BROCADE

Brocade Services Director Advanced User Guide, 17.2

Supporting Brocade Services Director 17.2

Contents

Preface

Document conventions

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

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential

hazards.

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

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.

DANGER
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Text formatting conventions

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

Format	Description
bold text	Identifies command names.
	Identifies keywords and operands.
	Identifies the names of GUI elements.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.
`Courier font`	Identifies CLI output.

Format	Description
	Identifies command syntax examples.

Command syntax conventions

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

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

Brocade resources

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

White papers, data sheets, and the most recent versions of Brocade software and hardware manuals are available at www.brocade.com. Product documentation for all supported releases is available to registered users at MyBrocade.

Click the **Support** tab and select **Document Library** to access product documentation on MyBrocade or www.brocade.com. You can locate documentation by product or by operating system.

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

Document feedback

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

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

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

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online or by telephone. Brocade OEM customers should contact their OEM/solution provider.

Brocade customers

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

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

Online	Telephone

Preferred method of contact for non-urgent issues:
   •   Case management through the MyBrocade portal.
   •   Quick Access links to Knowledge Base, Community, Document Library, Software Downloads and Licensing tools
	Required for Sev 1-Critical and Sev 2-High issues:
	   • Continental US: 1-800-752-8061
	   • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33)
	   • Toll-free numbers are available in many countries.
	   • For areas unable to access a toll-free number: +1-408-333-6061

Brocade OEM customers

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

- OEM/solution providers are trained and certified by Brocade to support Brocade® products.

- Brocade provides backline support for issues that cannot be resolved by the OEM/solution provider.

- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.

- For questions regarding service levels and response times, contact your OEM/solution provider.

Understanding the Services Director

Introduction

The Services Director enables you to deploy, configure and manage your Traffic Manager instances. Specifically, you can:

- license your Traffic Manager instances.
- deploy Traffic Manager instance hosts.
- deploy Traffic Manager instances using an instance host.
- deploy cloud-based Traffic Manager instances on AWS.
- register externally-deployed Traffic Manager instances.
- transition deployed Traffic Manager instances through a lifecycle.
- start, stop and restart your Services Director service.
- protect your instance configurations (on a cluster basis) by taking automated and manual backups.
- protect your Services Director configuration using a backup system.
- protect your Traffic Manager passwords using encryption based on a Master Password.
- perform health and monitoring reporting.
- perform usage metering.
- generate system logs and system dumps.

  NOTE
  Support for individual functions depends on your license type, see Overview of Services Director Licensing on page 19.

Services Director Customer Types

The Services Director supports two kinds of customers.

Cloud Service Provider (CSP) customers are typically service providers and hosting organizations who want to use metered usage data to bill their end users. They can offer unlimited registration, deployment and licensing of Traffic Manager instances, and there are no limits on the bandwidth or Services Director features they can consume. Services Director collects metered usage data on a regular basis, and accumulates this for eventual download and end user billing. The same metering data is sent to Brocade to bill the CSP customer.

Enterprise customers are typically end users of Services Director who want to pre-pay all usage charges. They can allocate features and bandwidth up to the capacity they have purchased as Bandwidth Packs and Add-On Packs. At this point, additional Bandwidth Packs and Feature Packs can be purchased from Brocade to allow further allocation. No metering is supported, and Brocade does not bill the Enterprise customer.

See Overview of Services Director Licensing on page 19 for a description of the software license types that support these customers.

NOTE

In this manual, a section that is relevant to only one customer license type will state this in its heading. For example:

- Bandwidth Pack Licenses (Enterprise Customers Only) on page 20.
- Usage Metering and Activity Metrics (CSP Customers Only) on page 185 .

Services Director Form-Factors

The Services Director is available as two form-factors:

- *Software form-factor* – this is an installation of Brocade Services Director on Ubuntu or RHEL/CentOS. This can be controlled using a Representational State Transfer (REST) API only.

  This is the primary configuration described in this manual.

- *Virtual appliance form-factor* – this is an installation of Brocade Services Director Virtual Appliance (Services Director VA) on any supported platform. This can be accessed using a REST API, a graphical user interface (GUI) and a Command-Line Interface (CLI).

  This is NOT the focus of this manual.

  NOTE

  For detailed information about the Services Director VA, see *Brocade Services Director Getting Started Guide.*

  NOTE

  For detailed information about the CLI, see the *Brocade Services Director Command Reference.*

FIGURE 1 Accessing the Services Director

The Services Director stores information about deployed Traffic Manager instances, including the resources that it needs to manage in a MySQL inventory database.

Supported actions are triggered through the Services Director REST API, through a REST API **instance** resource. You issue a REST API request for the Services Director server to update the inventory database, which in turn queues an action to implement the requested operation before responding to the request. To avoid time-out issues, the response is returned before the action completes. After the action completes, the inventory database is updated. There is no progress callback from the Services Director; you must poll the Services Director to check for the status of the action. For detailed information about REST API resources, see Using the REST API with the Services Director on page 123.

Instance Host Form-Factors

Instance hosts are required for the deployment and management of Traffic Managers by the Services Director. There are two supported *instance host form-factors:*

- *The instance host is a configured Linux machine* - this is a generic Linux machine that has been configured to run as an instance host. This machine can be either virtual or physical, and can run either Ubuntu or RHEL/CentOS.
- *The instance host is a configured virtual appliance* - this instance host is created as a virtual machine from vSphere using an image file that is provided with Services Director.

Support for combinations of the instance host form-factors and Services Director form-factors is as follows:

- A software form-factor Services Director supports the use of a Linux machine as an instance host. This is described in Using an Instance Host Machine with a Software Services Director on page 47.
- A software form-factor Services Director *does not support* the use of an instance host virtual appliance.
- The Services Director VA supports the use of a configured Linux machine as an instance host. This is described in Using an Instance Host Machine with a Services Director VA on page 97.
- The Services Director VA supports the use of an instance host virtual appliance. This is described in Using an Instance Host VA with a Services Director VA on page 67

FIGURE 2 Supported Combinations of Form-Factors

Instance Host Form-Factors

		Linux	Virtual Appliance
Services Director Form-Factors	Software	Supported	Not Supported
	Virtual Appliance	Supported	Supported

> **NOTE**
> For detailed information about the Services Director VA, see *Brocade Services Director Getting Started Guide*.

> **NOTE**
> For detailed information about the CLI, see the *Brocade Services Director Command Reference*.

The Services Director stores information about deployed Traffic Manager instances, including the resources that it needs to manage in a MySQL inventory database.

Supported actions are triggered through the Services Director REST API, through a REST API **instance** resource. You issue a REST API request for the Services Director server to update the inventory database, which in turn queues an action to implement the requested operation before responding to the request. To avoid time-out issues, the response is returned before the action completes. After the action completes, the inventory database is updated. There is no progress callback from the Services Director; you must poll the Services Director to check for the status of the action. For detailed information about REST API resources, see Using the REST API with the Services Director on page 123.

Using the Advanced User Guide

This manual guides you through the installation and configuration of your software form-factor Services Director. It provides a reference guide for specific functional areas.

The structure of the manual is as follows:

- Managing Services Director Licensing on page 19 – This chapter describes Services Director licenses and how to install them.

- Installing and Configuring the Software Form-Factor Services Director on page 29 – This chapter describes how to install and configure the Services Director.

- Using an Instance Host Machine with a Software Services Director on page 47 – This chapter describes how to deploy Traffic Manager instances with a combination of an instance host machine and the software form-factor Services Director.

- Using an Instance Host VA with a Services Director VA on page 67 – This chapter describes how to deploy Traffic Manager instances with a combination of an instance host VA and the Services Director VA.

- Using an Instance Host Machine with a Services Director VA on page 97 – This chapter describes how to deploy Traffic Manager instances with a combination of an instance host machine and the Services Director VA.

- Registering Externally-Deployed Traffic Managers on page 117 – This chapter describes how to register externally-deployed Traffic Manager instances when using an instance host machine on either the software form-factor Services Director or the Services Director VA.

- Using the REST API with the Services Director on page 123 – This chapter describes the REST API, which is the primary means of communicating with the Services Director for all configuration and control purposes.

- Metering and Monitoring the Services Director on page 185 – This chapter describes how to configure metering and monitoring for the Services Director.

- Upgrading Services Director VAs and Instance Host VAs on page 191 – This chapter describes the process for upgrading the Services Director VA and Instance Host VAs. It also includes procedures for working with the master password.

- Working with the Master Password on page 201 – This chapter describes how to store, change and reset the Master Password.

- Appendix: Deploying for Redundancy on page 205 – This appendix describes how to implement high availability configurations for the Services Director.

- Appendix: Managing the Services Director Using the CLI on page 219 – This appendix describes the configuration of the Services Director using the Command-Line Interface (CLI) only.

- Appendix: Email Notifications Generated By Services Director on page 231 – This appendix describes the e-mails that can be generated by the Services Director.

Using INSTALLROOT in This Guide

This guide uses the term `INSTALLROOT` to refer to the location of the Services Director software installation directory. It is not an environment variable and is used in this guide for consistency only.

Previous versions of the Services Director used a deprecated environment variable, **$SSCHOME** , for this purpose. If this is still set in your environment, you must explicitly unset it prior to installing or upgrading the Services Director.

The default installation locations for the Services Director software package on the supported platforms are:

- Ubuntu: /opt/riverbed_ssc_17.2/
- RHEL/CentOS: /opt/riverbed-ssc/

Managing Services Director Licensing

Overview of Services Director Licensing

The Services Director requires the following licenses, depending on your type of deployment and features:

* Services Director Software Licenses - The following Services Director licenses differ in how they license Traffic Manager instances.

    – Cloud Services Provider (CSP) license - A CSP license supports all Services Director features, and includes billing that is dependent on instance metering. Registration, deployment and licensing of any number of Traffic Manager instances is supported, and there are no limits on the features or bandwidth they can use. Using a CSP license, the Services Director implements a metering scheme to obtain throughput and other metrics from each Traffic Manager instance on a regular basis (typically, hourly) and records the data in central log files.

      Service providers and hosting organizations use the metrics data to bill end users accordingly. Brocade uses the same metrics data to charge the Services Director customer.

    – Enterprise license – An Enterprise license supports all pre-paid bandwidth and features. That is, the customer purchases Bandwidth Pack licenses and Add-On Pack licenses from Brocade for allocation to Traffic Managers. Bandwidth and features can be allocated up to the capacity purchased. Additional Bandwidth Packs and Feature Packs can be purchased from Brocade to allow further allocation. An Enterprise license does not provide a billing option. No metering data is collected and returned to Brocade.

      Bandwidth Pack – A Bandwidth Pack license is a secondary type of license for Enterprise customers. Each Bandwidth Pack license associates a specific amount of bandwidth (typically, 5 Gbps) with a Traffic Manager SKU. Each Bandwidth Pack is tied to a specific Services Director Enterprise license.

      Add-On – An Add-On license is a secondary type of license that provides a mechanism for adding feature specific licenses: for example, Federal Information Processing Standards (FIPS), Brocade Application Firewall (WAF), and Brocade Aptimizer Web Accelerator. Each Add-On license is tied to a specific Enterprise License Key.

* Traffic Manager Flexible Licensing Architecture (FLA) License - A Traffic Manager FLA license is intended for the Traffic Manager instances rather than Services Director itself. With the FLA license you do not have to obtain licenses for individual Traffic Manager instances. Instead, the Services Director applies a site-specific license to each instance and dynamically sets the feature set (SKU) and bandwidth desired for each instance. The FLA license requires a self-signed (or equivalent) certificate to be generated prior to its generation.

    – From Services Director v2.2 onwards, *Universal FLA Licenses* are provided with the installed product for both CSP and Enterprise customers. These are suitable for any Traffic Manager at version 10.1 or above which has its REST API enabled.

    – From Services Director v2.2 onwards, the term *Legacy FLA Licenses* refer to FLA licenses for any Traffic Manager at version 10.0 or earlier, or which has its REST API disabled. That is, it is not suitable for use with Universal FLA Licensing. Legacy FLA licensing can be added for older Traffic Manager versions.

To Retrieve Brocade vTM and Services Director Product Licenses

1. License tokens are automatically emailed to you when you order your product. If you have not received your tokens, contact Brocade support.

2. Redeem license tokens at Brocade's website. To redeem tokens you must have a support site login and password. You can register for a new account at Brocade Support.

3. Licenses are emailed to you as attachments.

Licensing on a Cluster of Services Directors

You can use a cluster of Services Directors (two or more) to provide resilience to your system.

Before starting the first Services Director in a cluster of Services Directors, you must place a valid license key file in the `INSTALLROOT/licenses` directory. New licenses in this location are automatically added to the database when the Services Director reboots and communication is established. A newly-installed Services Director that is configured to use an existing inventory database containing valid license keys will use those keys to operate after it has started.

Brocade recommends that you install all subsequent Services Director licenses via the REST API's **controller_license_key** resource. For details, see controller_license_key Resource on page 141.

For detailed information about installing Services Director licenses, see Installing the Services Director Software License on page 37.

Bandwidth Pack Licenses (Enterprise Customers Only)

A Bandwidth Pack is a secondary type of license that is tied to a specific Enterprise Services Director license.

Brocade recommends that you add the Bandwidth Pack license to the shared database from a running Services Director using the REST API's **bandwidth_pack_license_key** resource. For details, see "bandwidth_pack_license_key Resource".

However, you can also place the Bandwidth Pack license in the `INSTALLROOT/licenses` directory and restart the Services Director. New licenses are automatically added to the database as they are discovered.

Each Bandwidth Pack enables a specific amount of bandwidth (typically, 5 Gbps) to a Traffic Manager SKU. The Bandwidth Pack license allows you to deploy and license Traffic Manager instances with an aggregate bandwidth allowance equal to that of the Bandwidth Pack.

Each Bandwidth Pack is associated with one Services Director license and cannot be used unless that Services Director license has been loaded and found to be valid. A Bandwidth Pack only allows the deployment and licensing of Traffic Manager instances with one SKU. If you want to deploy Traffic Manager instances with different SKUs, then they require multiple Bandwidth Packs.

Services Director licenses and Bandwidth Packs are perpetual or they can have start and end dates.

Multiple Bandwidth Packs can license bandwidth for a SKU; their allowances are added to determine the total. Bandwidth Packs can be upgraded from one SKU to another. Where an existing Bandwidth Pack license is upgraded, a new license is issued with the same serial number as the existing one, but licensing a different SKU. Only one of a set of Bandwidth Pack licenses with a shared serial number is used at any one time.

Where licensed capacity is exceeded for a given SKU, all licensing requests for instances using that SKU are rejected. This behavior is also true of instances using an Add-On license SKU with insufficient licensed bandwidth.

If you are using a Bandwidth Pack license, the Services Director does not allow you to exceed the licensed bandwidth with your deployed Traffic Manager instances. This behavior includes failed instances (with a status of Failed to Deploy), but not instances scheduled to be

deleted. Only instances that have been deleted are excluded; instances scheduled to be deleted still count toward deployed totals. Any instance whose status is not Deleted continues to consume licensing bandwidth. The same rules apply to the consumption for Add-On license SKUs bandwidth for instances using Add-On SKUs.

Installing the Bandwidth Pack License (Enterprise Customers Only)

To Install the Bandwidth Pack License

1. Place the license on an accessible location in your infrastructure. For details about obtaining your license keys, see To Retrieve Brocade vTM and Services Director Product Licenses on page 20.

2. Install and configure the Services Director. For details, see Installing and Configuring the Software Form-Factor Services Director on page 29.

3. Brocade recommends that you add the Bandwidth Pack license to the shared database from a running Services Director using the REST API's **bandwidth_pack_license_key** resource. For details, see "bandwidth_pack_license_key Resource".

   Alternatively, copy a Services Director license file containing the license key to the `INSTALLROOT/licenses` directory of an Services Director and restart the Services Director. New licenses are automatically added to the database as they are discovered.

Upgrading Bandwidth Pack Licenses (Enterprise Customers Only)

When your Services Director is using the Enterprise Licensing model, you can upgrade a bandwidth pack to support a different STM SKU. This supports the replacement of existing purchased licensing with the same quantity of a more feature-rich STM SKU.

For example, a deployment of Traffic Manager instances is using the STM-300 STM SKU, and an upgrade to the STM-400 STM SKU is required.

For each existing bandwidth pack license key being upgraded, the Administrator will be provided with two new bandwidth pack licenses keys:

- The first license will contain the same serial number as the existing bandwidth pack license key.

  Only one of these licenses can contribute licensed bandwidth in your Services Director deployment at any time.

- The second bandwidth pack license key will be a time-limited key which provides extra bandwidth used during the switchover.

  This provides a workaround for the Services Director's protection against licensing compliance breaches.

To Upgrade Bandwidth Pack Licenses

1. Obtain the replacement (upgrade) license key and the supplementary temporary bandwidth pack license keys from Brocade.

2. Install replacement and supplementary bandwidth pack license keys on the Services Director.

   It may be necessary to set the upgrade bandwidth pack license key(s) to an Active status after installation. Once complete, the **controller_license_key** resource's **cluster_bandwidth** property should show sufficient unused STM-400 bandwidth for the instances that are to be switched to use this STM SKU.

3. Create a **feature_pack** resource using the STM-400 STM SKU if one does not already exist.

4. Set the **feature_pack** property of each affected Traffic Manager**instance** resource to the desired STM-400 **feature_pack** resource (as created in step 3).

5. Remove the supplementary bandwidth pack license keys.

   If the Services Director does not allow removal of the supplementary license keys, it may indicate a licensing shortage. This situation may result in unlicensed Traffic Managers after these keys expire.

Add-On Licenses

An Add-On license is a secondary type of license that is tied to a specific Enterprise License Key. Each Add-On license contributes license bandwidth for a single specific feature, known as an Add-On SKU. These Add-On SKUs can be combined with base SKUs when the user creates a Feature Pack. For an instance set to use such a Feature Pack, the feature capabilities of the base SKU are augmented by those of the Add-On SKU.

Add-On SKUs can be used with CSP licensing model, and do not require the use of an Add-On license.

The Services Director supports the following Add-On licenses:

- Federal Information Processing Standards (STM-B-ADD-FIPS, STM-CSP-U-ADD-FIPS)
- Brocade Application Firewall license (STM-B-ADD-WAF, STM-CSP-U-ADD-WAF)
- Brocade Aptimizer Web Accelerator (STM-B-ADD-WEBACCEL, STM-CSP-U-ADD-WEBACCEL)

Add-On licenses have unique serial numbers and do not support upgrades.

Installing an Add-On License

To retrieve an Add-On license, you are sent a token via email to redeem at Brocade Support.

To Install an Add-On License

1. Place the licenses on an accessible location in your infrastructure. For details about obtaining your license keys, see To Retrieve Brocade vTM and Services Director Product Licenses on page 20.
2. Install and configure the Services Director. For details, see To Install an Add-On License.
3. Brocade recommends that you add the Add-On license to the shared database from a running Services Director using the REST API's **add_on_pack_license_key** resource. For details, see "add_on_pack_license_key Resource" .

   Alternatively, copy a Services Director license file containing the license key to the `INSTALLROOT/licenses` directory of an Services Director and restart the Services Director. New licenses are automatically added to the database as they are discovered.

Traffic Manager FLA Licenses

Traffic Manager Flexible Licensing Architecture (FLA) License - A Traffic Manager FLA license is intended for the Traffic Manager instances rather than Services Director itself. With the FLA license you do not have to obtain licenses for individual Traffic Manager instances. Instead, the Services Director applies a site-specific license to each instance and dynamically sets the feature set (SKU) and bandwidth desired for each instance. The FLA license requires a self-signed (or equivalent) certificate to be generated prior to its generation.

- From Services Director v2.2 onwards, *Universal FLA Licenses* are provided with the installed product for both CSP and Enterprise customers. These are suitable for any Traffic Manager at version 10.1 or above which has its REST API enabled.

Brocade Services Director Advanced User Guide, 17.2
53-1005006-01

- From Services Director v2.2 onwards, the term *Legacy FLA Licenses* refer to FLA licenses for any Traffic Manager at version 10.0 or earlier, or which has its REST API disabled. That is, it is not suitable for use with Universal FLA Licensing. Legacy FLA licensing can be added for older Traffic Manager versions.

Universal FLA is available automatically when the product is installed, but all of the procedures in this section are supported by both Universal FLA and Legacy FLA.

Generating a Self-Signed SSL Server Certificate

NOTE

This section is supported by both Universal FLA and Legacy FLA. Universal FLA is available automatically when the product is installed, but you must still supply an SSL certificate for it.

