_conns!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. Value type: Yes / No Default value: "No"

Key	Description
request_client_cert	Whether or not the virtual server should request an identifying certificate from each client. Value type: enumeration Default value: "0" Permitted values: 0: Do not request a client certificate 1: Request, but do not require a client certificate 2: Require a client certificate
request_tracing!enabled	Record a trace of major connection processing events for each request and response. Value type: Yes / No Default value: "No"
request_tracing!trace_io	Include details of individual I/O events in request and response traces. Requires request tracing to be enabled. Requires: request_tracing!enabled is set to "Yes" Value type: Yes / No Default value: "No"
responserules	Rules to be applied to responses, in order, comma separated. Value type: list Default value: <none>
rtsp_streaming_portrange_high	If non-zero this controls the upper bound of the port range to use for streaming data connections. Value type: unsigned integer Default value: "0"
rtsp_streaming_portrange_low	If non-zero this controls the lower bound of the port range to use for streaming data connections. Value type: unsigned integer Default value: "0"
rtsp_streaming_timeout	If non-zero data-streams associated with RTSP connections will timeout if no data is transmitted for this many seconds. Value type: seconds Default value: "30"

Key	Description
rules	Rules to be applied to incoming requests, in order, comma separated. Value type: list Default value: <none>
serverfirst_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 " Value type: string Default value: <none>
sip_dangerous_requests	The action to take when a SIP request with body data arrives that should be routed to an external IP. Value type: enumeration Default value: "node" Permitted values: node: Send the request to a back-end node forbid: Send a 403 Forbidden response to the client forward: Forward the request to its target URI (dangerous)
sip_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. Value type: Yes / No Default value: "Yes"
sip_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. Value type: bytes Default value: "65536"
sip_mode	The mode that this SIP virtual server should operate in. Value type: enumeration Default value: "pi"

Key	Description
	Permitted values: lb: SIP Routing pi: SIP Gateway fc: Full Gateway
sip_rewrite_uri	Replace the Request-URI of SIP requests with the address of the selected back-end node. Value type: Yes / No Default value: "No"
sip_streaming_portrange_high	If non-zero this controls the upper bound of the port range to use for streaming data connections. Value type: unsigned integer Default value: "0"
sip_streaming_portrange_low	If non-zero, then this controls the lower bound of the port range to use for streaming data connections. Value type: unsigned integer Default value: "0"
sip_streaming_timeout	If non-zero a UDP stream will timeout when no data has been seen within this time. Value type: seconds Default value: "60"
sip_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. Value type: Yes / No Default value: "Yes"
sip_transaction_timeout	The virtual server should discard a SIP transaction when no further messages have been seen within this time. Value type: seconds Default value: "30"
sip_udp_associate_by_source	Require that SIP datagrams which are part of the same transaction are received from the same address and port. Value type: Yes / No Default value: "Yes"

Key	Description
slm	The service level monitoring class that this server should use, if any. Value type: string Default value: <none>
smtp!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. Value type: Yes / No Default value: "Yes"
so_nagle	Whether or not Nagle's algorithm should be used for TCP connections. Value type: Yes / No Default value: "No"
ssl_cipher_suites	The SSL/TLS cipher suites to allow for connections to this virtual server. Leaving this empty will make the virtual server use the globally configured cipher suites, see configuration key ssl!cipher_ suites in the Global Settings section of the System tab. See there for how to specify SSL/TLS cipher suites. Value type: string Default value: <none>
ssl_client_cert_headers	What HTTP headers the virtual server should add to each request to show the data in the client certificate. Value type: enumeration Default value: "none" Permitted values: none: No data simple: Certificate fields all: Certificate fields and certificate text
ssl_decrypt	Whether or not the virtual server should decrypt incoming SSL traffic. Value type: Yes / No Default value: "No"

Key	Description
ssl_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 preference list, ssl!elliptic_curves in the Global Settings section of the System tab. See there for how to specify elliptic curves. Value type: string Default value: <none>
ssl_headers	Whether or not the virtual server should add HTTP headers to each request to show the SSL connection parameters. Value type: Yes / No Default value: "No"
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. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for Fallback SCSV enabled: Enable Fallback SCSV disabled: Disable Fallback SCSV
ssl_ocsp!issuer!*!aia	Whether or not the traffic manager should use AIA information contained in a client certificate to determine which OCSP responder to contact. Value type: Yes / No Default value: <none>
ssl_ocsp!issuer!*!nonce	Use the OCSP nonce extension, which protects against OCSP replay attacks. Some OCSP servers do not support nonces. Value type: enumeration Default value: <none> 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

Key	Description
ssl_ocsp!issuer!*!required	Should we do an OCSP check for this issuer, and is it required or optional. Value type: enumeration Default value: <none> Permitted values: none: None optional: OCSP check optional strict: OCSP check required
ssl_ocsp!issuer!*!responder_cert	The expected responder certificate. Value type: string Default value: <none>
ssl_ocsp!issuer!*!signer	If set the request will be signed with the supplied certificate. Value type: string Default value: <none>
ssl_ocsp!issuer!*!url	Which OCSP responders this virtual server should use to verify client certificates. Value type: string Default value: <none>
ssl_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. Value type: seconds Default value: "0"
ssl_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. Value type: Yes / No Default value: "No"
ssl_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.

Key	Description
	Value type: seconds Default value: "30"
ssl_ocsp_timeout	The number of seconds after which OCSP requests will be timed out. Value type: seconds Default value: "10"
ssl_send_close_alerts	Whether or not to send an SSL/TLS "close alert" when the traffic manager is initiating an SSL socket disconnection. Value type: Yes / No Default value: "Yes"
ssl_session_cache_enabled	Whether or not use of the session cache is enabled for this virtual server. Choosing the global setting means the value of configuration key ssl!session_cache_enabled from the Global Settings section of the System tab will be enforced. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for use of the session cache enabled: Enable use of the session cache disabled: Disable use of the session cache
ssl_session_tickets_enabled	Whether or not use of session tickets is enabled for this virtual server. Choosing the global setting means the value of configuration key ssl!tickets!enabled from the Global Settings section of the System tab will be enforced. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for use of session tickets enabled: Enable use of the session tickets disabled: Disable use of the session tickets

Key	Description
ssl_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, ssl!signature_algorithms in the Global Settings section of the System tab. See there for how to specify TLS signature algorithms. Value type: string Default value: <none>
ssl_sites!*!alt_certificates	The SSL public certificates for a particular destination site IP or SNI hostname. The specific site replaces the * (asterisk) in the key name, the value must be a valid file name in the conf/ssl/server_keys directory without the private or public file name extensions. The key can be specified multiple times to cover multiple IP addresses. Requires: ssl_decrypt is set to "Yes" Value type: list Default value: <none>
ssl_sites!*!private_key	The SSL private key for a particular destination site IP. Requires: ssl_decrypt is set to "Yes" Value type: string Default value: <none>
ssl_sites!*!public_cert	The SSL public certificate for a particular destination site IP or SNI hostname. The specific site replaces the * (asterisk) in the key name, the value must be a valid certificate in the conf/ssl/server_keys directory. The key can be specified multiple times to cover multiple IP addresses. Requires: ssl_decrypt is set to "Yes" Value type: string Default value: <none>
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. Value type: enumeration Default value: "use_default"

Key	Description
	Permitted values: use_default: Use the global setting for SSLv3 enabled: Enable SSLv3 disabled: Disable 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. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for TLSv1.0 enabled: Enable TLSv1.0 disabled: Disable 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. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for TLSv1.1 enabled: Enable TLSv1.1 disabled: Disable TLSv1.1
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. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for TLSv1.2 enabled: Enable TLSv1.2 disabled: Disable TLSv1.2