The Services Director is commonly deployed using self-signed certificate/key pairs, using the self-signed server certificate in the FLA license.

The following information is required to generate a Traffic Manager FLA license:

- A list of the fully-qualified host names that is used for Services Directors acting as license servers, along with port numbers.
- The SSL server certificate that is used by all of the Services Directors (different controllers are not permitted to use different certificates).

An FLA license attempts to contact each of the listed license servers and succeeds if they are contacted in turn, until it makes a successful connection or has attempted and failed to contact each one.

The SSL server certificate is verified by the FLA license. If an SSL server certificate does not match what is required by the FLA license, then that FLA license will not connect to the Services Director license servers. If this failure occurs, you may need to generate a new FLA license or correct the key/certificate used by the Services Director.

To Generate a Self-Signed SSL Server Certificate

At the Linux prompt, enter:

```
openssl req -x509 -nodes -newkey rsa:2048 -keyout key.pem -out cert.pem -days 3650
```

Parameter	Description
req	Specifies an X509 certificate signing request management.
-x509	Specifies a self-signed certificate rather than a certificate request.
-nodes	Specifies that the private key will not be encrypted (otherwise, the server needs a password to start).
-newkey rsa:2048	Generates a new certificate request and sets the key size.
-keyout key.pem	Sets the target for the new private key.
-out cert.pem	Sets the target for the certificate.
-days 3650	Specifies the duration of the certificate (default is 30 days). A longer period may be desirable as a fresh FLA license will need to be generated and then deployed to all STM instances when the certificate expires.

The FLA license does not accept composite certificates that include a server certificate along with other information or certificates created by **ssh-keygen**.

To Verify the SSL Certificate

1. At the prompt, enter:

   ```
   openssl x509 -in certificate.crt -noout
   ```

   This command succeeds silently for a valid certificate or report errors.

2. To verify signed certificates, at the system prompt, enter:

   ```
   openssl verify <certificate name>
   ```

Installing FLA Licenses

> **NOTE**
> Universal FLA is available automatically when the product is installed, but this section is supported by both Universal FLA licenses and Legacy FLA licenses.

The Traffic Manager FLA license is installed by placing the FLA license file in the configured sources directory (the location for the Traffic Manager image and FLA license files), and then creating a **license** resource via the Services Director REST API.

To Install a FLA License

1. Choose a source location for FLA licenses. This is used during the installation process for the Services Director. For detailed information, see Required Configuration Parameters on page 32.

2. Place the license file in the location chosen in step 1. For details about obtaining your license keys, see To Retrieve Brocade vTM and Services Director Product Licenses on page 20.

3. Install and configure the Services Director. For details, see Installing and Configuring the Software Form-Factor Services Director on page 29.

4. Create a REST API **license** resource in the Services Director REST API. For details, see license Resource on page 157.

Checking the Health of an FLA License Manually

> **NOTE**
> This section is intended for use with Legacy FLA Licenses, and applies to all customers. No equivalent actions will be required for Universal FLA Licenses.

The Services Director supports an FLA Health Checker. This tool enables you to manually test the licensing of all your resources against an FLA license. This enables you to identify any licensing problems with the FLA before any instances start using it.

You will typically run the FLA Health Checker immediately after creating the dependent resources for instance deployment. That is, host, license, feature pack, and version.

> **NOTE**
> The health of an FLA license is checked automatically under some circumstances. See Checking the Health of an FLA License Automatically on page 27.

You start the FLA health checker using the REST API. To do this, issue a GET REST API request for the **license** resource, including the URL parameter **status_check=true**.

The response from the GET request will depend on the success of this operation.

Initially, the **health_check_status** property indicates that the FLA health check has started in the background, and is in progress.

For example:

```
{ "generic_errors" : null,
  "health_check_results" : [  ],
  "health_check_status" : "In Progress",
  "info" : "",
  "last_health_check_time" : "<timestamp>",
  "status" : "Active"
}
```

You can then poll the URI with normal GET request until the health check completes.

> **NOTE**
> Only one FLA health check can be running at any time.

- When a health check completes successfully:

  - the **health_check_status** is set to Completed.
  - the **health_check_result** is Passed.
  - the **details** property is empty.

  For example:

  ```
  { "generic_errors" : null,
    "health_check_results" : [ {
          "details" : {  },
          "health_check_result" : "Passed",
          "instance_host" : "<instance_host>.com",
          "services_director_host" : "<sd_host>:<port>",
          "services_director_port" : <port>
        } ],
    "health_check_status" : "Completed",
    "info" : "",
    "last_health_check_time" : "<timestamp>",
     "status" : "Active"
  }
  ```

- When the health check completes with SSL connection errors:

  - the **health_check_status** is set to Completed.
  - the **health_check_result** is Failed.
  - the **details** property includes the SSL errors.

  For example:

  ```
  { "generic_errors" : null,
     "health_check_results" : [{
          "details" : {
             "ssl_errors" : {
                  "errors" : [{
                  "err_code" : 18,
                  "err_text" : "Services Director sent a self-signed
                                   certificate which cannot be trusted"
              }],
              "fla_certs" : [{
                 "common_name" : "<common_name>",
                 "issuer_common_name" : "<issuer_common_name>",
                 "not_after" : "20150529135614Z",
                 "not_before" : "20130529135614Z"
               }],
              "services_director_certs" : [{
                 "common_name" : "<common_name>",
                 "issuer_common_name" : "<issuer_common_name>",
                 "not_after" : "20240513142858Z",
                 "not_before" : "20140514142858Z"
               }]
          }},
         "health_check_result" : "Failed",
  ```

```
                    "instance_host" : "<instance_host>",
                    "services_director_host" : "<sd_host>:<port>",
                    "services_director_port" : <port>
                 },
                 { "details" : {
                      "ssl_errors" : {
                          "errors" : [{
                               "err_code" : 18,
                               "err_text" : "Services Director sent a self-signed
                                            certificate which cannot be trusted"
                              }],
                      "fla_certs" : [{
                          "common_name" : "",
                          "issuer_common_name" : "",
                          "not_after" : "20150529135614Z",
                          "not_before" : "20130529135614Z"
                          }],
                       "services_director_certs" : [{
                          "common_name" : "<common_name>",
                          "issuer_common_name" : "<issuer_common_name>",
                          "not_after" : "20240513142858Z",
                          "not_before" : "20140514142858Z"
                              } ]
                          } },
                   "health_check_result" : "Failed",
                   "instance_host" : "<instance_host>",
                   "services_director_host" : "<sd_host>:<port>",
                   "services_director_port" : <port>
                   }
               ],
           "health_check_status" : "Completed",
           "info" : "test",
           "last_health_check_time" : "<timestamp>",
           "status" : "Active"
       }
```

In the above response, the **ssl_errors** property included:

– the details of the errors (**errors**)
– the certificate embedded in the FLA (**fla_certs**).
– the details of the certificate sent by the Services Director (**services_director_certs**).

• When the health check completes with network related errors:

– the **health_check_status** is set to Completed.
– the **health_check_result** is Failed.
– the **details** property includes the network errors.

For example:

```
       { "generic_errors" : null,
         "health_check_results" : [{
             "details" : {
                 "network_errors" : "Failed to resolve SSC host '<sd_host>':
                                    Name or service not known"
             },
             "health_check_result" : "Failed",
             "instance_host" : "<instance_host>",
             "services_director_host" : "<sd_host>:<port>",
             "services_director_port" : <port>
         },
         { "details" : {
             "network_errors" : "Failed to resolve SSC host '<sd_host>':
                                Name or service not known"
             },
             "health_check_result" : "Failed",
             "instance_host" : "<instance_host>",
             "services_director_host" : "<sd_host>:<port>",
             "services_director_port" : <port>
```

```
        }
    ],
    "health_check_status" : "Completed",
    "info" : "test",
    "last_health_check_time" : "<timestamp>",
    "status" : "Active"
}
```

The **generic_errors** top level property specifies errors that may happen while carrying out FLA health checks, but which is not related to the actual FLA health check. For instance, if there are no active hosts to carry out the checks.

Checking the Health of an FLA License Automatically

> **NOTE**
> This section is intended for use with Legacy FLA Licenses, and applies to all customers. No equivalent actions will be required for Universal FLA Licenses.

The Services Director supports an automated FLA Health Checker. This tool tests the FLA license as part of the following activities:

- The deployment of a new Traffic Manager instance.

    - If the FLA license check fails, the deployment action status is Blocked, with a reason for the failure. The instance has a status of Failed to Deploy.
    - If the FLA license check succeeds, the deployment continues. Once this action completes, the instance has a status of Idle.

    > **NOTE**
    > When you create a service, the deployment of a new Traffic Manager service instance does not trigger the FLA Health Checker.

- Any attempt to transition a Blocked deployment action to a status of Waiting.

    - If the FLA license check fails, the deployment action remains Blocked, with a reason for the failure.
    - If the FLA license check succeeds, the deployment continues. Once this action completes, the instance has a status of Idle.

- Any change to the FLA license of a deployed Traffic Manager instance. When the new license is applied, the instance is checked against the new FLA license.

    - If the FLA license check fails, the update action for the instance is Blocked, with a reason for the failure. The status of the instance is unchanged.
    - If the FLA license check succeeds, the status of the instance is unchanged.

This enables you to prevent any FLA licensing problems before they occur.

When you perform a deployment manually through the REST API, you can specify a URL parameter **?override_fla_check=true** to prevent the automatic FLA license check before deployment. This is not supported from the Services Director VA.

You are able to disable the operation of the FLA Health Checker at the Services Director level. See Disabling the FLA Health Checker on page 27.

> **NOTE**
> You can check the health of an FLA license manually at any time. See Checking the Health of an FLA License Manually on page 24.

Disabling the FLA Health Checker

To disable the FLA Health Checker at the Services Director level, either:

- In the REST API, make a call to the **/api/tmcm/2.5/settings/fla_check** resource, with the following JSON object:

    ```
    {"fla_checker_enabled": False}
    ```

• In the Services Director VA, disable the check from the **System** > **Service Status** page.

Reapplying a FLA License

If you want to re-license a Traffic Manager using its assigned FLA license, use the REST API. For example:

```
curl -v -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -u
user:passwd https://x.x.x.x:8100/api/tmcm/2.5/instance/<instance_name>?relicense=true -d '{ }'
```

Installing and Configuring the Software Form-Factor Services Director

To install and configure a software form-factor Services Director, follow the instructions in the following sections:

- Prerequisites on page 29.
- Configuring the MySQL Database for the Services Director on page 33.
- Either refer to Installing and Configuring the Services Director on Ubuntu on page 35 or Installing and Configuring the Services Director on RHEL/CentOS on page 36.
- Installing the Services Director Software License on page 37.
- To install an externally-deployed Traffic Manager with minimal supporting resources, refer to Adding Resources Using the REST API on page 38.
- Starting and Stopping the Services Director on page 40.
- Accessing the Operating System Shell on page 41

To add user authentication to your Services Director (either vTM user authentication or Services Director user authentication), you will need to:

- Review the user authentication material in the *Brocade Services Director Getting Started Guide*.
- Create any required authenticators, refer to authenticator Resource on page 129.
- Create any required permission groups, refer to permission_group Resource on page 164.
- Create any required access profiles (vTM user authentication only), refer to access_profile Resource on page 126.

  NOTE
  If you want to install and configure a Services Director Virtual Appliance (VA), see the *Brocade Services Director Getting Started Guide*.

Prerequisites

The Services Director supports the following:

- Operating system: Ubuntu 14.04 (x86_64), RHEL/CentOS 6 (x86_64)

> **NOTE**
> Brocade recommends that the Services Director machines and the instance hosts all use either Ubuntu or RHEL/CentOS.

- Database: MySQL 5.5 (5.5.32 is recommended)

Required Linux Packages

Make sure the following Linux packages are installed before you start the installation process. The packages differ according to the operating system you are using.

Ubuntu 14.04 (64 bit)	RHEL/CentOS 6 (64 bit)
mysql-common	mysql
libmysqlclient18	glibc-devel (v2.12 or later)
mysql-client-5.5 and mysql-client-core-5.5	Python Argparse (required for instance hosts)
OpenSSL (latest version for OS)	OpenSSL (latest version for OS)
libldap (v2.4 or later)	openldap (v2.4 or later)

> **NOTE**
> The latest version of OpenSSL is essential for security purposes.

> **NOTE**
> You must install and configure all software, including the Services Director, as superuser or a user account with administrator or equivalent access permissions.

Hardware Requirements

The Services Director requires the following hardware.

Hardware	Requirement
CPU	Intel Xeon / AMD Opteron
Minimum Memory	2 GB
Minimum Disk Space	10 GB (plus additional disk space for metering logs depending on number of instances metered)

Software and License Requirements

To install the Services Director, you need the following software and licenses:

- MySQL Database Server - The Services Director uses a MySQL database to store inventory data and other items relating to the deployment of your Traffic Manager instances. You must install and configure MySQL on your required platform. This must be accessible to Services Director using the configured credentials at the host and port specified. Consult the MySQL documentation for your platform to do this.

- Mail Server - The Services Director uses email to alert you of certain error conditions (for example, running low on log disk space). You must set up and provide SMTP mail server connection details to the Services Director for correct operation. The Services Director does not support SMTP connections that require authentication.

- Services Director Licenses - Before starting the first Services Director in a cluster of Services Directors, you must place a valid license key file in the `INSTALLROOT/licenses` directory. New licenses are automatically added to the database as they are discovered. The Services Director license is automatically copied to the inventory database after communication is established.

A newly installed Services Director instance that is configured to use an existing inventory database containing valid license keys uses those keys to operate after it has started.

Brocade recommends that you install all subsequent Services Director licenses via the REST API's **controller_license_key** resource. For details, see "controller_license_key Resource".

Required Services Director Files

To install the Services Director, you need the following files:

* SSL Certificate/Key – The SSL certificate and key are used by the Services Director REST API to enable authentication of the Service Controller when it is being accessed by remote clients. An important client is the Traffic Manager FLA license, which requires that you generate a self-signed SSL certificate and key (or equivalent compatible certificate and key).

You must place these files on an accessible location in your infrastructure. For details, see Generating a Self-Signed SSL Server Certificate on page 23.

Required Traffic Manager Files

The Services Director requires a set of external Traffic Manager files to deploy Traffic Manager instances:

* Traffic Manager Image – The Traffic Manager image (that is, its tarball file) is required to create Traffic Manager instances in the Services Director.

  You must place this in the Source File Location that you provide when you install the Services Director.

* Traffic Manager FLA License – The Traffic Manager FLA license is required to create instances in the Services Director.

  You must place this in the Source File Location that you provide when you install the Services Director.

These external files are not part of the initial Services Director installation. You might require different versions of the Traffic Manager, and the license files are customized for each Services Director deployment in conjunction with your support provider.

You use the Services Director REST API to create an inventory of resources for the Traffic Manager image and its license files. The REST API does not import the actual files and does not, by default, verify that the files are present. For detailed information about checking the status of REST API resources, see Using the REST API to Check Status on page 182.

Services Director User Types

Within the Services Director, there are these types of users:

* MySQL Database User – When you create the inventory database required by the Services Director, you create a MySQL user that has access to the database. Brocade recommends you create a specific Services Director user rather than using the database root user. You supply the name and password of this user when you install the Services Director so that it can access the database that you have pre-created. These credentials are recorded in the Services Director configuration file. See Configuring the MySQL Database for the Services Director on page 33.

* The Services Director Linux User – The Services Director software is run under a Linux user account. By default, this is the root user.

* The Services Director REST API User – To make Services Director REST API requests you must specify credentials for an admin user. This user is stored within the Services Director inventory database. You can create additional Services Director users or change the password using the REST API.

* Traffic Manager Instance Host User – When the Services Director carries out actions on its designated instance hosts, it does so by means of passwordless SSH access. You must create these users and provide credentials so that the Services Director can

communicate with each instance host. Typically, you create this user as **root** , although you can specify in the REST API resource the name of the user for each host.

- Traffic Manager Admin and Service Users - see Users in Deployed Traffic Manager Instances on page 32 and Users in Externally-Deployed Traffic Manager Instances on page 32.

Users in Deployed Traffic Manager Instances

When you deploy a Traffic Manager instance, the Services Director creates two default user accounts within the instance:

- **admin** - The administrative user for this instance, with full access to all features and capabilities.
- **service** - The service user, with privileges restricted to those required for service creation and management. That is, no system wide privileges.

You can change the password for each of these user accounts by updating the relevant password property in the **instance** resource.

Brocade strongly recommends that you do not provide **admin** user credentials to end-tenant users. You can safely provide the **service** user credentials, provided that the privileges for this account have not been altered from their default settings.

Users in Externally-Deployed Traffic Manager Instances

When you register an Externally-deployed Traffic Manager instance with the Services Director, it is not mandatory for the user to supply service user credentials.

Brocade recommends that you specify a username and password for an existing **admin** user when deploying an externally-deployed instance. This **admin** user (and a specified **rest_address**) for the instance resource enables monitoring of the externally-deployed instance, and provides access to its REST API proxy.

Brocade strongly recommends that you do not provide **admin** user credentials to end-tenant users.

Required Configuration Parameters

The Services Director installation prompts you to provide values for parameters used in setting up the software. Make sure you have access to all required information before you start the installation process.

The following table lists the required parameters to install the Services Director.

Parameter	Description
Database Host	The hostname or IP address of the server running the MySQL database server.
Database Port	The port number of the server running the MySQL database server.
Database Name	The name of the MySQL database used as the inventory database.
Database User	The name of a MySQL user authorized to use the inventory database. This username, along with the corresponding password, is used internally by the Services Director to access the MySQL database.
Database User Password	The password of the MySQL user. This password is used internally by the Services Director to access the MySQL database.
API Server Port	The number of the port on which the Services Director listens for REST API requests.
SSL Certificate File	The full path and filename of the certificate file to use for HTTPS connections to the REST API.
SSL Private Key File	The full path and filename of the private key file to use for HTTPS connections to the REST API.

Parameter	Description
Client Request Thread Pool Size	The maximum number of threads used for Services Director REST requests. This parameter limits the number of possible simultaneous HTTP requests.
Action Thread Pool Size	The maximum number of threads used for actions such as deploying, starting, and stopping Traffic Manager instances. Brocade recommends that you set **Action Thread Pool Size** and **Monitor Thread Pool Size** such that the sum of both is no greater than the MySQL maximum connections limit. If you have other applications that query the same MySQL database, you must make allowances for these additional connection requirements in your calculations.
Monitor Thread Pool Size	The maximum number of threads used to monitor Traffic Manager instances and other Services Directors. If you experience warnings about overdue monitoring actions, increase this value. Brocade recommends that you set **Action Thread Pool Size** and **Monitor Thread Pool Size** such that the sum of both is no greater than the MySQL maximum connections limit. If you have other applications that query the same MySQL database, you must make allowances for these additional connection requirements in your calculations.
Source File Location	The name of a directory accessible by the Services Director in which the Traffic Manager image and license files are placed.
Log Output Location	The name of a directory accessible by the Services Director server in which log files and metering log files are placed.
Alert Message Email Addresses	A list of one or more email addresses to which warnings are sent in case of problems. NOTE You can also configure the "From" e-mail address of alert e-mails. This address can be set in `INSTALLROOT/conf/email_config.txt`, in the **common** section, as **from_address**. The symbol "$fqdn" will be replaced by the fully-qualified domain name of the instance host, or an IP address where Universal Licensing is used. The other sections in this file should not normally be modified. For Services Director installs on AWS it is likely that you will need to change this setting to be an address that is resolvable to the instance's public IP.
SMTP Relay Host	The hostname or IP address of the SMTP server used to send warnings.
SMTP Relay Port	The port number of the SMTP server used to send warnings.

NOTE
You must configure the hostname of each Services Director server in your local DNS settings so that all Traffic Manager instances can correctly resolve the addresses of all Services Director servers.

Configuring the MySQL Database for the Services Director

These instructions assume you have already installed the MySQL server. For detailed instructions on installing the MySQL server, see the MySQL documentation for your platform.

You must create an empty MySQL database with an associated user account for use with the Services Director.

The Services Director requires information about the MySQL database during the installation process. Brocade recommends that you note the database name and the username and password before you start the installation process. For detail information about required installation parameters, see Required Configuration Parameters on page 32.

You can install the database on the same host as the Services Director if required.

You must configure the privilege settings for the MySQL user account to allow access to the database from all IP addresses (or at a minimum the IP addresses of the machines on which you install the Services Director).