Key	Description
ssl_support_tls1_3	Whether or not TLSv1.3 is enabled for this virtual server. Choosing the global setting means the value of configuration key ssl!support_tls1_3 from the Global Settings section of the System tab will be enforced. Value type: enumeration Default value: "use_default" Permitted values: use_default: Use the global setting for TLSv1.3 enabled: Enable TLSv1.3 disabled: Disable TLSv1.3
ssl_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. Value type: Yes / No Default value: "No"
ssl_use_ocsp	Whether or not the traffic manager should use OCSP to check the revocation status of client certificates. Value type: Yes / No Default value: "No"
strip_x_forwarded_proto	Whether or not the virtual server should strip the 'X-Forwarded-Proto' header from incoming requests. Value type: Yes / No Default value: "Yes"
syslog!enabled	Whether or not to log connections to the virtual server to a remote syslog host. Value type: Yes / No Default value: "No"
syslog!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. Requires: syslog!enabled is set to "Yes" Value type: string

Key	Description
	Default value: "%h %l %u %t "%r" %s %b "%{Referer}i" "%{User-agent}i""
syslog!ipendpoint	The remote host and port (default is 514) to send request log lines to. Requires: syslog!enabled is set to "Yes" Value type: string 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. Requires: syslog!enabled is set to "Yes" Value type: unsigned integer Default value: "2048"
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. Note that the default value may vary depending on the protocol selected. Value type: seconds Default value: "300"
transaction_export!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. Requires: transaction_export!enabled is set to "Yes" Value type: Yes / No Default value: "No"
transaction_export!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. Value type: Yes / No Default value: "Yes"

Key	Description
transaction_export!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. Value type: Yes / No Default value: "No"
transaction_export!http_ header_blacklist	The set of HTTP header names for which corresponding values should be redacted from the metadata exported by this virtual server. Value type: list Default value: "Authorization"
transparent	Whether or not bound sockets should be configured for transparent proxying. Value type: Yes / No Default value: "No"
udp_endpoint_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. Value type: Yes / No Default value: "Yes"
udp_port_smp	Whether or not UDP datagrams should be distributed across all traffic manager processes, if this behaviour is not normally selected automatically due to other settings. Value type: Yes / No Default value: "No"
udp_rbuff_size	If this setting is non-zero, the virtual server will set the socket receive buffer size to this number of bytes. If set, this will override the so_rbuff_size setting. An OS-specified limit on socket buffer sizes such as given by sysctl net.core.rmem_max can be exceeded using this setting. Value type: unsigned integer

Key	Description
	Default value: "0"
udp_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 udp_timeout is reached. Value type: int Default value: "1"
udp_smp_mode	Whether the traffic manager should try to use SO_REUSEPORT for distributing incoming UDP datagrams across multiple processes (if kernel support is detected) or whether the legacy (pre-20.2) multi-processing mode should be used. Value type: enumeration Default value: "auto" Permitted values: auto: auto legacy: legacy
udp_timeout	The virtual server should discard any UDP connection and reclaim resources when no further UDP traffic has been seen within this time. Value type: seconds Default value: "7"
udp_wbuff_size	If this setting is non-zero, the virtual server will set the socket send buffer size to this number of bytes. If set, this will override the so_wbuff_size setting. An OS-specified limit on socket buffer sizes such as given by sysctl net.core.wmem_max can be exceeded using this setting. Value type: unsigned integer Default value: "0"
webcache!control_out	The "Cache-Control" header to add to every cached HTTP response, no-cache or max-age=600 for example. Value type: string Default value: <none>

Key	Description
webcache!enabled	If set to Yes the traffic manager will attempt to cache web server responses. Value type: Yes / No Default value: "No"
webcache!errorpage_time	Time period to cache error pages for. Value type: seconds Default value: "30"
webcache!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. Value type: seconds Default value: "2"
webcache!time	Maximum time period to cache web pages for. Value type: seconds Default value: "600"

conf/zeusafm.conf

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 Pulse Secure 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.

Key	Description
decisionServerPort	The port to which the Enforcer rule should send traffic so it can be distributed between the decider processes. Value type: unsigned integer Default value: "8100"

Key	Description
updaterPort	The Application Firewall Updater Slave Port, this port is used on all IP addresses. Value type: unsigned integer Default value: "8092"

conf/zxtms

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.

Key	Description
admin!hsts_enable	Whether or not HSTS (RFC 6797) is enabled for admin server connections. Value type: Yes / No Default value: "No"
admin!hsts_max_age	The number of seconds that the HSTS header field max-age will be set to Value type: unsigned integer Default value: "31536000"
adminMasterXMLIP	The Application Firewall master XML IP. Value type: string Default value: "0.0.0.0"
adminMasterXMLPort	The Application Firewall XML Master port, this port is used on all IP addresses. Value type: unsigned integer Default value: "0"
adminServerPort	The Application Firewall Administration Server port, this port is only open on localhost.

Key	Description
	Value type: unsigned integer Default value: "0"
adminSlaveXMLIP	The Application Firewall slave XML IP. Value type: string Default value: "0.0.0.0"
adminSlaveXMLPort	The Application Firewall XML Slave port, this port is used on all IP addresses. Value type: unsigned integer Default value: "0"
aod-magic-fixed-decider-base-port	The base port from which the Application Firewall decider processes should run. Ports will be used sequentially above this for each additional decider process that runs. Value type: unsigned integer Default value: "0"
appliance!card!*!interfaces	The order of the interfaces of a network card Value type: list Default value: <none>
appliance!card!*!label	The labels of the installed network cards Value type: string Default value: <none>
appliance!disable_kpti	Whether the traffic manager appliance should run without kernel page table isolation (KPTI). KPTI provides protection to prevent unprivileged software from being potentially able to read arbitrary memory from the kernel (i.e. the Meltdown attack, CVE-2017-5754); however this protection incurs a general system performance penalty. If you are running trusted software on the appliance, and the trade-off between performance at the cost of 'defense in depth' favors the former in your deployment, you may wish to enable this configuration key. If you are unsure, it is recommended that you leave this key disabled, which is also the default. Value type: Yes / No Default value: "No"

Key	Description
appliance!dnscache	The DNS cache setting the appliance should use and place in /etc/systemd/resolved.conf. Value type: Yes / No Default value: "Yes"
appliance!dnssec	The DNSSEC setting the appliance should use and place in /etc/systemd/resolved.conf. Value type: enumeration Default value: "no" Permitted values: yes: DNSSEC enabled no: DNSSEC disabled allow_downgrade: Use DNSSEC when available
appliance!gateway	The default gateway. Value type: string Default value: <none>
appliance!gateway6	The default IPv6 gateway. Value type: string Default value: <none>
appliance!hostname	Name (hostname.domainname) of the appliance. Value type: string Default value: <none>
appliance!hosts!*	Static host name entries to be placed in the /etc/hosts file. The * (asterisk) in the key name is the host name, the value is the IP address. Value type: string Default value: <none>
appliance!if!*!autoneg	Enable or disable auto-negotiation for an interface, the interface name is used in place of the * (asterisk). Value type: Yes / No Default value: <none>
appliance!if!*!duplex	Enable or disable full-duplex for an interface, the interface name is used in place of the * (asterisk). Value type: Yes / No

Key	Description
	Default value: <none>
appliance!if!*!mode	Set the configuriation mode of an interface, the interface name is used in place of the * (asterisk). Value type: enumeration Default value: <none> Permitted values: Static: Static DHCP: DHCP
appliance!if!*!mtu	Set the maximum transmission unit (MTU) of the interface. Value type: unsigned integer Default value: <none>
appliance!if!*!speed	Set the speed of an interface, the interface name is used in place of the * (asterisk). Value type: enumeration Default value: <none> Permitted values: 10: 10Mbs 100: 100Mbs 1000: 1Gbs 10000: 10Gbs 40000: 40Gbs 100000: 100Gbs
appliance!ip!*!addr	Set the IP address for the interface, the interface name is used in place of the * (asterisk). Value type: string Default value: <none>
appliance!ip!*!isexternal	Set whether or not an interface is externally or internally facing, the interface name is used in place of the * (asterisk). Value type: Yes / No Default value: <none>
appliance!ip!*!mask	Set the IP mask (netmask) for an interface, the interface name is used in place of the * (asterisk). Value type: string