To Create a MySQL Database with Access to all IP Addresses

1. Log in to the MySQL monitor client program.

2. To create a MySQL database named (for example) **ssc** with a user named (for example) **ssc** , execute the following SQL commands:

    ```
    CREATE DATABASE ssc;
    CREATE USER 'ssc'@'localhost' IDENTIFIED BY '<YOUR PW>';
    GRANT ALL ON ssc.* TO 'ssc'@'%' IDENTIFIED BY '<YOUR PW>' \
    WITH GRANT OPTION;
    FLUSH PRIVILEGES;
    ```

3. Exit the MySQL monitor program.

Configuring the MySQL Database for Remote Availability

To allow two Services Directors to be placed in a cluster of Services Directors, the MySQL database must be set up to allow remote access. An example outlining one approach is below.

To Allow Database Access from a Remote IP Address on Ubuntu

1. Open the **my.cnf** configuration file (for example: `/etc/mysql/my.cnf`) in any text editor.

2. Set the option **bind-address** to the public IP address of the MySQL server

3. Close and save the file.

4. Restart the MySQL daemon, at the system prompt, enter:

    ```
    service mysql restart
    ```

To Allow Database Access from a Remote IP Address on RHEL/CentOS

1. Open the `my.cnf` configuration file (for example: `/etc/mysql/my.cnf`) in any text editor.

2. Set the option **bind-address** to the public IP address of the MySQL server

3. Close and save the file.

4. Restart the MySQL daemon, at the system prompt, enter:

    ```
    service mysqld restart
    ```

Installing and Configuring the Services Director on Ubuntu

For Ubuntu hosts, the Services Director software is provided in the form of a Debian package.

> **NOTE**
> For Ubuntu installations using Linux Containers (LXC), Brocade recommends LXC v0.7.5.

> **NOTE**
> Brocade recommends that the Services Director machines and the instance hosts all use either Ubuntu or RHEL/CentOS.

> **NOTE**
> Brocade recommends that you plan your LXC container deployment strategy before installing the Services Director to ensure adequate system resource availability. For more information about LXC, see the configuration and management instructions supplied with your Linux distribution.

To Install the Services Director on Ubuntu

1. As super user, use the standard **dpkg** command to install the Services Director. At the system prompt, enter:

   ```
   dpkg -i brocade-services-director_17.2_amd64.deb
   ```

   You are prompted for several configuration parameters. For detailed information, see Required Configuration Parameters on page 32.

2. To initialize the software, execute the configuration script. At the system prompt, enter:

   ```
   /opt/riverbed_ssc_17.2/bin/configure_ssc --liveconfigonly
   ```

   You are prompted for the REST API administrative username and password. This login is the administrative user for the Services Director REST API using HTTP authorization. The configuration script validates whether you have configured all the required directories. The script also populates the contents of the Services Director database.

   The initial installation is complete. You can change the configuration parameters after the initial installation by re-running the configuration script.

To Change the Configuration Parameters

1. At the system prompt, enter:

   ```
   configure_ssc
   ```

2. For your changes to take effect, restart the Services Director. At the system prompt, enter:

   ```
   service ssc stop
   service ssc start
   ```

To Re-Enter Configuration Parameters

- At the system prompt, enter:

   ```
   dpkg-reconfigure riverbed-ssc
   ```

Login Settings

You can view and modify login settings for the Services Director in the **/api/tmcm/2.5/settings/security** resource. For details, see Using the REST API with the Services Director on page 123.

You set login settings for the Services Director as follows:

- **max_login_attempts** - The maximum number of failed Services Director login attempts for a user. This has a default of zero, which indicates that there is no maximum.

- **user_lockout_duration_minutes** - A suspension lockout duration (in minutes). If the **max_login_attempts** threshold limit is reached, the suspension duration lockout is applied. This has a default of 1 minute, and a maximum of 1440 minutes (equal to one day).

Installing and Configuring the Services Director on RHEL/CentOS

For an RHEL/CentOS host, the Services Director software is provided as an RPM package.

> **NOTE**
> For RHEL/CentOS installations using Linux Containers (LXC), Brocade recommends LXC v0.7.5.

> **NOTE**
> Brocade recommends that the Services Director machines and the instance hosts all use either Ubuntu or RHEL/CentOS.

> **NOTE**
> Brocade recommends that you plan your LXC container deployment strategy before installing the Services Director to ensure adequate system resource availability. For more information about LXC, see the configuration and management instructions supplied with your Linux distribution.

> **NOTE**
> By default, RHEL/CentOS has more restrictive **iptables** rules than Ubuntu. These **iptables** rules must be amended to enable access to Services Directors and Instance Hosts, either on all or selected ports. For example, to configure an ssh port for an Instance Host.

To Install the Services Director on RHEL/CentOS

1. If it is not already installed, install the GNU C Library. At the system prompt, enter:

   ```
   yum install glibc-devel
   ```

2. At the system prompt, enter:

   ```
   rpm -i brocade-services-director-17.2-0.x86_64.rpm
   ```

3. After installation, configure the Services Director software using the configuration script. At the system prompt, enter:

   ```
   /opt/riverbed-ssc/bin/configure_ssc
   ```

   You are prompted for several configuration parameters. For detailed information, see Required Configuration Parameters on page 32.

To Change the Configuration Parameters

1. To change configuration parameters after the initial installation, at the system prompt, enter:

   ```
   /opt/riverbed-ssc/bin/configure_ssc
   ```

2. For your changes to take effect, restart the Services Director. At the system prompt, enter:

   ```
   stop ssc
   start ssc
   ```

Configuring RHEL/CentOS Instance Hosts

For RHEL/CentOS instance hosts, since the default Python version is 2.6, you must install the Python Argparse module.

To Install the Python Argparse Module

- On a Services Director instance host, enter:

  ```
  yum install python-setuptools
  easy_install argparse
  ```

Login Settings

You can view and modify login settings for the Services Director in the **/api/tmcm/2.5/settings/security** resource. For details, see Using the REST API with the Services Director on page 123.

You set login settings for the Services Director as follows:

- **max_login_attempts** – The maximum number of failed Services Director login attempts for a user. This has a default of zero, which indicates that there is no maximum.
- **user_lockout_duration_minutes** – A suspension lockout duration (in minutes). If the **max_login_attempts** threshold limit is reached, the suspension duration lockout is applied. This has a default of 1 minute, and a maximum of 1440 minutes (equal to one day).

Installing the Services Director Software License

Your license file must be readable by the Services Director Linux user. For detailed information about Services Director licenses, see Managing Services Director Licensing on page 19.

To Install the Services Director License for the First Services Director

1. Place the licenses on an accessible location in your infrastructure. For details about obtaining your license keys, see To Retrieve Brocade vTM and Services Director Product Licenses on page 20.

2. Copy a Services Director license file containing the license key to the `INSTALLROOT/licenses` directory. The Services Director scans the directory for changes at start up and at daily intervals. New licenses are automatically added to the database as they are discovered.

    To confirm that you have correctly installed your license, run the Services Director software manually. Any problems with the license will result in error messages being displayed on the command line. For detailed information about running the Services Director manually, see

    If the Services Director shuts down due to a licensing failure, the start and end dates of any decoded licenses are added as a warning to the event log.

To Install Services Director Licenses Once the First Services Director is Running

•   After you have installed your licenses and started the Services Director for the first time, you can use the REST API to update existing licenses and install new licenses.

```
POST /api/tmcm/2.5/controller_license_key HTTP/1.1 XX1-RSSC123456-3E33-3E3A-5-0123-4567-89AB
```

Creating Licensing Reports

The licensing decode tool (**license_decode**) produces a detailed report for all your license keys, including invalid ones. The license decode tool is part of the Services Director installation package; it is stored under the `INSTALLROOT/bin/` directory.

Invalid license keys might be malformed or might have a dependency on a missing Services Director license key. For license keys listed as invalid, verify that the associated Services Director license key is present.

To create a licensing report, enter the following command at the prompt:

```
license_decode <KEY1> <KEY2> <KEY3>
```

You can decode multiple license keys by specifying each one in a space-separated argument list.

Adding Resources Using the REST API

For a minimal installation of Services Director with one or more externally-deployed Virtual Traffic Manager (vTM) instances, you must add a Feature Pack resource and an Owner resource to the Services Director, and then start adding vTMs.

The section that follows describes a basic installation of Feature Pack, Owner and vTM instance resources, based on the following assumptions:

•   The Services Director is licensed and running.

•   You are using Universal licensing for your vTM instances.

•   You are using externally-deployed vTM instances, each of which has an enabled REST API.

•   You are using standard ports for all operations. That is, 9090 for GUI, 9070 for REST, 161 for SNMP, and so on.

•   You are not using instance-level authentication.

    NOTE
    For a more detailed explanation of vTM deployment, refer to

To Add Resources Using the REST API

NOTE

The examples in this section use CURL statements, but these can be adapted to suit your preferred way of accessing the REST API.

1. Create a Feature Pack using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -u admin:password -
d '{"stm_sku":"STM-300","excluded":""}' https://localhost:8100/api/tmcm/2.5/feature_pack/STM_300_FP
```

In this example, a Feature Pack called "STM_300_FP" is created. This makes use of the "STM-300" SKU, excluding no features. For details of Feature Pack resources, refer to feature_pack Resource on page 144.

When this command completes, a result is displayed. For example:

```
{"info": "", "status": "Active", "stm_sku": "STM-300", "add_on_skus": [], "excluded": ""}
```

NOTE

You may need to create additional Feature Packs for different vTM instances.

2. Create an Owner resource using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -u admin:password -
X POST -d '{"tag":"MegaCorp"}' https://localhost:8100/api/tmcm/2.5/owner/
```

In this example, an Owner called "MegaCorp" is created. For details of Owner resources, refer to owner Resource on page 163.

When this command completes, a result is displayed. For example:

```
{"instances": [], "secret": "tt6YHDgj", "tag": "MegaCorp", "clusters": [], "timezone": "Etc/UTC",
"email_address": "", "owner_id": "Owner-BILV-LIKP-UOL2-DKFV"}
```

NOTE

You may need to create additional Owners for different vTM instances.

3. Register an externally-deployed vTM instance ("vtm-01.cam.demo.com" in this example) with a POST request. This uses the "STM_300_FP" Feature Pack from step 1:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -u admin:password -
X POST -d
'{"status":"Active","stm_feature_pack":"STM_300_FP","owner":"MegaCorp","management_address":"vtm-01.c
am.demo.com","bandwidth":1000,"rest_enabled":true,
"admin_username":"admin","admin_password":"password","tag":"MyRegisteredVTM"}' https://localhost:
8100/api/tmcm/2.5/instance?managed=false
```

In this example, the instance is Active, and has its REST API enabled. For details of instance resources, refer to instance Resource on page 146.

When this command completes, a result is displayed. For example:

```
{"status": "Active", "config_options": "", "managed": false, "cpu_usage": "", "license_name":
"universal_v4", "rest_address": "vtm-01.cam.demo.com:9070", "container_configuration_json": {},
"snmp_address": "vtm-01.cam.demo.com:161", "creation_date": "2016-11-30 15:06:43", "bandwidth":
1000,"tag": "MyRegisteredVTM", "cluster_id": "Cluster-RUZX-APUA-9XUF-9DXH", "container_name": "",
"owner": "Owner-BILV-LIKP-UOL2-DKFV", "service_username": "", "service_password": "",
"config_options_json": {},"admin_username": "admin", "instance_id": "Instance-HBBH-57JR-MHG0-60UL",
"stm_feature_pack": "STM_300_FP","stm_version": "", "host_name": "", "management_address":
"vtm-01.cam.demo.com", "container_configuration": "","rest_enabled": true, "ui_address":
"vtm-01.cam.demo.com:9090", "admin_password": "password"}
```

The **cluster** property is the retrieved cluster ID of the vTM. The **owner** property is set to the ID of the Owner resource rather than its tag.

The instance will have a licence installed and will begin making licensing requests.

4. (Optional) Check the license for the instance at any point with the REST API. You can use either the instance's ID or tag. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -u admin:password
https://localhost:8100/api/tmcm/2.5/instance/Instance-HBBH-57JR-MHG0-60UL
```

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -u admin:password
https://localhost:8100/api/tmcm/2.5/instance/MyRegisteredVTM
```

5. (Optional) Extract the monitoring information for the instance at any point with the REST API. You can use either the instance's ID or tag. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -u admin:password
https://localhost:8100/api/tmcm/2.5/monitoring/instance/Instance-HBBH-57JR-MHG0-60UL
```

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -u admin:password
https://localhost:8100/api/tmcm/2.5/monitoring/instance/MyRegisteredVTM
```

Starting and Stopping the Services Director

You must select the appropriate method of running the Services Director depending on the status of your Services Director installation:

- **Manual** - If you are running the Services Director for the first time or if you are having problems with your Services Director at start up, run the Services Director manually to see additional diagnostic messages on the command line.

- **Upstart** - For an established and normally operating Services Director installation, run the Services Director as an *upstart* service.

To Manually Start the Services Director Software

- At the prompt, enter:

```
run_ssc_server [-v/--version]
```

-v or **--version** - Displays the current version and build number of the Services Director software, but does not start the server.

Manually starting the Services Director provides output and startup messages that you can use to identify and resolve Services Director system configuration issues.

> **NOTE**
> To perform a controlled shutdown of the Services Director from this state, you must press Ctrl+C.

To Start the Services Director as an Upstart Service (Normal Operation)

- At the prompt, enter:

```
start ssc
```

To Stop the Services Director Running as an Upstart Service

- At the prompt, enter:

```
stop ssc
```

Accessing the Operating System Shell

This chapter described how to access the operating system shell so that you can issue commands.

> **NOTE**
> Once you have accessed the operating system shell, the **OS Shell** function is automatically enabled in the Services Director VA GUI. This cannot be disabled once enabled.

1. Connect to the CLI:

```
login as: admin
Brocade Services Director
admin@10.62.167.199's password:
Last login: Tue Sep 11 09:43:12 2016 from 10.62.134.242
amnesia > enable
amnesia # configure terminal
amnesia (config) #
```

2. Start the OS shell:

```
amnesia (config) # _shell
[admin@amnesia ~]#
```

You are now in the operating system shell.

3. To exit the OS shell, type **ctrl + D**.

Upgrading the Services Director on Ubuntu

For Ubuntu hosts, the Services Director software is provided in the form of a Debian package.

You can upgrade the Services Director software using the following procedure.

This procedure causes a loss of service from your Services Director deployment, although you should schedule the upgrade for a time of least interruption to your end users.

> **NOTE**
> Earlier versions of the Services Director used an environment variable called **$SSCHOME** to refer to the software installation directory. This is now deprecated. Before upgrading the Services Director to the latest version, you must make sure that this variable is unset.

> **NOTE**
> To support the encryption of stored passwords for Traffic Manager instances, Services Director encrypts passwords during the upgrade of the Virtual Appliance. A default master password of *master1M@* is used to do this. It is strongly recommended that you update the master password after an upgrade. For a software install, this step is performed automatically after the package upgrade when **configure_ssc --liveconfigonly** is performed. This command prompts you to replace the default master password.

To Upgrade a Single Services Director

1. At the prompt, enter:

   ```
   dpkg -i brocade-services-director_17.2_amd64.deb
   ```

   This stops the Services Director software but does not restart it.

2. Ensure that the `INSTALLROOT/licenses` directory contains the correct license files, including any licenses for Bandwidth Packs.

3. At the prompt, enter:

   ```
   /opt/riverbed_ssc_17.2/bin/configure_ssc --liveconfigonly
   ```

   > **NOTE**
   > This command makes backward-compatible changes to the Services Director database, and prompts you to replace the default master password *master1M@* that was assigned during the upgrade.

4. At the prompt, enter:

   ```
   start ssc
   ```

5. Confirm that the Services Director has not shut down due to licensing issues and that the software has started by issuing a GET request via the Services Director REST API.

Upgrading the Services Director on RHEL/CentOS

For an RHEL/CentOS host, the Services Director software is provided as an RPM package. For RHEL/CentOS installations using Linux Containers (LXC), Brocade recommends LXC v0.7.5.

You can upgrade the Services Director software using the following procedure.

This procedure causes a loss of service from your Services Director deployment, although you should schedule the upgrade for a time of least interruption to your end users.

> **NOTE**
> Earlier versions of the Services Director used an environment variable called **$SSCHOME** to refer to the software installation directory. This is now deprecated. Before upgrading the Services Director to the latest version, you must make sure that this variable is unset.

> **NOTE**
> To support the encryption of stored passwords for Traffic Manager instances, Services Director encrypts passwords during the upgrade of the Virtual Appliance. A default master password of *master1M@* is used to do this. It is strongly recommended that you update the master password after an upgrade. For a software install, this step is performed automatically after the package upgrade when **configure_ssc** is performed. This command prompts you to replace the default master password.

To Upgrade a Single Services Director

1. At the prompt, enter:

   ```
   rpm -U brocade-services-director-17.2-0.x86_64.rpm
   ```

   This stops the Services Director software but does not restart it.

2. Make sure the `INSTALLROOT/licenses` directory contains the correct license files, including any licenses for Bandwidth Packs.

3. At the prompt, enter:

   ```
   /opt/riverbed-ssc/bin/configure_ssc
   ```

   > **NOTE**
   > This command makes backward-compatible changes to the Services Director database, and prompts you to replace the default master password (*master1M@*) that was assigned during the upgrade.

4. At the prompt, enter:

   ```
   start ssc
   ```

5. To confirm that the Services Director has not shut down due to licensing issues and that the software has started by issuing a GET request via the Services Director REST API.

Upgrading Clustered Services Directors

Part of the upgrade process for a Services Director is to apply changes to the inventory database schema, in order to support additional features in newer versions. Since clustered Services Directors share the same inventory database, upgrades of these clustered Services Directors must be carefully sequenced to ensure that:

- inventory records are not locked during the upgrade of the database schema, as this will cause the upgrade to fail.
- once the database schema has been upgraded, changes made to the inventory by remaining pre-upgrade Services Directors are avoided.

To support these goals, it is recommended that the following sequence is used:

1. Before starting the upgrade, select one Services Director in your Services Director cluster as the upgrade master. This Services Director will be upgraded first and will be responsible for applying changes to the database schema.

2. Stop the other Services Director in the Services Director cluster.

3. Upgrade the upgrade master as instructed in To Upgrade a Single Services Director on page 42, and restart this Services Director once the upgrade is complete.

> **NOTE**
> Deactivation of the upgrade master Services Director may cause instances to briefly enter the licensing grace period, but they will return to normal operation once the upgrade master is restarted.

4. Upgrade the second Services Director in the cluster as instructed in To Upgrade a Single Services Director on page 42, and restart this Services Director once the upgrade is complete.

5. Repeat step 4 until all clustered Services Directors are upgraded.

Downgrading the Services Director on Ubuntu

For Ubuntu hosts, the Services Director software is provided in the form of a Debian package.

To revert to the previous software version in a cluster of Services Directors, follow this procedure on all Services Directors in turn.

To Revert to the Previous Software Version

1. At the prompt, enter:

```
stop ssc
```

2. At the prompt, enter:

```
dpkg --force-downgrade -i riverbed_ssc_1.3_amd64.deb
```

--force-downgrade - Downgrades to the desired software version. In this example, version 1.3. This includes a database schema downgrade.

3. Make sure that the INSTALLROOT/licenses directory in your downgraded software installation contain the required license keys compatible with the downgraded version of the Services Director.

4. At the prompt, enter:

```
sudo INSTALLROOT/bin/configure_ssc --liveconfigonly
```

5. At the prompt, enter:

```
start ssc
```

Downgrading the Services Director on RHEL/CentOS

For an RHEL/CentOS host, the Services Director software is provided as an RPM package. For RHEL/CentOS installations using Linux Containers (LXC), Brocade recommends LXC v0.7.5.

To revert to the previous software version in a cluster of Services Directors, follow this procedure on all Services Directors in turn.

To Revert to the Previous Software Version

1. At the prompt, enter:

    ```
    stop ssc
    ```

2. If you intend to revert to version 2.4 or earlier, enter the following command:

    > **NOTE**
    > Do not use this command if you intend to revert to version 2.5 or later.

    ```
    /opt/riverbed-ssc/bin/configure_ssc --liveconfigonly --to_version=2.1
    ```

    --to_version=2.1 - Specifies the downgrade version. In this example, version 2.1. This includes a database schema downgrade.

    You will be asked if you want to add another REST API user. You can add a user, but this is not needed for the downgrade process.

3. For all versions, enter:

    ```
    rpm -U --oldpackage riverbed-ssc-2.1-0.x86_64.rpm
    ```

    --oldpackage - Downgrades to the desired software version. In this example, 2.1.

4. Make sure that the `INSTALLROOT/licenses` directory in your downgraded software installation contain the required license keys compatible with the downgraded version of the Services Director.

5. At the prompt, enter:

    ```
    sudo INSTALLROOT/bin/configure_ssc --liveconfigonly
    ```

6. At the prompt, enter:

    ```
    start ssc
    ```

Using an Instance Host Machine with a Software Services Director

This section describes how to deploy Traffic Managers to an instance host machine from a software form-factor Services Director.

> **NOTE**
> Other form-factor configurations are supported:
>
> * To deploy Traffic Managers to an instance host VA from a Services Director VA, see Using an Instance Host VA with a Services Director VA on page 67.
>
> * To deploy Traffic Managers to an instance host machine from a Services Director VA, see Using an Instance Host Machine with a Services Director VA on page 97
>
> * To register externally-deployed Traffic Managers, see Registering Externally-Deployed Traffic Managers on page 117.

See also Services Director Form-Factors on page 14 and Instance Host Form-Factors on page 15.

Overview: Using an Instance Host Machine with a Software Services Director

This section describes how to deploy Traffic Manager instances from a system that combines a software form-factor Services Director with an *instance host* machine.

> **NOTE**
> This section assumes that you have installed, licensed and configured your software form-factor Services Director.

An instance host machine is an Ubuntu 14.04 machine (virtual or physical) that is configured specifically for the purpose of allowing Services Director to deploy instances to it.

> **NOTE**
> It is also possible to use a RedHat/CentOS machine in the same way.

> **NOTE**
> If you want an instance host machine that will support interaction with a Services Director VA, see Using an Instance Host Machine with a Services Director VA on page 97.

To create an instance host machine, the required steps are:

1. Create an Ubuntu 14.04 machine and its network interfaces, see To Create the Instance Host on page 50.

NOTE

Ensure that an SSH Server is present on the Ubuntu 14.04 machine.

2. On the Ubuntu 14.04 machine, set up the required directory structures on the Ubuntu 14.04 machine, see To Prepare the Directory Structure on page 51.

3. Set up local networking (interfaces/NICs and bridges) on the Ubuntu 14.04 machine, see To Configure the Local Network on page 51. Exact details will depend on your networking requirements (see example below).

4. On the Ubuntu 14.04 machine, set up an LXC configuration file for each required LXC container, see To Create an LXC Container Configuration File on page 52.

5. Establish SSH communications between the software form-factor Services Director and the Ubuntu 14.04 machine, see To Enable Passwordless SSH Communication on page 54.

6. Transfer any required Virtual Traffic Manager (vTM) images onto the Services Director, see To Prepare Traffic Manager Images on page 54.

7. Create any required resources (versions, feature packs and legacy FLA if required) on the Services Director VA, see Creating Required Resources on page 54.

8. Register the instance host on the Services Director, see Adding an Instance Host Machine to the Software Services Director on page 57.

For example:

FIGURE 3 Administration Tasks to Prepare the Instance Host
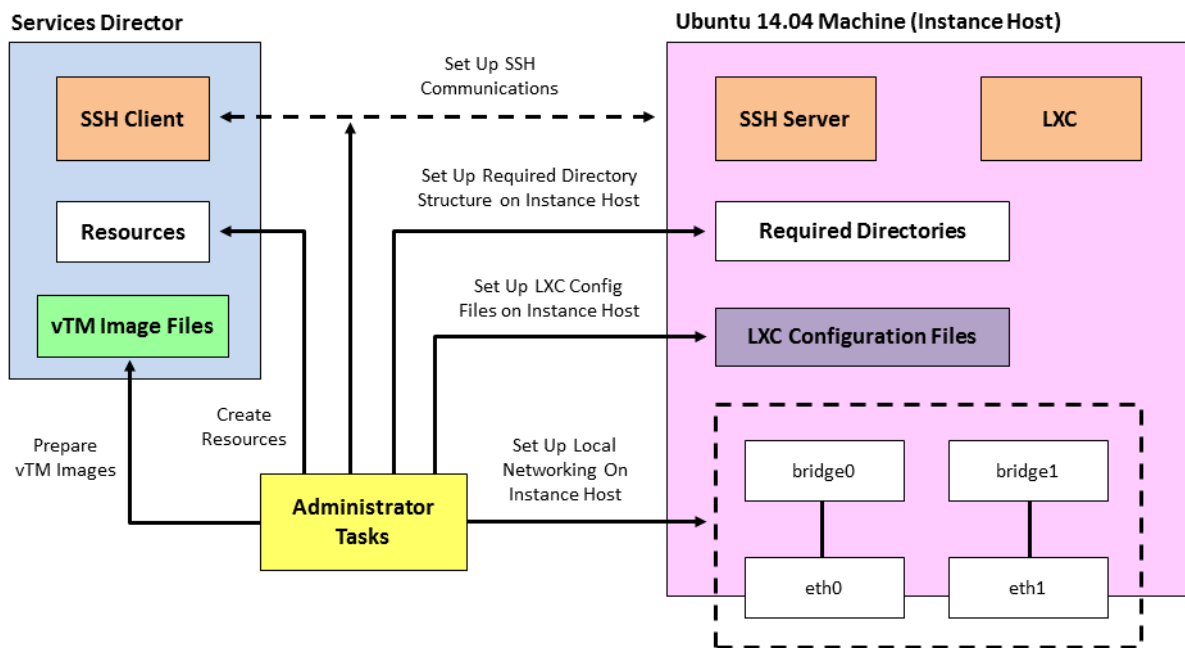
Once these steps are complete the instance host is ready.

To create a vTM on the instance host, request the deployment from the Services Director, see Deploying a Traffic Manager to an Instance Host Machine on page 57. The following events then occur automatically:

- The Services Director contacts the instance host using SSH to request that LXC deploys a vTM on the instance host.

- Services Director transfers the required vTM image file to the instance host.
- LXC retrieves the required LXC configuration file based on the information passed from the Services Director.
- LXC uses the LXC configuration file to create an LXC container and configure its interfaces.
- The instance host uses the required vTM image to deploy a vTM to the LXC container.
- The new vTM is added to the list of vTMs that are in the estate of the Services Director.

Once these events are complete, the first vTM is ready for use.

FIGURE 4 Creating a vTM on an Instance Host
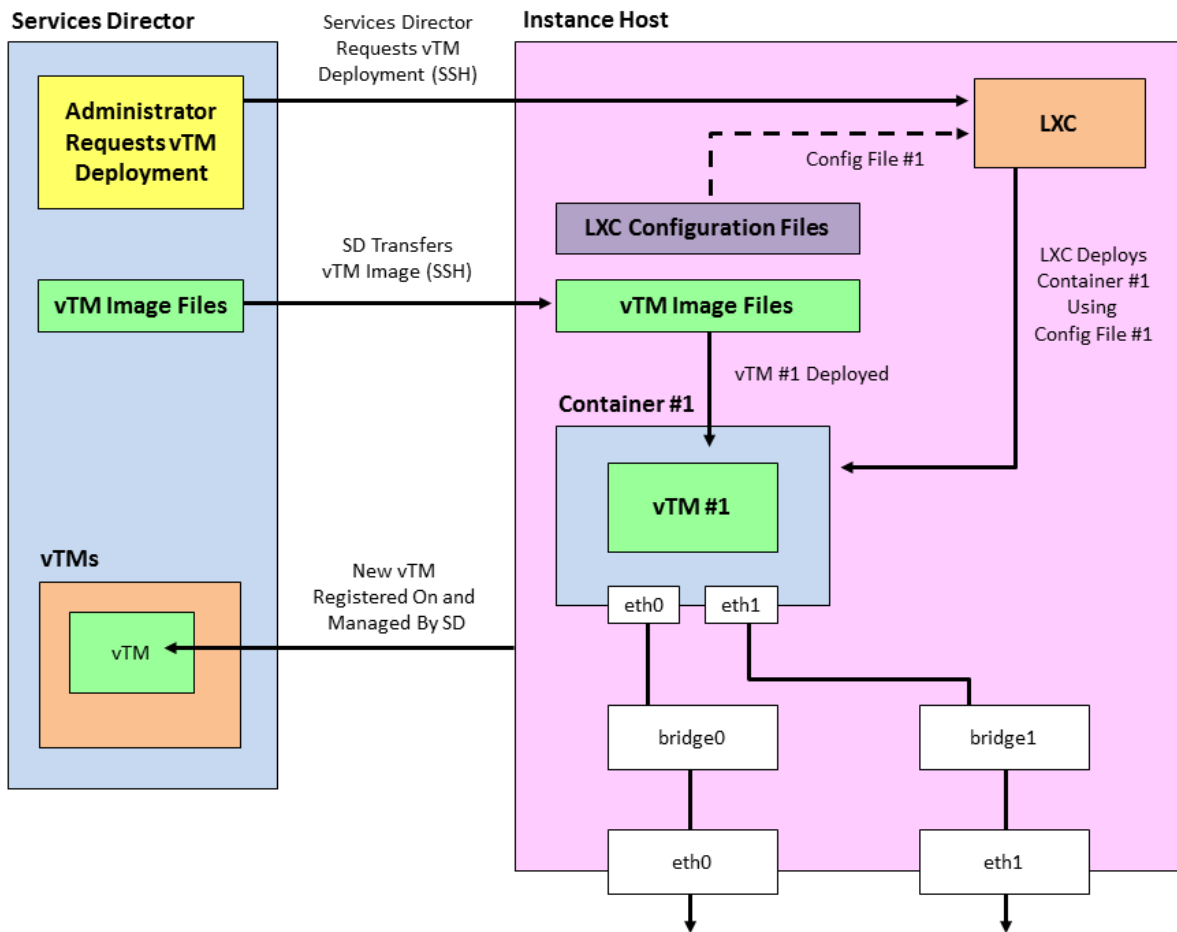
You can repeat this process to deploy a second (and subsequent) vTMs onto the same instance host. This is broadly similar to the first vTM. Please note that:

- This requires a different LXC configuration file.
- A second container configuration file must already exist or be created. You cannot deploy a second vTM to the same container as the first vTM.
- A vTM image file is only transferred to the instance host if the vTM image file is not already cached there.
- All other events are the same as for the first vTM.

FIGURE 5 Creating another vTM on an Instance Host

Creating and Configuring an Instance Host Machine

You can use an instance host machine (either virtual or physical) with both the software form-factor Services Director and the Services Director VA.

To Create the Instance Host

To create an instance host, you must first create an Ubuntu 14.04 machine. During this process, you will need:

- The IP address and hostname that is required for the instance host.
- The IP address for the management interface (eth0).
- Optionally, an IP address for each of the required network interfaces (eth1, eth2, and so on).
- Optionally, the IP address(es) of the DNS name server(s).

Perform the following steps:

1. Create a machine to contain the Ubuntu 14.04 operating system. This machine can be either virtual or physical.

   The machine's management network interface must be accessible from the Services Director

2. Install the Ubuntu 14.04 operating system using the standard set-up software.

3. Ensure that the Ubuntu 14.04 machine has SSH server installed and running.

4. Install LXC and bridge utilities on the Ubuntu 14.04 machine to enable container deployment:

   ```
   apt-get install lxc bridge-utils
   ```

To Prepare the Directory Structure

You must prepare the Ubuntu 14.04 instance host directory structure to receive the required Traffic Manager files. These are required when you register the instance host on the Services Director.

1. On the instance host, create a directory to place files for use. For example, for Traffic Manager image files and license files. For example:

   ```
   /var/cache/ssc
   ```

2. Then, create a parent directory for all deployed Traffic Managers. For example:

   ```
   /root/install
   ```

3. Check that suitable permissions are set for both of the new directories (from steps 1 and 2).

   Services Director assumes that all required directories on the instance host have read/write/execute permissions for the root user / group. Check this on the instance host:

   ```
   stat --format=%a <directory>
   ```

   And set if required by using:

   ```
   chmod 770 <directory>
   ```

To Configure the Local Network

Configure a virtual network on the instance host for the Traffic Manager instance. Typically, this configuration requires setting up a virtual bridge (using, for example, Linux bridge or Open vSwitch), then configuring the Linux container to attach to the bridge when running Traffic Manager.

NOTE
You must disable the default bridge configured by LXC. This is unsuitable for Services Director operations.

1. On the instance host, edit the `/etc/network/interfaces` file and configure the network interfaces and bridges as required. For example:

```
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet static
    address XX.XX.XX.XX
    netmask YY.YY.YY.YY
    network ZZ.ZZ.ZZ.ZZ
    broadcast AA.AA.AA.AA
    gateway BB.BB.BB.BB
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers DD.DD.DD.DD
    dns-search cam.demo.com
# The primary bridge
auto br0
iface br0 inet static
    bridge_ports eth1
    bridge_fd 0
    address EE.EE.EE.EE
    netmask FF.FF.FF.FF
    network GG.GG.GG.GG
# The secondary network interface
auto eth1
iface eth1 inet static
    address 0.0.0.0
```

In this example, there is a Linux bridge setup to which LXC containers can attach virtual network interfaces.

2. You must now disable the LXC default bridge. Edit the `/etc/default/lxc-net` file and set the following:

```
USE_LXC_BRIDGE="false"
```

Also comment out any other lines that contain **'LXC_*='**.

3. You must now turn off strict return path filtering. Edit the `/etc/default/sysctl.conf` file:

```
net.ipv4.conf.default.rp_filter=2
net.ipv4.conf.all.rp_filter=2
```

4. Either reboot, or restart the networking services. This ensures that the networking configuration is correct and the default LXC bridge is removed.

To Create an LXC Container Configuration File

Traffic Manager instances can cohabit on an instance host machine using LXC containers.

LXC containers support network isolation and a degree of resource isolation. The software form-factor Services Director expects prepared container configuration files in advance of instance deployment. You must put these files in the installation root directory of the instance host.

Before you create an LXC container for a Traffic Manager, you must create an LXC configuration file. This file describes the properties for the container.

1. On the instance host, create an LXC configuration text file.

    This must be located in the parent directory for installations (see To Prepare the Directory Structure on page 51). For example, in the `/root/install` directory.

    The container configuration file must include networking entries for your Traffic Manager instance IP address. You can use either Linux virtual networking or Open vSwitch to set up network isolation for your Traffic Manager instance.

    The filename of the container configuration file must match the FQDN or IP address of the required container. That is: `<container-fqdn-or-ip>.conf`.

    > **NOTE**
    > This filename must be specified as the **container_name** property when you deploy the instance.

    > **NOTE**
    > For Instance resources that are not able to use the Universal FLA License, `<container-fqdn-or-ip>.conf` must match the value of the **lxc.utsname** used in the file, and the **management_address** specified in the REST request.

2. Create contents for the LXC container configuration file. For example:

    ```
    lxc.network.type = veth
    lxc.network.flags = up
    lxc.network.link = br0
    lxc.network.ipv4 = XX.XX.XX.XX/SS
    lxc.network.ipv4.gateway = YY.YY.YY.YY
    lxc.utsname = vtm-mgmt.example.com
    ```

    In this example:

    *   XX.XX.XX.XX/SS is the IP address and subnet mask of the LXC container network.

    *   YY.YY.YY.YY is the IP address of the subnetwork gateway inside the LXC container.

    *   If the Legacy FLA License is in use for this instance, the filename of the container configuration file must be `vtm-mgmt.example.com.conf`.

Configuring the Services Director Software

Before you register the instance host machine and deploy Traffic Managers to it, you must perform the following tasks:

*   Enable passwordless SSH communication between the Services Director and the instance host machine, see To Enable Passwordless SSH Communication on page 54.

*   Transfer any required Traffic Manager images to the Services Director, see To Prepare Traffic Manager Images on page 54.

*   Create any required version resources on the Services Director, see To Create a Version Resource on page 54.

*   If you cannot use Universal FLA Licensing, create any required Legacy FLA license resources on the Services Director, see To Create a Legacy FLA License Resource on page 55.

*   Create any required feature pack resources on the Services Director, see To Create a Feature Pack Resource on page 55.

You can also verify that these have been created, see To Verify a Resource on page 56.

To Enable Passwordless SSH Communication

The Services Director uses passwordless SSH to communicate with your instance hosts. Passwordless SSH enables the Services Director to copy files and remotely run commands to deploy, start, stop, upgrade, and delete Traffic Manager instances.

Each Services Director in a cluster of Services Directors has its own unique SSH authentication key pair.

When you add a new instance host to the Services Director, you must set up passwordless SSH between each Services Director and the new instance host.

1. If do not yet have an SSH authentication key pair for your Services Director, run the following command on the Services Director.

    > NOTE
    > You must be logged in as the user used to run the Services Director software.

    ```
    ssh-keygen -t rsa
    ```

    When prompted, enter a blank pass-phrase, and accept the default key location. This operation results in the generation of these key files:

    *   `~/.ssh/id_rsa` (certificate)
    *   `~/.ssh/id_rsa.pub` (public key)

    This SSH authentication key pair will be used for SSH passwordless communication between this Services Director and all of its instance hosts.

2. Use the **ssh-copy-id** command (included in the **openssh-client** package) to install your public key to the instance host **authorized_keys** file. For example:

    ```
    ssh-copy-id root@<instancehost>
    ```

3. Return to the command line of the Services Director and connect to the new instance host by SSH as the root user. If you are successful, the credentials for passwordless SSH are correctly set up.

    After these credentials have been set up, you can add the host to the Services Director system by means of a REST request. See Adding an Instance Host Machine to the Software Services Director on page 57.

To Prepare Traffic Manager Images

1. Obtain your Virtual Traffic Manager (vTM) installation image (that is, a .TGZ file) from Brocade Support and store them locally.
2. Log into the Services Director and transfer the local .TGZ image files into the directory you created to place files for use, see To Prepare the Directory Structure on page 51. For example: `/var/cache/ssc`.

Creating Required Resources

You must create any required version, and feature pack resources.

Also, if you are unable to use Universal FLA Licensing, you must create a Legacy FLA License resource.

To Create a Version Resource

After you have added the required Virtual Traffic Manager (vTM) installation images to the Services Director, you can create all required **version** resources.

Each **version** resource represents a software version number for a vTM release. This is associated with the image file that is used to deploy a Traffic Manager that has this version.

Perform the following procedure to create a REST API **version** resource. For detailed information about the properties for this resource, see version Resource on page 181.

> To create a **version** resource, perform a PUT request:
>
> ```
> https://<sdhost>:<port>/api/tmcm/2.5/version/11.0
> ```
>
> Use the following JSON structure as the body of your request:
>
> ```
> {
>     "version_filename" : "ZeusTM_110_Linux-x86_64.tgz",
>     "info" : "Version 11.0"
> }
> ```
>
> In this example, the 11.0 **version** resource is associated with the `ZeusTM_110_Linux-x86_64.tgz` image file.

To Create a Legacy FLA License Resource

If you unable to use Universal FLA Licensing, you must create a Legacy FLA **license** resource using the REST API. For detailed information about the properties for this resource, see license Resource on page 157.

1. Obtain all required FLA-style license keys. For details about retrieving your licenses, see To Retrieve Brocade vTM and Services Director Product Licenses on page 20.

2. Put the retrieved license keys into the directory you created to place files for use, see To Prepare the Directory Structure on page 51. For example: `/var/cache/ssc`.

3. To create a **license** resource for an FLA-style license key file (for example **flexlic1**), perform a PUT request:

   ```
   https://<sdhost>:<port>/api/tmcm/2.5/license/flexlic1
   ```

   Include the following JSON structure as the body of your request:

   ```
   {
       "info" : "This is the resource for the flexlic1 license"
   }
   ```

   > **NOTE**
   > You can pass an empty JSON structure (**{}**) to the license resource because there are no mandatory properties for this resource type.

To Create a Feature Pack Resource

To complete the set of supporting resources required by a Traffic Manager instance, you must specify one or more feature packs with the required feature sets. The value of the property of the feature pack resource must be the name of one of the available **sku** resources created when the Services Director was installed. The choice of SKU depends on a combination of:

- the features you want to enable on the instances that will use the feature pack.
- the available licensed bandwidth for the SKU (for Enterprise Licensed customers).

- the metered cost of the SKU (for Cloud Service Provider licensed customers).

1. To create a **feature_pack** resource (for example, **STM-U-CSP-200-FP**), perform a PUT request:

```
https://<sdhost>:<port>/api/tmcm/2.5/feature_pack/STM-U-CSP-200-FP
```

Use the following JSON structure as the body of your request to define the SKU:

```
{
    "stm_sku" : "STM-U-CSP-200",
    "excluded" : ""
}
```

The **excluded** property is a space-separated list of features that are specifically excluded from the default list of features specific to each SKU, see the

In this example, **STM-U-CSP-200** represents an available **sku** resource in the Services Director that is compatible with your Services Director base license.

2. When complete, the following response is displayed:

```
{
    "info": "",
    "status": "Active",
    "stm_sku": "STM-400",
    "add_on_skus": [],
    "excluded": ""
}
```

To Verify a Resource

You can use the Services Director REST API to verify that all the required files are present and accessible. Perform a file status check GET request to verify that the Services Director can find and access the files that are referred to in any of the defined resources.

To verify that the Services Director can find and access the Traffic Manager package and license key files, perform a file status check GET request:

```
https://<sdhost>:<port>/api/tmcm/2.5/status/files
```

The following output is displayed:

```
{
    "licenses": [{
        "href": "/api/tmcm/2.5/license/flexlic1",
        "name": "flexlic1",
        "present": true,
        "filename": "/var/lib/ssc/flexlic1"
    }],
    "versions": [{
        "href": "/api/tmcm/2.5/version/11.0",
        "name": "11.0",
        "present": true,
        "filename": "/var/lib/ssc/ZeusTM_110_Linux-x86_64.tgz"
    }]
}
```

In the above example, the **present** property showing as true indicates that the file is in place. If it is false, the file must be placed in the location indicated.

Adding an Instance Host Machine to the Software Services Director

To complete the introduction of an instance host machine, you must register it on your software form-factor Services Director.

To Add an Instance Host Machine to the Software Services Director

Once all preparations are complete, you can register the instance host machine on the software form-factor Services Director.

1. To create the instance host, perform a PUT request:

    ```
    https://mysscontroller.mydomain.com:8000/api/tmcm/2.5/host/ myhost-01.mydomain.com
    ```

    Include the following JSON structure as the body of your request:

    ```
    {
      "cpu_cores":"0-3",
      "work_location":"/space/workspace",
      "install_root":"/space/install",
      "username":"root",
      "retained_info_dir":"/space/retain",
    }
    ```

2. Test the registered host by making a REST request to get the host resource, with the addition of the **?status_check=true** query parameter. For example, make the request to:

    ```
    https://mysscontroller.mydomain.com:8100/api/tmcm/2.5/host/myhost-01.mydomain.com?status_check=true
    ```

    The response will include a **status_check** property that will be an empty object if the host is correctly configured for deployments.

    If not, **status_check** will have a further **setup_errors** property describing the problems found with the host.

Deploying a Traffic Manager to an Instance Host Machine

After you have completed preparations to deploy a Traffic Manager instance, you can deploy it to the instance host.

You will need to be aware of:

*   The properties required when deploying the instance, see Properties for a Deployed Instance on page 58.
*   The configuration and container options you want to specify when deploying the instance, see Specifying Configuration Options and Container Options on page 60.

There are two common deployment scenarios:

*   Deploying a Traffic Manager instance with a fixed unique identifier and a changeable user-facing name. This enables Traffic Manager names to be changed and for names of deleted Traffic Manager instances to be re-used under most circumstances. See To Deploy a Traffic Manager Instance with a UUID and a Tag on page 61.

    > NOTE
    > Brocade recommends this approach wherever possible.

- Deploying a Traffic Manager instance with a permanent chosen name. See To Deploy a Traffic Manager Instance with a Chosen Name on page 63.

NOTE
As a part of the deployment process for all deployed instances, a check of the FLA license is performed automatically, although this feature may optionally be disabled. See Checking the Health of an FLA License Automatically on page 27.

NOTE
Instances can be also be registered using the procedures described in the *Brocade Services Director Getting Started Guide*.

Properties for a Deployed Instance

The table below describes the REST API properties for a deployed instance in an LXC container.

When you create the instance, make sure you assign the same name to both the instance and the container, because the instance uses the container name as its hostname.

For details of all **instance** properties, see instance Resource on page 146.

Property	Description
host_name	The name of the Traffic Manager instance host on which you deploy the instance. This name must match the name of the instance host that was created. This property is a fully-qualified domain name, or may optionally be an IP address where Universal Licensing is used. You must create a host entry before you create an instance.
container_name	The name of the LXC container for the Traffic Manager instance. This name must match the name of the required container configuration (**.conf**) file in the installation directory. Do not include the **.conf** file extension. You must create an appropriate container configuration file of the form **<containername>.conf** in the `install_root` directory of the container host. For example, a **container_name** of stm1.example.com requires a container configuration file named `stm1.example.com.conf`. The container configuration file must set **lxc.utsname** to the container name for the licensing server to operate correctly.
container_configuration	A space-separated string used to set up the default network gateway inside the LXC container. See Specifying Container Options on page 61. Use this format: `"{\"gateway\":\"<ip_address>\"}"` This is the IP address raised on the bridge interface to which this container is connected. NOTE Brocade recommends the use of the **container_configuration_json** property instead of this property. NOTE When you specify this property, do not specify the **container_configuration_json** property.
container_configuration_json	A JSON data structure that is equivalent to the **container_configuration** property, but which avoids the need to perform JSON escaping.

Property	Description
	NOTE Brocade recommends the use of this property rather than **container_configuration**. See Specifying Container Options on page 61. **NOTE** When you specify this property, do not specify the **container_configuration** property.
owner	Specify who owns the instance.
stm_version	The name of the Traffic Manager**version** resource for the instance.
stm_feature_pack	The name of the *feature_pack* resource associated with the Traffic Manager instance. This feature pack represents the set of features that are available for the instance.
license_name	The name of the FLA **license** resource you want to use for this instance. When you modify this property, the Services Director updates the license on the Traffic Manager instance.
rest_enabled	Should set to true. This is required to enable monitoring and to allow licenses to be pushed to instances, and for the use of Universal FLA licensing.
config_options	A space-separated string used to define configuration options. See Specifying Configuration Options on page 61. If you specify the **cluster_id** property, then you must also set the **config_options** property to include **admin_ui=yes** and **start_flipper=yes**. **NOTE** The **config_options** property is visible in the graphical user interface of the Services Director VA. It is listed for an expanded instance on the **vTM Instances** page as **Extra Options**. Refer to the *Brocade Services Director Getting Started Guide*. **NOTE** Brocade recommends the use of the **config_options_json** property instead of this property. **NOTE** Whenever the **config_options** property is set, all currently modified options must be specified again in the REST call. Any options that are not specified will lose their current value and be reset to their default value. **NOTE** Any change to the **config_options** settings on a deployed instance will cause a restart of the instance. Externally-deployed instances are not affected. **NOTE** When you specify this property, do not specify the **config_options_json** property.
config_options_json	A JSON data structure that is equivalent to the **config_options** property. See Specifying Configuration Options on page 61. **NOTE** Brocade recommends the use of this property rather than **config_options** property.

Property	Description
	NOTE When you specify this property, do not specify the **config_options** property.
bandwidth	The maximum allowed bandwidth for the Traffic Manager instance (in Mbps).
tag	A text property which provides an alternative way of referring to an instance. Unlike the unique ID for an instance, the tag value can be changed or reused, subject to some restrictions. See Understanding the Tag Property on page 125.
cpu_usage	A string that describes which CPUs are used for this Traffic Manager instance. The CPU affinity is defined using the **lxc.cgroups.cpuset** setting in the container configuration file, in which case **cpu_usage** is set to an empty string. **NOTE** Any change to the **cpu_usage** settings of a deployed instance will cause a restart of the instance. Externally-deployed instances are not affected.
cluster_id	Optionally, the name of a **cluster** resource to which the instance belongs. If you specify an entry for this property, it must refer to a **cluster** resource. **NOTE** This must be a *User-Created* cluster, and not a *Discovered* cluster. The **cluster_id** property cannot be changed after you create an instance. Instances must be added to a cluster when you create them. (This requirement also applies to the first instance in a cluster.) If you specify the **cluster_id property** , then you must also set the **config_options** property to include **admin_ui=yes** and **start_flipper=yes**.
management_address	When using Legacy licensing, the **management_address** property is typically the same as the **lxc.utsname** and the container configuration filename (minus the **.conf** file extension). This property is a fully-qualified domain name, or an IP address where Universal Licensing is used. For example, if the container configuration filename is: `stm1.example.com.conf` and the **lxc.utsname** is defined as follows: `lxc.utsname = stm1.example.com` then the **management_address** is defined as follows: `management_address = stm1.example.com`

Specifying Configuration Options and Container Options

When deploying and configuring either deployed and externally-deployed instances, you can specify two sets of options:

- Configuration options, see Specifying Configuration Options on page 61.
- Container options, see Specifying Container Options on page 61.

These option sets can be specified as either a space-separated string, or as a JSON data structure.

Specifying Container Options

When you specify container options, you can use either the **container_configuration** or **container_configuration_json** properties.

For example, for the **container_configuration** property, you can specify:

```
"container_configuration": "{\"gateway\":\"10.62.128.1\"}"
```

Alternatively, you can specify this as a JSON data structure in the **container_configuration_json** property:

```
"container_configuration_json": {
      "gateway": "10.62.128.1"
  }
```

> **NOTE**
> Brocade recommends the use of the **container_configuration_json**property rather than **container_configuration** property.

Once one of these properties is set, the result of any GET will show the **container_configuration** and **container_configuration_json** properties set to equivalent values. See "instance Resource".

Specifying Configuration Options

When you specify container options, you can use either the **config_options** or **config_options_json** properties.

For example, for the **config_options** property, you can specify:

```
"config_options": "admin_ui=yes start_flipper=no java!enabled=yes
        webcache!size=5mb flipper!monitor_interval=1000 snmp!community=notpublic "
```

Alternatively, you can specify this as a JSON data structure in the **config_options_json** property:

```
"config_options_json": {
      "admin_ui": true,
      "start_flipper": false,
      "java": {
          "enabled": true
      },
      "webcache": {
          "size": "5mb"
      },
      "flipper": {
          "monitor_interval": 1000
      },
      "snmp": {
          "community": "notpublic"
      }
  }
```

> **NOTE**
> Brocade recommends the use of the **config_options_json**property rather than **config_options** property.

Once one of these properties is set, the result of any GET will show the **config_options** and **config_options_json** properties set to equivalent values. See "instance Resource".

To Deploy a Traffic Manager Instance with a UUID and a Tag

Perform this procedure to create a Traffic Manager instance whose name can be changed, and may be reused for another instance after deletion.

NOTE
Brocade recommends the use of tags when creating an instance.

The resource name for the instance is an automatically-generated unique identifier. This cannot be changed after it is assigned, and is unique among all Traffic Manager instances (included deleted instances).

However, the Traffic Manager instance also has a "user-facing" name stored as its **tag** property. This can be changed as required, but it must always be unique among non-deleted Traffic Manager instances, see Understanding the Tag Property on page 125. The tag must be a valid alphanumeric name suitable for use in a URI. For example, **stm1**.

1. Choose a Traffic Manager instance host to which you want to deploy the instance. This is the fully-qualified domain name for your LXC container, or may optionally be the IP address of your container if Universal FLA Licensing is used.

2. To create the **instance** resource with a unique identifier, perform a POST request. For example, the URI for a POST request to create an instance with an automatically-generated unique identifier is:

```
https://<sdhost>:<port>/api/tmcm/2.5/instance
```

This request supports the following URL parameters:

- **?managed=true** to indicate that this is a deployed instance. However, this is the default setting, and can be omitted for deployed instances.

- **?override_fla_check=true** to prevent an automatic FLA license check before deployment. The default setting is false. That is, an automatic FLA license check is performed. You can also disable the FLA license check at the Services Director level. See Disabling the FLA Health Checker on page 27.

Use the following example JSON structure as the body of your request:

```
{
    "owner" : "ACME Corp",
    "stm_version" : "11.0",
    "host_name": "1h1.mydomain.com",
    "container_name": "1c1.mydomain.com",
    "container_configuration": "{\"gateway\": \"XX.XX.XX.XX\" }",
    "config_options": "admin_ui=yes",
    "cpu_usage": "0",
    "stm_feature_pack": "STM-U-CSP-400-FP",
    "bandwidth": 100,
    "tag": "stm1",
    "rest_enabled": true,
    "license_name": "flexlic1",
    "management_address": "1m1.mydomain.com"
}
```

In this example:

- XX.XX.XX.XX is the IP address of your gateway.

- When the **container_name** and **container_configuration** properties are used, the **container_name_json** and **container_configuration_json** properties are not required. See Specifying Configuration Options and Container Options on page 60.

- **1c1.mydomain.com** is the LXC container. This property is a fully-qualified domain name, or an IP address where Universal Licensing is used.

- **1m1.mydomain.com** is the management host. This is the same as 1c1.mydomain.com.

- The **tag** property is set to **stm1**.

The REST API response indicates that the instance is scheduled to deploy.

3. A check of the FLA license is performed automatically, unless prevented in step 2 by the **?override_fla_check=true** URL parameter.

   - If the license check fails, this procedure ends. The instance has a status of Failed to Deploy, and a deployment action shows as Blocked.

   - If the license check succeeds, this procedure continues.

   See

4. To poll the Services Director until the instance is successfully deployed, perform a GET request for the URI of the instance that you created, using its tag to identify it. For example:

   ```
   https://<sdhost>:<port>/api/tmcm/2.5/instance/stm1
   ```

   The response to this request contains a JSON structure representing the **instance** resource. This contains additional properties, one of which gives the status of the instance.

   > **NOTE**
   > If the deployment fails, you will first need to identify and resolve the the issue identified in the **block_reason** property of the associated deployment action. Then, you must remove the installation directory for the instance. You are then able to re-attempt the deployment by setting the Blocked deployment action to a Waiting state.

To Deploy a Traffic Manager Instance with a Chosen Name

Perform this procedure to create a Traffic Manager instance when you have chosen a permanent unique identifier (name) for instance. Your chosen name must use a valid alphanumeric name suitable for use as a directory name and as part of a URI. For example, **stm1**.

> **NOTE**
> Brocade recommends the use of tags when creating an instance, see

1. Choose a Traffic Manager instance host to which you want to deploy the instance. This is the fully-qualified domain name for your LXC container, or may optionally be the IP address of your container if Universal FLA Licensing is used.

2. To create the named **instance** resource, perform a PUT request. For example, the URI for a PUT request to create an instance called **stm1** is:

```
https://<sdhost>:<port>/api/tmcm/2.5/instance/stm1
```

This request supports the following URL parameters:

- **?managed=true** to indicate that this is a deployed instance. However, this is the default setting, and can be omitted.

- **?override_fla_check=true** to prevent an automatic FLA license check before deployment. The default setting is false. That is, an automatic FLA license check is performed. You can also disable the FLA license check at the Services Director level. See Disabling the FLA Health Checker on page 27.

Use the following example JSON structure as the body of your request:

```
{
    "owner" : "ACME Corp",
    "stm_version" : "11.0",
    "host_name": "1h1.mydomain.com",
    "container_name": "1c1.mydomain.com",
    "container_configuration": "{\"gateway\": \"XX.XX.XX.XX\" }",
    "config_options": "admin_ui=yes",
    "cpu_usage": "0",
    "stm_feature_pack": "STM-U-CSP-400-FP",
    "bandwidth": 100,
    "tag": "",
    "rest_enabled": true,
    "license_name": "flexlic1",
    "management_address": "1m1.mydomain.com"
}
```

In this example:

- XX.XX.XX.XX is the IP address of your gateway.

- When the **container_name** and **container_configuration** properties are used, the **container_name_json** and **container_configuration_json** properties are not required. See Specifying Configuration Options and Container Options on page 60.

- **1c1.mydomain.com** is the LXC container. This property is a fully-qualified domain name, or an IP address where Universal Licensing is used.

- **1m1.mydomain.com** is the management host. This is the same as 1c1.mydomain.com.

- **tag** is an optional text property which provides an alternative way of referring to an instance in the URI. In this example, the fixed (permanent) resource name of **stm1** is sufficient, and the tag is not set. You can set the tag (and use it to refer to the instance) later if required. See Understanding the Tag Property on page 125.

The REST API response indicates that the instance is scheduled to deploy.

3. A check of the FLA license is performed automatically, unless prevented in step 2 by the **?override_fla_check=true** URL parameter.

- If the license check fails, this procedure ends. The instance has a status of Failed to Deploy, and a deployment action shows as Blocked.

- If the license check succeeds, this procedure continues.

See Checking the Health of an FLA License Automatically on page 27.

4. To poll the Services Director until the instance is successfully deployed, perform a GET request for the URI of the instance that you created. For example:

    ```
    https://<sdhost>:<port>/api/tmcm/2.5/instance/stm1
    ```

    The response to this request contains a JSON structure representing the **instance** resource. This contains additional properties, one of which gives the status of the instance.

    > **NOTE**
    > If the deployment fails, you will first need to identify and resolve the the issue identified in the **block_reason** property of the associated deployment action. Then, you must remove the installation directory for the instance. You are then able to re-attempt the deployment by setting the Blocked deployment action to a Waiting state.

Working with Deployed Traffic Managers

Making Database-Only Updates

The Services Director uses the inventory database to store and maintain information about the state of each Traffic Manager instance it is aware of. This information includes the current status of each instance. For example, Inactive or Active.

However, the Services Director does not actively monitor this state. If you start or stop a Traffic Manager instance outside of the Services Director, it is unaware of this change of state.

You can use these techniques to resolve this monitoring issue:

*   Issue a GET REST API request for the Traffic Manager**instance** resource, including the URL parameter **status_check=true**. The Services Director actively checks the state of the Traffic Manager instance and updates the stored status where it can determine that the stored state is incorrect.

*   Issue a PUT REST API request to modify the Traffic Manager**instance** resource and set the **status** property to the known correct state with URL parameter **deploy=false**. The Services Director updates the status of the Traffic Manager instance in the inventory database but does not attempt to start or stop the instance itself.

You can also use a database-only update if you need to update the recorded admin user password. This action is useful if the password has been set or changed on the Traffic Manager instance directly.

Using an Instance Host VA with a Services Director VA

This section describes how to deploy Traffic Managers to an instance host VA from a Services Director VA.

> **NOTE**
> Other form-factor configurations are supported:

* To deploy Traffic Managers to an instance host machine from a software form-factor Services Director, see Using an Instance Host Machine with a Software Services Director on page 47.
* To deploy Traffic Managers to an instance host machine from a Services Director VA, see Using an Instance Host Machine with a Services Director VA on page 97
* To register externally-deployed Traffic Managers, see Registering Externally-Deployed Traffic Managers on page 117.

See also Services Director Form-Factors on page 14 and Instance Host Form-Factors on page 15.

Prerequisites

Resource Requirements for Instance Host Virtual Appliances

The Services Director VA requires the following resources for the Instance Host Virtual Appliance. You can deploy flavors (which are different sizes) of instance host VMs.

Instance Host VA Type	CPU	Memory	Disk
Small Flavor Instance Host VA	2 vCPU	4 GB	70 GB
Medium Flavor Instance Host VA	4 vCPU	8 GB	140 GB
Large Flavor Instance Host VA	8 vCPU	16 GB	280 GB

Required Software

To install the instance host, the following software is required:

Software	Description
Instance Host OVA	Uploads the Instance Host OVA file and creates an instance host. Make sure this file is located on an accessible location in your infrastructure.

Installing the Instance Host VA

This section describes the installation of the Instance Host VA for the Services Director VA. If you want to install a custom (non-VA) instance host, see Using an Instance Host Machine with a Services Director VA on page 97.

Perform the following steps:

1. Obtain the latest OVA for the Instance Host.

2. Install the Instance Host OVA via vSphere, using settings that are similar to those used for your Services Director VA.

> **NOTE**
> All networks can be set to the same setting.

3. The Instance Host VM will have an initial IP, but you will have a target IP to switch it to.

   After the instance host has been installed, you must configure it.

Configuring an Instance Host VA from a Terminal Session

This section describes the configuration of the Instance Host VA for the Services Director. If you want to configure your own instance host machine, see Using an Instance Host Machine with a Services Director VA on page 97

Once the Instance Host VA has been installed (see Installing the Instance Host VA on page 68), you must configure it from a terminal session (such as puTTy). To do this, you will need:

- the IP address assigned to the host instance, see Installing the Instance Host VA on page 68.
- the management IP address for the instance (eth0).
- the IP address for the outward-facing network (eth1). For example, your WAN.
- the IP address for the back-end servers (eth2). For example, your LAN. This is optional.
- the IP address(es) of the DNS name server(s).

To Configure Management Front-Facing and Rear-Facing Interfaces

1. Start a terminal session for the Services Director. Login as the **admin** user.

   ```
   Using username "admin".
   Brocade Services Director
   Last login: Thu Sep 10 14:25:33 2015 from 10.62.135.88
   ```

2. In the Services Director terminal session, start a CLI session:

   ```
   amnesiac > enable
   amnesiac # configure terminal
   amnesiac (config) #
   ```

3. Ping the new instance host.