Key	Description
	Default value: <none>
appliance!ipmi!lan!access	Whether IPMI LAN access should be enabled or not. Value type: Yes / No Default value: "No"
appliance!ipmi!lan!addr	The IP address of the appliance IPMI LAN channel. Value type: string Default value: <none>
appliance!ipmi!lan!gateway	The default gateway of the IPMI LAN channel. Value type: string Default value: "0.0.0.0"
appliance!ipmi!lan!ipsrc	The addressing mode the IPMI LAN channel operates. Value type: enumeration Default value: "static" Permitted values: static: Static IP Address dhcp: Address obtained by DHCP
appliance!ipmi!lan!mask	Set the IP netmask for the IPMI LAN channel. Value type: string Default value: <none>
appliance!ipv4_forwarding	Whether or not IPv4 forwarding is enabled. Value type: Yes / No Default value: "No"
appliance!ipv6_forwarding	Whether or not IPv6 forwarding is enabled. Value type: Yes / No Default value: "No"
appliance!licence_agreed	Whether or not the license agreement has been accepted. This determines whether or not the Initial Configuration wizard is displayed. Value type: Yes / No Default value: "No"
appliance!manageazureroutes	Whether or not the software manages the Azure policy routing.

Key	Description
	Value type: Yes / No Default value: "Yes"
appliance!managebootloader	Whether or not the software manages the system bootloader's password Value type: Yes / No Default value: "Yes"
appliance!managecron	Whether or not the software manages the system's cronjobs to ensure they are running as the correct user. Value type: Yes / No Default value: "Yes"
appliance!manageec2conf	Whether or not the software manages the EC2 config. Value type: Yes / No Default value: "Yes"
appliance!managegateway	Whether or not the software manages the system's gateway configuration. Value type: Yes / No Default value: "Yes"
appliance!managegceroutes	Whether or not the software manages the GCE routing. Value type: Yes / No Default value: "Yes"
appliance!managehostname	Whether or not the software manages the system's hostname. Value type: Yes / No Default value: "Yes"
appliance!managehosts	Whether or not the software manages the system's /etc/hosts file. Value type: Yes / No Default value: "Yes"
appliance!manageif	Whether or not the software manages system's network interfaces. Value type: Yes / No Default value: "Yes"

Key	Description
appliance!manageip	Whether or not the software manages the system's IP addresses. Value type: Yes / No Default value: "Yes"
appliance!manageipmi	Whether or not the software manages the system's IPMI configuration. Value type: Yes / No Default value: "Yes"
appliance!manageiptrans	Whether or not the software manages the IP transparency Value type: Yes / No Default value: "Yes"
appliance!managenat	Whether or not the software manages the system's NAT configuration. Value type: Yes / No Default value: "Yes"
appliance!managentpservers	Whether or not the software manages which NTP servers the system uses. Value type: Yes / No Default value: "Yes"
appliance!managereservedports	Whether or not the software manages the system configuration for reserved ports Value type: Yes / No Default value: "Yes"
appliance!manageresolver	Whether or not the software manages the system's name resolution (i.e. the /etc/systemd/resolved.conf file). Value type: Yes / No Default value: "Yes"
appliance!managereturnpath	Whether or not the software manages return path routing. If disabled, the appliance won't modify iptables / rules / routes for this feature. Value type: Yes / No Default value: "Yes"

Key	Description
appliance!manageroute	Whether or not the software manages the system's routing tables. Value type: Yes / No Default value: "Yes"
appliance!manageservices	Whether or not the software manages the system services Value type: Yes / No Default value: "Yes"
appliance!managesnmp	Whether or not the software manages a system net-snmp service as a proxy to the internal SNMP service. Value type: Yes / No Default value: "Yes"
appliance!managessh	Whether or not the software manages the system's SSH server settings. Value type: Yes / No Default value: "Yes"
appliance!managetimezone	Whether or not the software manages the system's timezone setting. Value type: Yes / No Default value: "Yes"
appliance!manageusers	Whether or not the software manages system users. If enabled then users in the software's 'admin' group will be able to log into the system as a local 'admin' user with root privileges and the local 'root' user will have its password kept in sync with the software's 'admin' user. Value type: Yes / No Default value: "Yes"
appliance!managevpcconf	Whether or not the software manages the EC2-VPC secondary IPs. Value type: Yes / No Default value: "Yes"
appliance!nameservers	The IP addresses of the nameservers the appliance should use and place in /etc/systemd/resolved.conf. Value type: string

Key	Description
	Default value: <none>
appliance!ntpservers	The NTP servers the appliance should use to synchronize its clock. Value type: string Default value: "0.zeus.pool.ntp.org 1.zeus.pool.ntp.org 2.zeus.pool.ntp.org 3.zeus.pool.ntp.org"
appliance!routes!*!gw	One of the keys used to specify a route. The IP of the route destination is used in place of the * (asterisk) and the value is the gateway IP to configure for the route. See also appliance!routes!mask and appliance!routes!if. Value type: string Default value: <none>
appliance!routes!*!if	One of the keys used to specify a route. The IP of the route destination is used in place of the * (asterisk) and the value is the network interface to configure for the route. See also appliance!routes!mask and appliance!routes!gw. Value type: string Default value: <none>
appliance!routes!*!mask	One of the keys used to specify a route. The IP of the route destination is used in place of the * (asterisk) and the value is the netmask to apply to the IP. See also appliance!routes!gw and appliance!routes!if. Value type: string Default value: <none>
appliance!searchdomains	The search domains the appliance should use and place in /etc/systemd/resolved.conf. Value type: string Default value: <none>
appliance!ssh!enabled	Whether or not the SSH server is enabled on the appliance. Value type: Yes / No Default value: "Yes"
appliance!ssh!passwordallowed	Whether or not the SSH server allows password based login.

Key	Description
	Value type: Yes / No Default value: "Yes"
appliance!ssh!port	The port that the SSH server should listen on. Value type: unsigned integer Default value: "22"
appliance!timezone	The timezone the appliance should use. This must be a path to a timezone file that exists under /usr/share/zoneinfo/. Value type: string Default value: "US/Pacific"
appliance!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. Value type: list Default value: <none>
authenticationServerIP	The Application Firewall Authentication Server IP. Value type: string Default value: "0.0.0.0"
cloud_platform	Cloud platform where the traffic manager is running. Value type: string Default value: <none>
control!bindip	The IP address that the software should bind to for internal administration communications. See also controlport. 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 controlallow (in the conf/settings.cfg file) is compatible with the IP configured here. Value type: string

Key	Description
	Default value: "*"
control!canupdate	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 control!update set to Yes. If set to No then it will not be possible to log into the admin server for this instance. Value type: Yes / No Default value: "Yes"
controlport	The port that the software should listen on for internal administration communications. See also control!bindip. Value type: unsigned integer Default value: "9080"
decisionServerPortBase	The Application Firewall internal communication base port. The Application Firewall will use ports sequentially above this for internal communication. These ports are bound only to localhost. Value type: unsigned integer Default value: "10000"
ec2!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. Value type: list Default value: <none>
externalip	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. Value type: string Default value: <none>
flipper!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.

Key	Description
	Specifying 0.0.0.0 will stop the traffic manager routing software from running the BGP protocol. Value type: string Default value: <none>
flipper!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. Value type: string Default value: <none>
flipper!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. Value type: list Default value: "%gateway%"
gid	The group ID that the software's worker processes will run as. For example, on typical Linux installations this could be set to 65534 for the unprivileged "nobody" group. Value type: string Default value: <none>
iptables!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. Value type: Yes / No Default value: "Yes"

Key	Description
iptrans!fwmark	The netfilter forwarding mark to use for IP transparency rules Value type: unsigned integer Default value: "320"
iptrans!iptables_enabled	Whether IP transparency may be used via netfilter/iptables. This requires the iptables socket extension. Value type: Yes / No Default value: "Yes"
iptrans!routing_table	The special routing table ID to use for IP transparency rules Value type: unsigned integer Default value: "320"
java!port	The port the Java Extension handler process should listen on. This port will be bound for localhost communications only. Value type: unsigned integer Default value: "9060"
location	This is the location of the local traffic manager is in. Value type: string Default value: <none>
nameip	Replace Traffic Manager name with an IP address. Value type: string Default value: <none>
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. Value type: unsigned integer Default value: "0"
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. Value type: unsigned integer Default value: "0"
numberOfCPUs	The number of Application Firewall decider process to run.

Key	Description
	Value type: unsigned integer Default value: "0"
remote_licensing!email_address	The e-mail address sent as part of a remote licensing request. Value type: string Default value: <none>
remote_licensing!message	A free-text field sent as part of a remote licensing request. Value type: string Default value: <none>
rest!bindips	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. Value type: list Default value: "*"
rest!port	The port on which the REST API should listen for requests. Value type: unsigned integer Default value: "9070"
restServerPort	The Application Firewall REST Internal API port, this port should not be accessed directly Value type: unsigned integer Default value: "0"
snmp!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. Value type: list Default value: "all"
snmp!auth_password	The authentication password. Required (minimum length 8 bytes) if snmp!security_level includes authentication. Requires: snmp!security_level is set to "authNoPriv" Value type: password Default value: <none>

Key	Description
snmp!bindip	The IP address the SNMP service should bind its listen port to. The value * (asterisk) means SNMP will listen on all IP addresses. Value type: string Default value: "*"
snmp!community	The community string required for SNMPv1 and SNMPv2c commands. (If empty, all SNMPv1 and SNMPv2c commands will be rejected). Value type: string Default value: "public"
snmp!enabled	Whether or not the SNMP command responder service should be enabled on this traffic manager. Value type: Yes / No Default value: "No"
snmp!hash_alg	The hash algorithm for authenticated SNMPv3 communications. Requires: snmp!security_level is set to "authNoPriv" Value type: enumeration Default value: "md5" Permitted values: md5: MD5 sha1: SHA-1
snmp!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. Value type: string Default value: "default"
snmp!priv_password	The privacy password. Required (minimum length 8 bytes) if snmp!security_level includes privacy (message encryption). Requires: snmp!security_level is set to "authPriv" Value type: password Default value: <none>
snmp!security_level	The security level for SNMPv3 communications.

Key	Description
	Value type: enumeration Default value: "noAuthNoPriv" Permitted values: noAuthNoPriv: No Authentication, No Privacy authNoPriv: Authentication only authPriv: Authentication and Privacy
snmp!username	The username required for SNMPv3 commands. (If empty, all SNMPv3 commands will be rejected). Value type: string Default value: <none>
trafficip!*!networks	A configuration of networks keyed by interface, used by flipper to choose an interface to raise a Traffic IP on. Value type: list Default value: <none>
uid	The user ID that the software's worker processes will run as. For example, on typical Linux installations this could be set to 65534 for the unprivileged "nobody" user. Value type: string Default value: <none>
updateControlCenterPort	The Application Firewall Updater GUI Backend Port, this port is used on localhost only. Value type: unsigned integer Default value: "0"
updateExternControlCenterPort	The Application Firewall Updater External Control Center Port, this port is used on localhost only. Value type: unsigned integer Default value: "8091"
updateGUIServerPort	The Application Firewall Updater GUI Server Port, this port is used on localhost only. Value type: unsigned integer Default value: "0"
updaterIP	The Application Firewall Updater IP. Value type: string

Key	Description
	Default value: "0.0.0.0"