```
amnesiac (config) # ping <IP_address>
64 bytes from <IP_address>: icmp_seq=4 ttl-64 time=0.761 ms
64 bytes from <IP_address>: icmp_seq=4 ttl-64 time=0.761 ms
```

4. Start a terminal session for the new instance host.

   Use the **sscadmin** user (for REST API access). The default password is **password**.

5. In the instance host, display host information:

```
$ show host info
+-------------+--------------------------------+
| Field       | Value                          |
+-------------+--------------------------------+
| hostname    | Brocade-Services-Director-Host |
| domain_name |                                |
+-------------+--------------------------------+
```

6. Change the name of the host if required:

   • For a DNS-based system, set the hostname and domain to match your DNS entry.

```
$ host hostname <new_hostname> domain-name <domain>
Updated host_info
```

   • For an IP-based system, no change is required.

7. Change the management (eth0) IP address, netmask and gateway.

```
$ host interface eth0 ip <ip-addr> netmask <netmask> gateway <ip-gateway> auto
```

   Your terminal session will be lost at this point, as the IP address of the instance host has changed.

8. Start a new terminal session for the instance host using its new management (eth0) IP address.

9. Change the front-facing (eth1) and the rear-facing (eth2) IP addresses.

```
$ host interface eth1 ip <ip-addr> netmask <netmask> gateway <ip-gateway> auto
Updated interface (eth1)
$ host interface eth2 ip <ip-addr> netmask <netmask> gateway <ip-gateway> auto
Updated interface (eth2)
```

   For example, eth1 may be the IP address of your WAN, and eth2 the IP address of your LAN.

To Configure DNS on the Instance Host (DNS-based networks only)

In a DNS-based network, check whether DNS has been set up, and set up if required.

1. In a terminal session for the instance host session, check DNS:

```
$ show host dns
+------------------+-------+
| Field            | Value |
+------------------+-------+
| nameservers      |       |
| dhcp_nameservers |       |
| dns_search       |       |
+------------------+-------+
```

   In this example, DNS is not defined.

2. Define DNS to point to your name servers.

```
$ host dns dns-nameservers <IP addresses> dns-search <domain>
```

To Enable Passwordless SSH Communication

You must now create a public SSH key to enable passwordless communication between the Services Director and the instance host. This SSH key will be used for all deployed instance hosts.

1. In a terminal session for the Primary Services Director, create an SSH public key for the administrator user to perform passwordless communication to the instance host.

```
amnesiac (config) # show ssh client private
No user identities configured.
SSH authorized keys:
amnesiac (config) # ssh client generate identity user admin
amnesiac (config) # show ssh client private
User Identities:
     User admin:
>>ssh public and private keys are displayed
```

2. In a terminal session for the instance host (user **sscadmin**), inject the SSH public key for the Services Director's **admin** user into the instance host:

```
$ user root sshkey "ssh-rsa <public_key> admin"
Added public SSH key for user (root)
```

3. Repeat steps 1) and 2) for the Secondary Services Director.

To Enable Deployment of Traffic Managers on IP-Based Systems

The deployment of Traffic Managers on IP-based systems (with no configured DNS) requires an additional configuration stage.

NOTE

This method supports Traffic Managers at version 10.1 and above with REST API enabled.

Each Traffic Manager is deployed into its own container, and an FQDN is required for each container. On an IP-based system, these FQDNs do not exist.

You can create a mapping between a container's IP address and a unique placeholder FQDN. A mapping is required for each Traffic Manager that you wish to deploy. For example:

- container1.aaa.bbb.ccc maps to 10.62.167.192
- container2.aaa.bbb.ccc maps to 10.62.167.193

Additionally, a mapping between your instance host's IP address and a unique placeholder FQDN is required.

The following procedure describes how to establish these mappings.

1. Start a terminal session on the Services Director. Use the **admin** user.

2. View any current mappings:

```
amnesiac (config) # show hosts
Hostname: amnesiac
No name servers active.
No domain names active.
IP 127.0.0.1 maps to hostname localhost
IP ::1 maps to hostname localhost
```

3. Establish a mapping for your containers. For example:

```
amnesiac (config) # ip host container1.aaa.bbb.ccc 10.62.167.192
amnesiac (config) # ip host container2.aaa.bbb.ccc 10.62.167.193
amnesiac (config) # ip host instancehost.aaa.bbb.ccc 10.62.167.194
```

4. Check these mappings:

```
amnesiac (config) # show hosts
Hostname: amnesiac
No name servers active.
No domain names active.
IP 10.62.167.192 maps to hostname container1.aaa.bbb.ccc
IP 10.62.167.193 maps to hostname container2.aaa.bbb.ccc
IP 10.62.167.194 maps to hostname instancehost.aaa.bbb.ccc
IP 127.0.0.1 maps to hostname localhost
IP ::1 maps to hostname localhost
```

If you need to remove a mapping, use the **no ip host <container> <ip address>** command. For example:

```
amnesiac (config) # no ip host container1.aaa.bbb.ccc 10.62.167.192
```

5. From your Services Director terminal session, start a session for the **root** user on your instance host. This requires no password from your Services Director, as passwordless SSH has been set up (see To Enable Passwordless SSH Communication on page 70).

```
amnesiac (config) # slogin root@<instance host IP address>
```

6. From this **root** session, view the current /etc/hosts file:

```
root@Brocade-Services-Director-Host:/# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       Brocade-Services-Director-Host
```

7. Edit the /etc/hosts files to add in the required mappings, using a text editor such as **vi** or **nano**.

8. Confirm the changes to the /etc/hosts file:

```
root@Brocade-Services-Director-Host:/# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       Brocade-Services-Director-Host
10.62.167.192   container1.aaa.bbb.ccc
10.62.167.193   container2.aaa.bbb.ccc
```

9. Close the **root** session to return to the Services Director CLI session.

The placeholder FQDNs are now established. These are used on the **Add vTM Instance** pop-up dialog box (see "To Deploy a Traffic Manager" on page 67).

Adding a Configured Instance Host to the Services Director VA

Once you have configured your instance host (see Configuring an Instance Host VA from a Terminal Session on page 68), you can add it to the Services Director VA.

To Add a Configured Instance Host

1. Start your Services Director VA from a browser, accessing it using its Service Endpoint IP Address.

2. Log in as the administration user. The **Home** page appears.

3. Click the **Services** menu, and then click **Services Director: vTM Instance Hosts**. The **vTM Instance Hosts** page appears.

FIGURE 6 vTM Instance Hosts Page
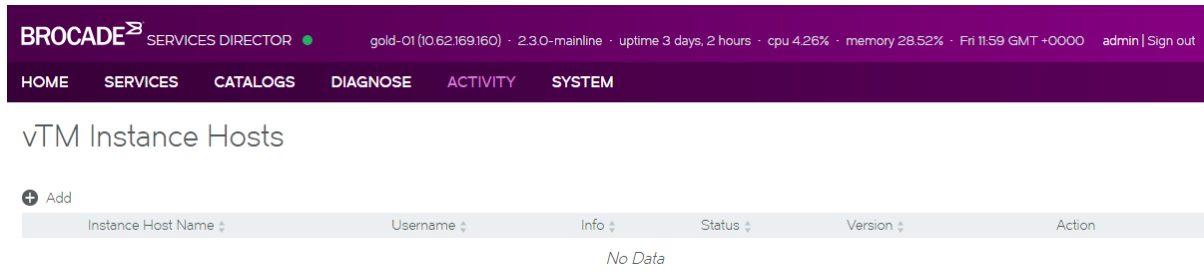
In this example, no instance hosts are present.

4. Click the **Add** button above the instance host table. A pop-up window appears.
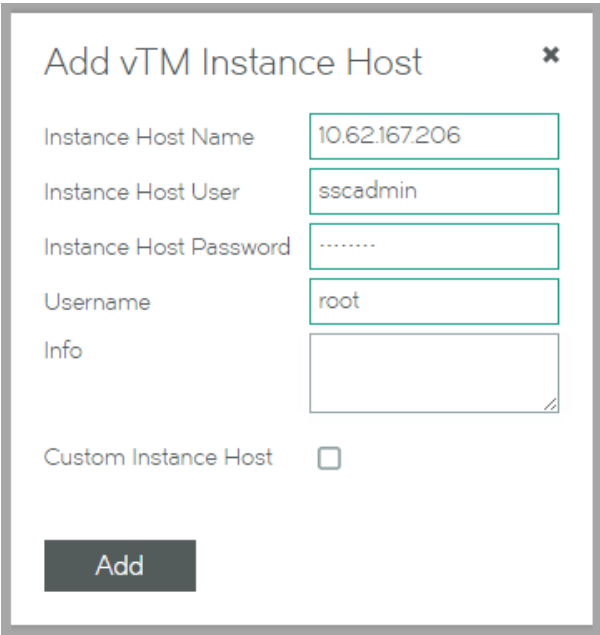
FIGURE 7 vTM Instance Hosts Page: Specify the Instance Host

5. Complete the fields:

    - **Instance Host Name** – the FQDN or IP address of the instance host.

    - **Instance Host User** – this is always **sscadmin**.

    - **Instance Host Password** – this is the password (default **password**) for the **sscadmin** user.

    - **Instance User** – this is always **root**.

    - Ensure that the **Custom Instance Host** check box is clear.
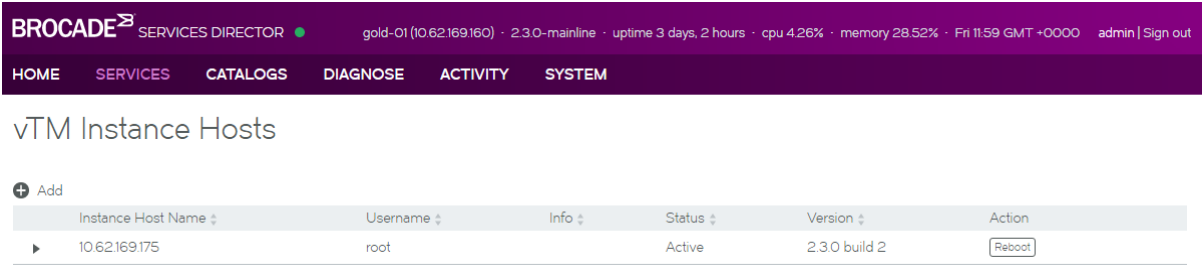
    For example:

    FIGURE 8 vTM Instance Hosts Page: Instance Host Specified

6. Click **Add**. The instance host is added to the **vTM Instance Hosts** page.

    FIGURE 9 Instance Hosts Page: Instance Host Added

Creating Required Resources

You must create required Legacy FLA licenses, Feature Packs and vTM Clusters using the Services Director VA.

> **NOTE**
> You must also create a Version resource, but that is created when you upload a Traffic Manager image to the Services Director VA, see Uploading a Traffic Manager Image on page 82.

To Create a Feature Pack Resource

Before you register a Traffic Manager, you must define a Feature Pack for the Traffic Manager.

When you install a primary Services Director VA, you provide a software license. This license is based on a single Stock-Keeping Unit (SKU).

Your choice of SKU determines whether you are a Cloud Service Provider (CSP) or Enterprise customer. It also defines the features that are available on your Services Director VA.

A SKU is used when you create a Feature Pack. Each Feature Pack is a subset (or total set) of the available features in a SKU.

To create a Feature Pack, you must select your SKU (historical SKUs that are still supported are also listed) and then identify any features from your SKU that are excluded in your Feature Pack. All other features supported by your SKU are included in the Feature Pack.

When you register a Traffic Manager, you select a single Feature Pack.

A default Feature Pack (typically a SKU with no exclusions) is created automatically when you install the Services Director VA.

1. Access your Services Director VA from a browser, using its Service Endpoint IP Address.
2. Log in as the administration user. The **Home** page appears.
3. Click the **Catalogs** menu, and then click **SKUS and Feature Packs**. The **SKUS and Feature Packs** page appears.

   FIGURE 10 SKUS and Feature Packs Page: Typical Configuration

SKUs and Feature Packs

Feature Packs
➕ Add

Feature Pack Name	SKU	Add-on SKUs	Status	Info
▶ STM-400_full	STM-400		Active	

SKUs
Show only compatible SKUs ☑

SKU Name	Details	Compatible	Status
▶ STM-100		✔	Active
▶ STM-200		✔	Active
▶ STM-300		✔	Active
▶ STM-400		✔	Active
▶ STM-WAFPROXY		✔	Active

4. In the table of SKUs, locate the SKU from which you wish to create a Feature Pack.

5. Expand this SKU to see its supported features. For example, the STM-400 SKU:

   FIGURE 11 SKUS and Feature Packs Page: Expanded SKU

6. Locate the function(s) that you wish to exclude, and make a note of the feature name. For example, the Lbrnd (Random Load Balancing) feature. That is, this Feature Pack will not support the Random load balancing feature. Other load balancing features, such as Round Robin, will still be supported.
7. Collapse the SKU in the table.

8. Click the **Add** button. A popup appears.

   FIGURE 12 SKUS and Feature Packs Page: Add Feature Pack

9. Enter a **Feature Pack Name**.

   This name will appear in the table of Feature Packs.

10. Select the required feature tier from your SKU.

11. Enter a space-separated list of excluded features.

12. Select any required add-on SKUs.

13. Enter a description for the Feature pack as **Info**.

   This description will appear in the table of Feature Packs.

   FIGURE 13 SKUS and Feature Packs Page: Specify New Feature Pack

14. Click **Add**. The new Feature Pack is added to the table of Feature Packs.

   FIGURE 14 SKUS and Feature Packs Page: New Feature Pack Added

15. Expand the Feature Pack to see its full details.

FIGURE 15 SKUS and Feature Packs Page: Full Details

16. Repeat this process to create all required Feature Packs.

    Once you have created all required Feature Packs, you can use these to register and deploy Traffic Managers.

To Create a Legacy FLA Resource

The Services Director comes with a pre-installed *Universal FLA License*. This is suitable for any Traffic Manager at version 10.1 with an active REST API. In all other cases, a *Legacy FLA License* is required. That is:

- The Traffic Manager version is 10.0 or earlier.
- The Traffic Manager (any version) has its REST API disabled.

You can install a Legacy FLA License using the Services Director VA, after which you can install either of these Traffic Manager types.

1. Access your Services Director VA from a browser, using its Service Endpoint IP Address.
2. Log in as the administration user. The **Home** page appears.

3. Click the **Catalogs** menu, and then click **Licensing: FLA Licenses**. The **Flexible Licenses** page appears.

> NOTE
> When the Services Director is first installed, only the pre-installed Universal FLA License is shown on this page; no Legacy FLA Licenses are present.

FIGURE 16 Flexible Licenses Page: No Legacy FLA
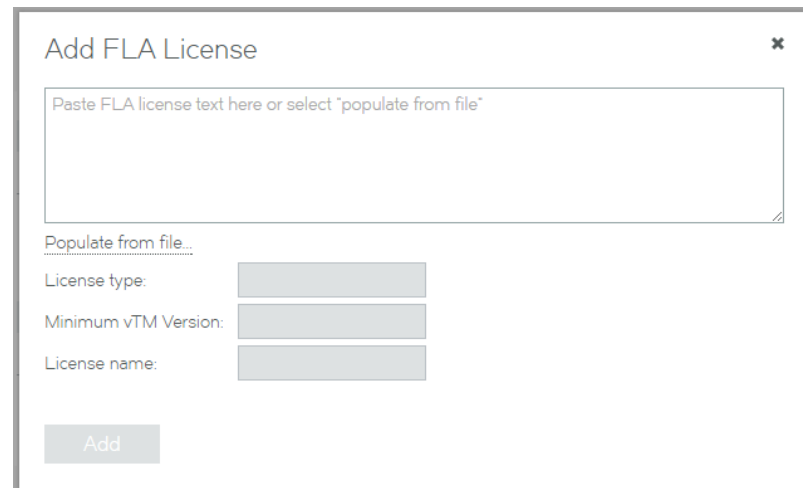
4. Click the **Add License** plus symbol. A licensing pop-up window appears.

FIGURE 17 Add FLA License Page

5. Either:

    •   –   Paste the text of the Legacy FLA License into the text box, OR

        –   Click **Populate from File**, select the file and then click **Upload**. This will populate the text box.

The remainder of the fields in the dialog will then update to provide license information:

FIGURE 18 License Information

6. Click **Add**. A relicensing pop-up appears.

7. Click **Later**.

The new license is added to the **FLA Licenses** page.

FIGURE 19 Flexible Licenses Page: Legacy FLA Added

8. Repeat this procedure if you require additional licenses.

9. Both Legacy FLA Licenses and Universal FLA Licenses have a default FLA. If you have more than one FLA license for either type, and want to make it the default license for that type, click **Make Default**.

Creating a vTM Cluster

The **vTM Cluster** page displays a list of all clusters known to the Services Director VA.

The **vTM Cluster** page also enables you to assign a backup schedule to each cluster, and to inspect the details of the cluster backups taken. See the *Brocade Services Director Getting Started Guide*.

FIGURE 20 vTM Cluster Page

vTM Clusters

✚ Add

	Cluster Name ▲	Cluster Port Offset ⇕	Type ⇕	In Use ⇕	Backup Schedule ⇕	Next Backup Time ⇕	Action	Last Action ⇕	Last Action Status
▶	Cluster-AQJE-R4HV-QYR1-9F4O	N/A	Discovered	✔	sched-hourly-01	2016-07-03 08:30:00	Backup Now		
▶	Cerise-Cluster	N/A	Discovered	✔	N/A		Backup Now		
▶	Cluster-RNPP-UIP9-RUA7-Q2JU	N/A	Discovered	✔	N/A		Backup Now		
▶	Violet-Cluster		User Created		N/A		Backup Now		

There are two types of clusters used by the Services Director VA:

- *Discovered* - this is a cluster present on one or more externally-deployed Traffic Managers. When an externally-deployed Traffic Manager is registered, a cluster name is displayed automatically. See *Brocade Services Director Getting Started Guide* for details.

  This cluster type *cannot* be used by Traffic Managers that you deploy from the Services Director VA.

  > **NOTE**
  > You cannot create a Discovered cluster from the **vTM Clusters** page.

  > **NOTE**
  > Services Director's awareness of Discovered clusters is limited to Traffic Managers at version 10.2 or above with an enabled REST API.

- *User Created* - this is a cluster that you create manually on the **vTM Clusters** page. This cluster type can *only* be used for Traffic Managers that you deploy from the Services Director VA.

- User Created - this is a cluster name that you create manually on the **vTM Clusters** page. *User Created* - this is a cluster that you create manually on the **vTM Clusters** page. This cluster type can *only* be used for Traffic Managers that you deploy from the Services Director VA.

  > **NOTE**
  > You must select a cluster when you deploy a Traffic Manager if you want to use cluster-level backup and restore. See the *Brocade Services Director Getting Started Guide* for full details.

Services Director supports backup and restore for cluster configurations.

To Create a vTM Cluster Name

1. Start your Services Director VA from a browser, accessing it using its Service Endpoint IP Address.
2. Log in as the administration user. The **Home** page appears.
3. Click the **Services** menu, and then click **Services Director: vTM Clusters**. The **vTM Clusters** page appears.

4. Click the **Add** button above the cluster table. A cluster pop-up window appears.

    FIGURE 21 Creating a vTM cluster name

5. Provide a vTM **Cluster Name**. This will be available as the **vTM Cluster ID** when you deploy a Traffic Manager. This name must be unique within the current subnet.

6. Enter an **Owner**.

7. Specify a **Cluster Port Offset**. This offset must be unique within the current subnet.

8. Optionally, specify a **Backup Schedule**.

9. Click **Add**. The new User Created cluster is added.

    FIGURE 22 vTM Cluster Page: New vTM Cluster Added

10. Repeat for any other required clusters.

    All User Created clusters are available for use as the **vTM Cluster ID** when you deploy a Traffic Manager. See

Uploading a Traffic Manager Image

Before you can deploy a Traffic Manager, you must upload the required Traffic Manager image onto the Services Director VA. During this process, you also specify the required version resource for the image.

1. Access your Services Director VA from a browser, using its Service Endpoint IP Address.

2. Log in as the administration user. The **Home** page appears.

3. Click the **Catalogs** menu, and then click **vTM Images**.

    The **vTM Images** page appears.

4. Click the plus symbol above the table of images. A pop-up window appears:

    FIGURE 23 Adding a vTM Image

5. Click **From file** and then **Choose File**. Then, select the required local Traffic Manager image file. This is in TGZ format.

6. Specify the required **Version Name**. Typically, this is the version number, for example 11.0.

7. Optionally, complete the Info field.

8. Click **Apply**.
    The image file is uploaded to the Services Director VA. Wgen complete, it is displayed in the **vTM Images** page. This may take several minutes.

Deploying a Traffic Manager from the Services Director VA

Once the instance host is added, it appears as a configured instance host on the **Instance Hosts** page of the Services Director VA.

You must create any required User Created clusters before deploying Traffic Managers from the Services Director VA.

> **NOTE**
> You must select a cluster when you deploy a Traffic Manager if you want to use cluster-level backup and restore. See the *Brocade Services Director Getting Started Guide* for full details.

You can then install a vTM image onto the Services Director VA as a .TGZ file, and deploy it Services Director VA using the instance host.

To View an Instance Host

1. Start your Services Director VA from a browser, accessing it using its Service Endpoint IP Address.

2. Log in as the administration user. The **Home** page appears.

3. Click the **Services** menu, and then click **Services Director: Instance Hosts**. The **Instance Hosts** page appears.

FIGURE 24 Instance Hosts Page: Basic Details
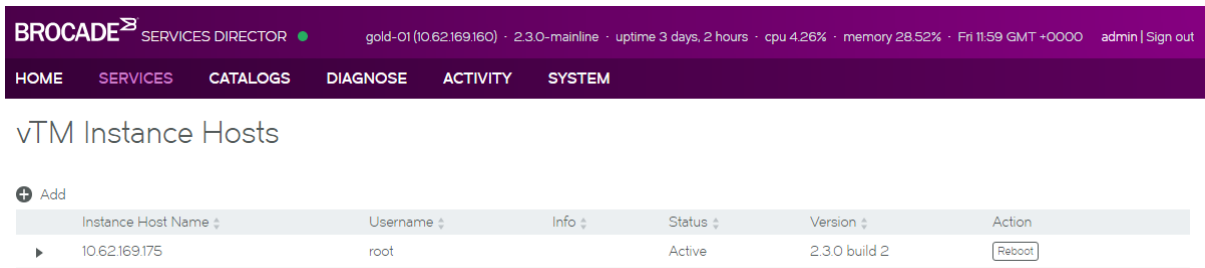
To Add a vTM Image

Use this procedure to add a new vTM image.

> **NOTE**
> Also use this procedure to replace a vTM image that is showing a warning message. For example, when the file is missing after a restore operation. In this instance, specify the same **Version Name** for the image; the warning message will clear once the image is reloaded. See *Brocade Services Director Getting Started Guide* for details of restore operations.

1. Click the **Catalogs** menu, and then click **vTM Images**. The **vTM Images** page appears.

2. Click the **Add** button above the vTM images table. A popup appears.

FIGURE 25 Adding a vTM Image
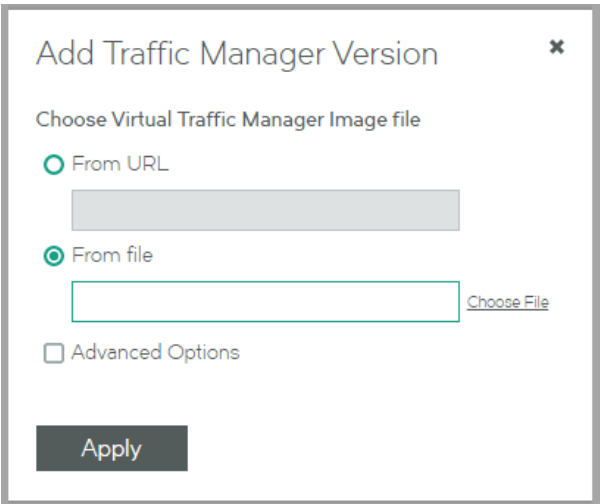
3. Select **From file** and click **Choose File**.

4. Select the required vTM image (.TGZ) file and click **Open**.

5. Select the **Advanced Options** check box.

FIGURE 26 Adding a vTM: Advanced Options

6. Add a **Version Name** that matches the software version. For example: "10.3".

> NOTE
> If you are replacing a vTM image after a restore operation, use the same **Version Name** as the image that is showing the warning.

7. Optionally, add any reference notes in the **Info** field.

8. Click **Apply**. The image is added.

FIGURE 27 vTM Images: Image Added

> NOTE
> If you are replacing a vTM image after a restore operation, the warning associated with the message clears.

Once the image is added, you can deploy a Traffic Manager.

To Deploy a Traffic Manager

1. Click the **Services** menu, and then click **Services Director: vTM Instances**. The **vTM Instances** page appears.

2. Click the **Add** button above the vTM instances table. A wizard appears.

    FIGURE 28 Deploying an Traffic Manager Instance: Starting Page

3. Select **Deploy an instance to a container** and click **OK**. The next page of the wizard is displayed.

    FIGURE 29 Deploying a Traffic Manager Instance: Defining Instance Details

    NOTE
    Click **Revert** to return this page of the wizard to its default state at any point.

4. Select the **Instance Hostname** for this deployment. This is the FQDN of the instance host.

5. Enter an **Instance Name** for the Traffic Manager instance.

   This is a user-facing name for the instance that will be used throughout the Services Director VA user interface. This tag can be changed at any time. It must be unique among non-deleted Traffic Manager instances registered on the Services Director, but can be reused as required.

   That is, if an instance is deleted, its tag can be reused for a different instance.

   It does not have to be the assigned DNS name for the IP address.

6. Enter a numeric **Bandwidth** (in Mbps) for the Traffic Manager instance.

   This bandwidth must be available within your Services Director's Bandwidth License.

7. Optionally, specify a **CPU Usage**.

   Specify a string that describes the CPUs that is used for the instance. The value is used by the **taskset** command and is typically a single CPU number.

8. Enter an **Owner** for the Traffic Manager instance.

9. Select the correct FLA **License Name** :

   - For software versions 10.1 and above, this will be a Universal FLA License.

   - For software versions 10.0 and below, this will be a Legacy FLA License.

   The selected FLA License must be the same for all Traffic Managers in a vTM cluster.

   > **NOTE**
   > The REST API is always enabled on deployed Traffic Manager instances.

10. Select a **Feature Pack** for the Traffic Manager instance. This must be supported

11. Select the **vTM Version** for the Traffic Manager instance.

    This is the **Version Name** specified when you added in the Instance Host to the Services Director VA.

12. Enter the **Container Name** for the Traffic Manager instance.

    - In DNS-based systems, this is the FQDN of the container that you wish to use for the deployed Traffic Manager instance.

    - In IP-based systems, this is the placeholder FQDN for the container that you wish to use for the deployed Traffic Manager instance (see To Enable Deployment of Traffic Managers on IP-Based Systems on page 70).

    When you move the UI focus away from this field, the FQDN resolves and indicates the assigned IP address. For example:

    FIGURE 30 Deploying a Traffic Manager Instance: Resolving the Container

    Container Name: | container1.aaa.bbb.ccc | (Resolves to 10.62.167.192)

    The resolved IP address is required for the **Mgmt IP/Subnet** field.

13. Select a **Flavor** for the Traffic Manager instance. This indicates the number of CPU cores and the available RAM. For example:

    - Small - Specifies two CPUs and 4 GB of memory. (default)

    - Medium - Specifies four CPUs and 8 GB of memory.

    - Large - Specifies eight CPUs and 16 GB of memory.

14. Under **Management Plane** , enter the **Gateway** IP address for your Traffic Manager instance.

    This is the same as the gateway specified when you created your Services Director VA.

15. Under **Management Plane** , enter the **Mgmt IP/Subnet** for your Traffic Manager instance.

    This combines the management (eth0) IP address of your container with a CIDR subnet suffix. For example:

    • Your **Container Name** resolves to an IP address of 10.62.167.192. (see step 12)

    • Your netmask is 255.255.192.0, which is equivalent to a CIDR subnet suffix of "/18".

    The **Mgmt IP/Subnet** for the **Management Plane** is 10.62.167.192/18.

16. Under **Data Plane** , enter the **Gateway** IP address for your front-facing interface (eth1).

    > NOTE
    > Data plane addresses cannot be on the same subnet as management plane addresses.

17. Under **Data Plane** , enter the **Data1 IP/Subnet** for your front-facing interface.

    This combines the eth1 IP address of your container with a CIDR subnet suffix. For example:

    • The IP address of your front-facing interface (eth1) is 10.63.167.192.

    • Your netmask is 255.255.192.0, which is equivalent to a CIDR subnet suffix of "/18".

    The **Data1 IP/Subnet** for the **Data Plane** is 10.63.167.192/18.

18. Optionally, under **Data Plane** , enter the **Data2 IP/Subnet** for your rear-facing interface (eth2).

19. If you want to deploy the Traffic Manager to a vTM cluster, select the **Advanced Options** check box and select the required User Created cluster from the **vTM Cluster ID** list.

>   NOTE
>   User Created clusters are defined on the **vTM Cluster** page, and must exist before the Traffic Manager is deployed. See Creating a vTM Cluster on page 81.

>   NOTE
>   If you want to use cluster-level backup and restore, you must select a cluster at this point. See the *Brocade Services Director Getting Started Guide* for full details of cluster-level backup and restore.

>   NOTE
>   The FLA License must be the same for all Traffic Managers in a vTM cluster.

FIGURE 31 Deploying an Traffic Manager Instance: Ready to Deploy

>   NOTE
>   The **Extra Options** property is for optional advanced configuration options. These are described under the **config_options** property in Properties for a Deployed Instance on page 58.

20. Click **Add** to deploy the Traffic Manager instance.

The Traffic Manager instance is added to the **vTM Instances** page.

Initially, its **Lifecycle Status** is shown as the transitional state New. For example:

FIGURE 32 Deploying an Traffic Manager Instance: New Status

▸	viridian-01	legacy_9.3	150	STM-400-Full			Active	N/A ⚠	Licensed	N/A
▸	cobalt-02	universal_v3	100	STM-400-Full	10.2	Cluster-A1BQ-V577-V8NY-UIDZ	Active	N/A	Licensed	N/A
▸	indigo-01	universal_v3	75	STM-400-Full	10.3	TK-421	New	N/A	N/A	Loading...

The **Lifecycle Status** becomes Idle after deployment is complete.

FIGURE 33 Deploying an Traffic Manager Instance: Idle Status

▸	viridian-01	legacy_9.3	150	STM-400-Full			Active	N/A ⚠	Licensed	N/A
▸	cobalt-02	universal_v3	100	STM-400-Full	10.2	Cluster-A1BQ-V577-V8NY-UIDZ	Active	N/A	Licensed	N/A
▸	indigo-01	universal_v3	75	STM-400-Full	10.3	TK-421	Idle	N/A	N/A	Start

> **NOTE**
> The **Action** column for the Traffic Manager instance now has a **Start** button. This enables you to perform a state transition to the Active state. See Understanding Lifecycle Status (Deployed Traffic Managers) on page 92.

Starting a Deployed Traffic Manager

The **vTM Instances** page enables you to start and stop Traffic Managers that were deployed by the Services Director VA. The **Action** column includes a button to perform one of these actions.

For details of the state model for deployed Traffic Managers, see Understanding Lifecycle Status (Deployed Traffic Managers) on page 92.

Once you have deployed a Traffic Manager, its **Lifecycle Status** is Idle. From this state, you can start the Traffic Manager.

> **NOTE**
> You cannot start and stop externally-deployed Traffic Managers. See Understanding Lifecycle Status (Deployed Traffic Managers) on page 92.

To Start a Deployed Traffic Manager

1. Start your Services Director VA from a browser, accessing it using its Service Endpoint IP Address.
2. Log in as the administration user. The **Home** page appears.

3. Click the **Services** menu, and then click **Services Director: vTM Instance Hosts**. The **vTM Instance Hosts** page appears. For example:

FIGURE 34 Starting a Traffic Manager Instance: Idle Status

	viridian-01	legacy_9.3	150	STM-400-Full			Active	N/A ⚠	Licensed	N/A
▶	cobalt-02	universal_v3	100	STM-400-Full	10.2	Cluster-A1BQ-V577-V8NY-UIDZ	Active	N/A	Licensed	N/A
▶	indigo-01	universal_v3	75	STM-400-Full	10.3	TK-421	Idle	N/A	N/A	Start

4. Start an Idle Traffic Manager instance by clicking **Start** in its **Action** column.

   Initially, its **Lifecycle Status** is shown as the transitional state Starting.

FIGURE 35 Starting a Traffic Manager Instance: Starting Status

	viridian-01	legacy_9.3	150	STM-400-Full			Active	N/A ⚠	Licensed	N/A
▶	cobalt-02	universal_v3	100	STM-400-Full	10.2	Cluster-A1BQ-V577-V8NY-UIDZ	Active	N/A	Licensed	N/A
▶	indigo-01	universal_v3	75	STM-400-Full	10.3	TK-421	Starting	N/A	N/A	Stop

The **Lifecycle Status** becomes Active after it starts. At this point:

The licensing of the Traffic Manager is checked, and its **Licensing Health** becomes Pending. See Understanding the Licensing Health of a Traffic Manager on page 95.

The instance health is also updated.See Understanding the Licensing Health of a Traffic Manager on page 95.

FIGURE 36 Starting a Traffic Manager Instance: Active/Pending Status

	viridian-01	legacy_9.3	150	STM-400-Full			Active	N/A ⚠	Licensed	N/A
▶	cobalt-02	universal_v3	100	STM-400-Full	10.2	Cluster-A1BQ-V577-V8NY-UIDZ	Active	N/A	Licensed	N/A
▶	indigo-01	universal_v3	75	STM-400-Full	10.3	TK-421	Active	N/A	Pending	Stop

After the check is complete, the **Licensing Health** becomes Licensed.

FIGURE 37 Starting a Traffic Manager Instance: Active/Licensed Status

	viridian-01	legacy_9.3	150	STM-400-Full			Active	N/A ⚠	Licensed	N/A
▶	cobalt-02	universal_v3	100	STM-400-Full	10.2	Cluster-A1BQ-V577-V8NY-UIDZ	Active	N/A	Licensed	N/A
▶	indigo-01	universal_v3	75	STM-400-Full	10.3	TK-421	Active	OK	Licensed	Stop

The Traffic Manager instance is now deployed, running and licensed.

Stopping a Deployed Traffic Manager

The **vTM Instances** page enables you to start and stop Traffic Managers that were deployed by the Services Director VA. The **Action** column includes a button to perform one of these actions.

For details of the state model for deployed Traffic Managers, see Understanding Lifecycle Status (Deployed Traffic Managers) on page 92.

When a Traffic Manager is running, its **Lifecycle Status** is Active. From this state, you can stop the Traffic Manager.

> **NOTE**
> You cannot start and stop externally-deployed Traffic Managers. See the *Brocade Services Director Getting Started Guide*.

To Stop a Deployed Traffic Manager

1. Start your Services Director VA from a browser, accessing it using its Service Endpoint IP Address.

2. Log in as the administration user. The **Home** page appears.

3. Click the **Services** menu, and then click **Services Director: vTM Instance Hosts**. The **vTM Instance Hosts** page appears. For example:

FIGURE 38 Stopping a Traffic Manager Instance: Active Status

4. Stop an Active Traffic Manager instance by clicking **Stop** in its **Action** column.

   Initially, its **Lifecycle Status** is shown as the transitional state Stopping.

   The **Lifecycle Status** becomes Idle after it stops. Also:

   - – When licensing is next checked, the licensing of the Traffic Manager halts, and its **Licensing Health** becomes N/A. See Understanding the Licensing Health of a Traffic Manager on page 95.
   - – When monitoring is next performed, the Instance Health becomes N/A. See Understanding the Instance Health of a Traffic Manager on page 94

FIGURE 39 Stopping a Traffic Manager Instance: Idle N/A Status

The Traffic Manager instance is now stopped.

Understanding Lifecycle Status (Deployed Traffic Managers)

The **Instance Lifecycle** state of each Traffic Manager is displayed in the **vTM Instances** page.

When you deploy a Traffic Manager from the Services Director VA, it is deployed into a *container*. This container enables full control of lifecycle operations for the Traffic Manager.

See The *Brocade Services Director Advanced User Guide*.

FIGURE 40 Lifecycle State Model: Traffic Managers Deployed by the Services Director VA
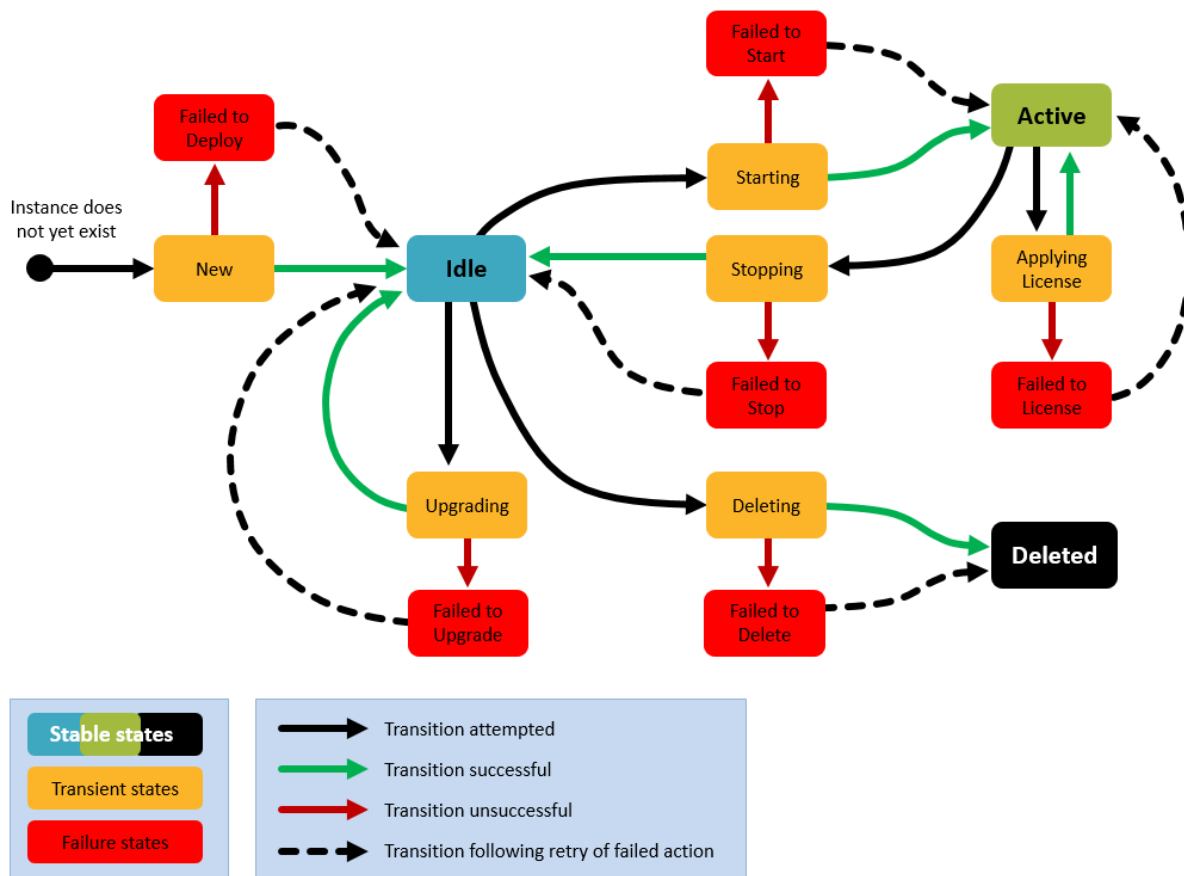
The **Lifecycle Status** column displays the result of the most recent state transition attempted from the Services Director.

> NOTE
> The **Lifecycle Status** column does *not* display a live monitoring status. As a result, if a Traffic Manager fails independently, this will not be indicated until a state transition is attempted.

FIGURE 41 Lifecycle Status Column: Traffic Managers Deployed by the Services Director VA

Active	A stable state
Idle	A stable state
(Transitional)	A transitional state, indicating that an operation is in progress
Failed	A transitional state, indicating that an operation has failed
Deleted	A stable state

Note that:

- where additional information is available, a red triangle is displayed next to the text of the status. Hover over this with your mouse to read the additional information in a pop-up.
- the displayed states are subject to the **Instance Status** filter, see the *Brocade Services Director Getting Started Guide*.

You can affect the **Lifecycle Status** of a deployed Traffic Manager as follows:

- By starting or stopping a deployed Traffic Manager using the **Start** and **Stop** buttons in the **Action** column. See Stopping a Deployed Traffic Manager on page 91.
- By deleting a Traffic Manager from its entry in the Traffic Manager table. See the *Brocade Services Director Getting Started Guide*.
- Other states are visible during initial deployment, upgrade and relicensing.

Understanding the Instance Health of a Traffic Manager

The **Instance Health** of each Traffic Manager is displayed in the **vTM Instances** page.

The displayed **Instance Health** of a Traffic Manager is a summary status that reflects the health of the *cluster* to which the Traffic Manager belongs.

Instance Health is reported as follows:

FIGURE 42 Instance Health Column

N/A (black)	The cluster status on the instance is either not being monitored or the instance is deleted
N/A (grey)	The Traffic Manager uses an older software version (pre-v10.3) which does not support this feature
N/A (blue)	The cluster status on the instance is configured, but is not available. For example, when its REST API is not enabled
Okay	The cluster is in a *Healthy* state. This matches the dashboard state on the Traffic Manager itself.
Warning	The cluster is in a *Warning* state. This matches the dashboard state on the Traffic Manager itself.
Error	The cluster is in a *Serious* state. This matches the dashboard state on the Traffic Manager itself.

Note that:

- instance health checks are only performed for Traffic Managers at version 10.3 or above with an active REST API. For all other cases, the **Instance Health** is reported as N/A.
- where additional information is available, a red triangle is displayed next to the text of the status. Hover over this with your mouse to read the additional information in a pop-up.
- the displayed states are subject to the **Instance Health** filter, see the *Brocade Services Director Getting Started Guide*.

Understanding the Licensing Health of a Traffic Manager

The **Licensing Health** of each Traffic Manager is displayed in the **vTM Instances** page.

The displayed **Licensing Health** of a Traffic Manager is a summary status, based on a number of licensing checks. Licensing is tested every three minutes using a callback mechanism. Testing varies, depending on whether a Universal FLA or Legacy FLA License is in use on a Traffic Manager.

Licensing Health is reported as follows:

FIGURE 43 Licensing Health Column

Licensed	The Traffic Manager has called back and been licensed in the past three minutes
Pending	The Traffic Manager has been installed, but has not yet called back
Grace Period	The licensed Traffic Manager has not called back for over three minutes (but less than six weeks)
Expired	The licensed Traffic Manager has not called back for over six weeks (the grace period has expired)
N/A	Not Applicable: the Traffic Manager is not Active, and licensing checks are not required.

Note that:

- license checks are only performed for Traffic Managers with an Active **Lifecycle Status**. For all other lifecycle states, the **Licensing Health** is reported as N/A.
- where additional information is available, a red triangle is displayed next to the text of the status. Hover over this with your mouse to read the additional information in a pop-up.
- the displayed states are subject to the **Licensing Health** filter, see the *Brocade Services Director Getting Started Guide*.

Using an Instance Host Machine with a Services Director VA

This section describes how to deploy Traffic Managers to an instance host machine from a Services Director VA.

> **NOTE**
> Other form-factor configurations are supported:
> - To deploy Traffic Managers to an instance host machine from a software form-factor Services Director, see Using an Instance Host Machine with a Software Services Director on page 47.
> - To deploy Traffic Managers to an instance host VA from a Services Director VA, see Using an Instance Host VA with a Services Director VA on page 67.
> - To register externally-deployed Traffic Managers, see Registering Externally-Deployed Traffic Managers on page 117.

See also Services Director Form-Factors on page 14 and Instance Host Form-Factors on page 15.

Overview: Using an Instance Host Machine with a Services Director VA

This section describes how to deploy Traffic Manager instances from a system that combines a Services Director VA with an *instance host* machine.

> **NOTE**
> The following Services Director VA functions are known to have issues when using an instance host machine, see Deploying a Traffic Manager Instance on page 114:
> - Deploying Traffic Managers. You must deploy from the REST API.
> - Editing properties of deployed instances. Only container properties can be edited through the GUI. For all other properties, you must use the CLI or REST API.
> - The **vTM Instance Host** page displays warning information about the limitations of instance host machines.

An instance host machine is an Ubuntu 14.04 machine (virtual or physical) that is configured specifically for the purpose of allowing Services Director to deploy instances to it.

> **NOTE**
> It is also possible to use a RedHat/CentOS machine in the same way.

NOTE

If you want an instance host machine that will support interaction with a software form-factor Services Director, see Using an Instance Host Machine with a Software Services Director on page 47.

This section assumes that you have already installed and configured your Services Director VA.

To create an instance host machine, the required steps are:

1. Create an Ubuntu 14.04 machine and its network interfaces, see To Create the Instance Host on page 101.

    NOTE

    Ensure that an SSH Server is present on the Ubuntu 14.04 machine.

2. On the Ubuntu 14.04 machine, set up the required directory structures on the Ubuntu 14.04 machine, see To Prepare the Directory Structure on page 102.

3. Set up local networking (interfaces/NICs and bridges) on the Ubuntu 14.04 machine, see To Configure the Local Network on page 102. Exact details will depend on your networking requirements (see example below).

4. On the Ubuntu 14.04 machine, set up an LXC configuration file for each required LXC container, see To Create an LXC Container Configuration File on page 103.

5. Establish SSH communications between the Services Director VA and the Ubuntu 14.04 machine, see Configuring Passwordless SSH on page 112.

6. Create any required resources (feature packs and legacy FLA if required) on the Services Director VA, see Creating Required Resources on page 104.

7. Create any required version resources on the Services Director VA by the upload of any required Virtual Traffic Manager (vTM) images, see Uploading a Traffic Manager Image on page 112.

8. Register the instance host on the Services Director VA, see Adding the Instance Host Machine to the Services Director VA on page 113.

For example:

FIGURE 44 Administration Tasks to Prepare the Instance Host and Services Director VA
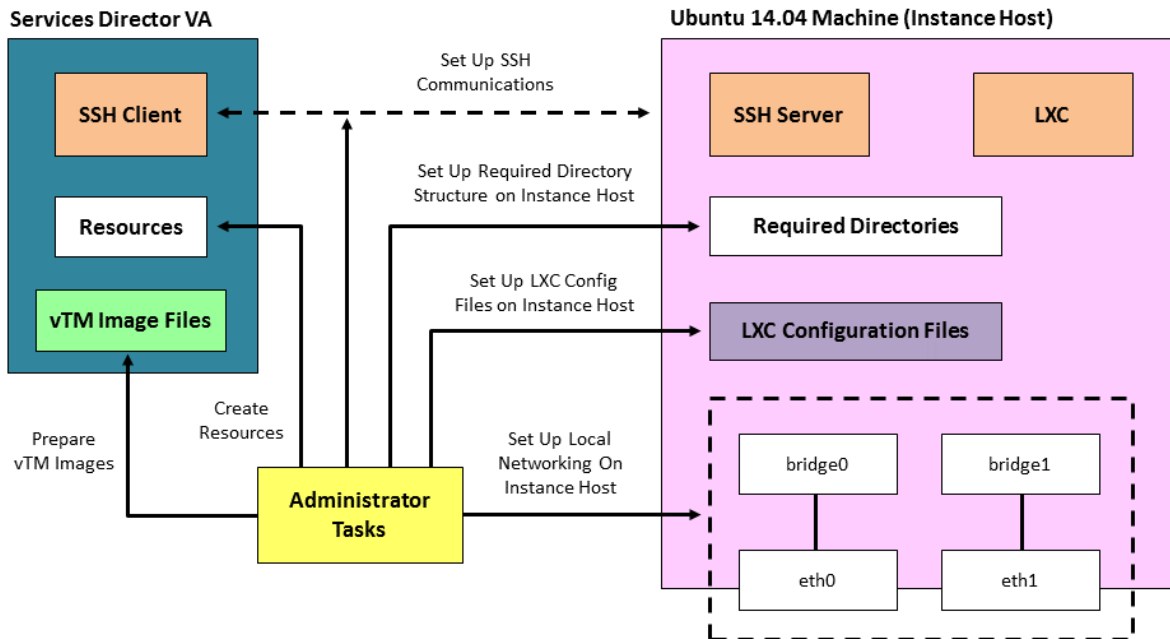
Once these steps are complete the instance host is ready.

To create a vTM on the instance host, request the deployment from the Services Director REST API, see Deploying a Traffic Manager Instance on page 114. The following events then occur automatically:

- The Services Director VA contacts the instance host using SSH to request that LXC deploys a vTM on the instance host.
- The Services Director VA transfers the required vTM image file to the instance host.
- On the instance host, LXC retrieves the required LXC configuration file based on the information passed from the Services Director VA.
- On the instance host, LXC uses the LXC configuration file to create an LXC container.
- The instance host uses the required vTM image to deploy a vTM to the LXC container, and then configures its interfaces.
- The new vTM is added to the list of vTMs that are in the estate of the Services Director VA.

Once these events are complete, the first vTM is ready for use.

FIGURE 45 Creating a vTM on an Instance Host from the Services Director VA
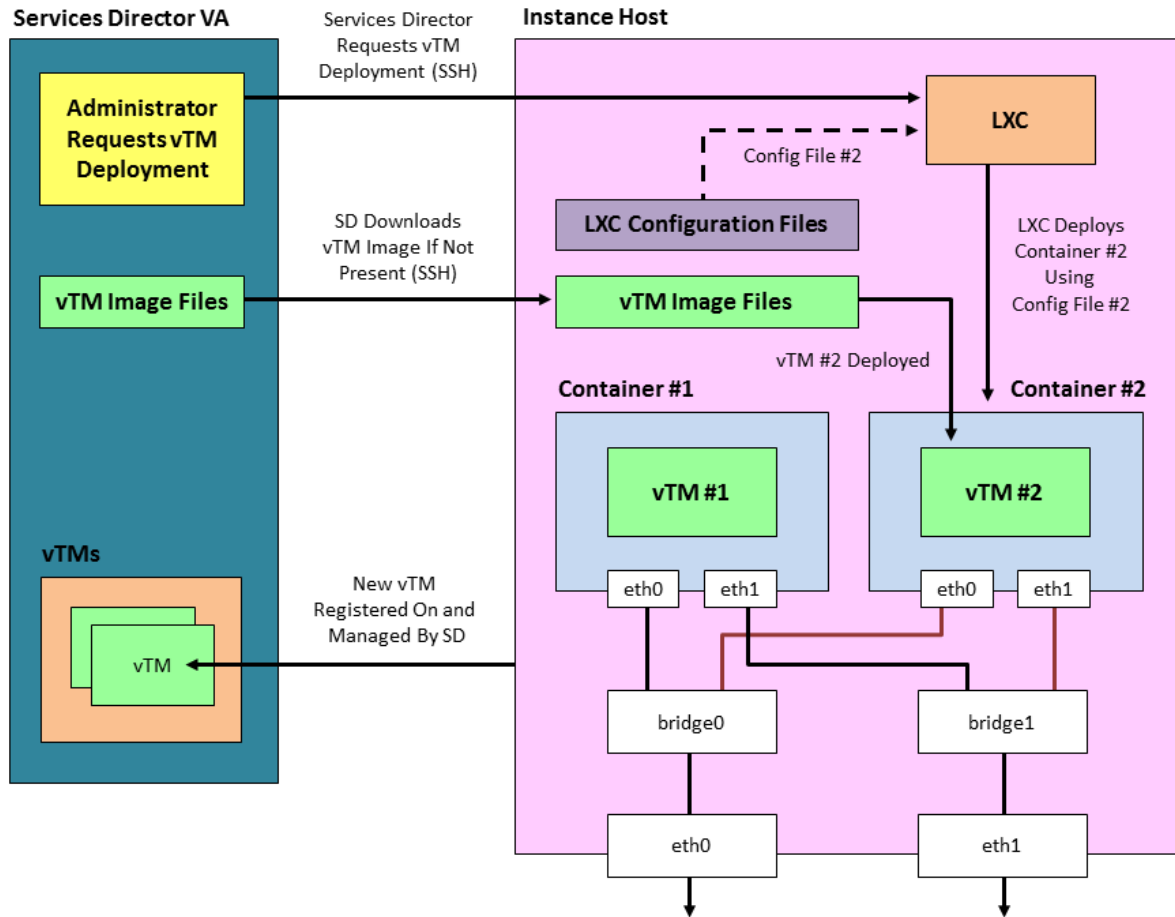
You can repeat this process to deploy a second (and subsequent) vTMs onto the same instance host. This is broadly similar to the first vTM. Please note that:

- This requires a different LXC configuration file.
- A second container configuration file must already exist or be created. You cannot deploy a second vTM to the same container as the first vTM.
- A vTM image file is only transferred to the instance host if the vTM image file is not already cached there.
- All other events are the same as for the first vTM.

FIGURE 46 Creating another vTM on an Instance Host from the Services Director VA

Creating and Configuring an Instance Host Machine

You can use an instance host machine (either virtual or physical) with both the software form-factor Services Director and the Services Director VA.

To Create the Instance Host

To create an instance host, you must first create an Ubuntu 14.04 machine. During this process, you will need:

- The IP address and hostname that is required for the instance host.
- The IP address for the management interface (eth0).
- Optionally, an IP address for each of the required network interfaces (eth1, eth2, and so on).
- Optionally, the IP address(es) of the DNS name server(s).

Perform the following steps:

1. Create a machine to contain the Ubuntu 14.04 operating system. This machine can be either virtual or physical.

   The machine's management network interface must be accessible from the Services Director

2. Install the Ubuntu 14.04 operating system using the standard set-up software.

3. Ensure that the Ubuntu 14.04 machine has SSH server installed and running.

4. Install LXC and bridge utilities on the Ubuntu 14.04 machine to enable container deployment:

```
apt-get install lxc bridge-utils
```

To Prepare the Directory Structure

You must prepare the Ubuntu 14.04 instance host directory structure to receive the required Traffic Manager files. These are required when you register the instance host on the Services Director.

1. On the instance host, create a directory to place files for use. For example, for Traffic Manager image files and license files. For example:

```
/var/cache/ssc
```

2. Then, create a parent directory for all deployed Traffic Managers. For example:

```
/root/install
```

3. Check that suitable permissions are set for both of the new directories (from steps 1 and 2).

   Services Director assumes that all required directories on the instance host have read/write/execute permissions for the root user / group. Check this on the instance host:

```
stat --format=%a <directory>
```

   And set if required by using:

```
chmod 770 <directory>
```

To Configure the Local Network

Configure a virtual network on the instance host for the Traffic Manager instance. Typically, this configuration requires setting up a virtual bridge (using, for example, Linux bridge or Open vSwitch), then configuring the Linux container to attach to the bridge when running Traffic Manager.

NOTE
You must disable the default bridge configured by LXC. This is unsuitable for Services Director operations.

1. On the instance host, edit the `/etc/network/interfaces` file and configure the network interfaces and bridges as required. For example:

```
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet static
    address XX.XX.XX.XX
    netmask YY.YY.YY.YY
    network ZZ.ZZ.ZZ.ZZ
    broadcast AA.AA.AA.AA
    gateway BB.BB.BB.BB
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers DD.DD.DD.DD
    dns-search cam.demo.com
# The primary bridge
auto br0
iface br0 inet static
    bridge_ports eth1
    bridge_fd 0
    address EE.EE.EE.EE
    netmask FF.FF.FF.FF
    network GG.GG.GG.GG
# The secondary network interface
auto eth1
iface eth1 inet static
    address 0.0.0.0
```

In this example, there is a Linux bridge setup to which LXC containers can attach virtual network interfaces.

2. You must now disable the LXC default bridge. Edit the `/etc/default/lxc-net` file and set the following:

```
USE_LXC_BRIDGE="false"
```

Also comment out any other lines that contain **'LXC_*='**.

3. You must now turn off strict return path filtering. Edit the `/etc/default/sysctl.conf` file:

```
net.ipv4.conf.default.rp_filter=2
net.ipv4.conf.all.rp_filter=2
```

4. Either reboot, or restart the networking services. This ensures that the networking configuration is correct and the default LXC bridge is removed.

To Create an LXC Container Configuration File

Traffic Manager instances can cohabit on an instance host machine using LXC containers.

LXC containers support network isolation and a degree of resource isolation. The software form-factor Services Director expects prepared container configuration files in advance of instance deployment. You must put these files in the installation root directory of the instance host.

Before you create an LXC container for a Traffic Manager, you must create an LXC configuration file. This file describes the properties for the container.

1. On the instance host, create an LXC configuration text file.

    This must be located in the parent directory for installations (see To Prepare the Directory Structure on page 51). For example, in the `/root/install` directory.

    The container configuration file must include networking entries for your Traffic Manager instance IP address. You can use either Linux virtual networking or Open vSwitch to set up network isolation for your Traffic Manager instance.

    The filename of the container configuration file must match the FQDN or IP address of the required container. That is: `<container-fqdn-or-ip>.conf`.

    > **NOTE**
    > This filename must be specified as the **container_name** property when you deploy the instance.

    > **NOTE**
    > For Instance resources that are not able to use the Universal FLA License, `<container-fqdn-or-ip>.conf` must match the value of the **lxc.utsname** used in the file, and the **management_address** specified in the REST request.

2. Create contents for the LXC container configuration file. For example:

    ```
    lxc.network.type = veth
    lxc.network.flags = up
    lxc.network.link = br0
    lxc.network.ipv4 = XX.XX.XX.XX/SS
    lxc.network.ipv4.gateway = YY.YY.YY.YY
    lxc.utsname = vtm-mgmt.example.com
    ```

    In this example:

    *   XX.XX.XX.XX/SS is the IP address and subnet mask of the LXC container network.
    *   YY.YY.YY.YY is the IP address of the subnetwork gateway inside the LXC container.
    *   If the Legacy FLA License is in use for this instance, the filename of the container configuration file must be `vtm-mgmt.example.com.conf`.

Creating Required Resources

You must create required Legacy FLA licenses and Feature Packs using the Services Director VA.

> **NOTE**
> You must also create a Version resource, but that is created when you upload a Traffic Manager image to the Services Director VA, see Uploading a Traffic Manager Image on page 112.

To Create a Feature Pack Resource

Before you register a Traffic Manager, you must define a Feature Pack for the Traffic Manager.

When you install a primary Services Director VA, you provide a software license. This license is based on a single Stock-Keeping Unit (SKU).

Your choice of SKU determines whether you are a Cloud Service Provider (CSP) or Enterprise customer. It also defines the features that are available on your Services Director VA.

A SKU is used when you create a Feature Pack. Each Feature Pack is a subset (or total set) of the available features in a SKU.

To create a Feature Pack, you must select your SKU (historical SKUs that are still supported are also listed) and then identify any features from your SKU that are excluded in your Feature Pack. All other features supported by your SKU are included in the Feature Pack.

When you register a Traffic Manager, you select a single Feature Pack.

A default Feature Pack (typically a SKU with no exclusions) is created automatically when you install the Services Director VA.

1. Access your Services Director VA from a browser, using its Service Endpoint IP Address.

2. Log in as the administration user. The **Home** page appears.

3. Click the **Catalogs** menu, and then click **SKUS and Feature Packs**. The **SKUS and Feature Packs** page appears.

FIGURE 47 SKUS and Feature Packs Page: Typical Configuration

4. In the table of SKUs, locate the SKU from which you wish to create a Feature Pack.

5. Expand this SKU to see its supported features. For example, the STM-400 SKU:

   FIGURE 48 SKUS and Feature Packs Page: Expanded SKU

6. Locate the function(s) that you wish to exclude, and make a note of the feature name. For example, the Lbrnd (Random Load Balancing) feature. That is, this Feature Pack will not support the Random load balancing feature. Other load balancing features, such as Round Robin, will still be supported.

7. Collapse the SKU in the table.

8. Click the **Add** button. A popup appears.

    FIGURE 49 SKUS and Feature Packs Page: Add Feature Pack

9. Enter a **Feature Pack Name**.

    This name will appear in the table of Feature Packs.

10. Select the required feature tier from your SKU.

11. Enter a space-separated list of excluded features.

12. Select any required add-on SKUs.

13. Enter a description for the Feature pack as **Info**.

    This description will appear in the table of Feature Packs.

    FIGURE 50 SKUS and Feature Packs Page: Specify New Feature Pack

14. Click **Add**. The new Feature Pack is added to the table of Feature Packs.

    FIGURE 51 SKUS and Feature Packs Page: New Feature Pack Added

15. Expand the Feature Pack to see its full details.

FIGURE 52 SKUS and Feature Packs Page: Full Details

16. Repeat this process to create all required Feature Packs.

    Once you have created all required Feature Packs, you can use these to register and deploy Traffic Managers.

To Create a Legacy FLA Resource

The Services Director comes with a pre-installed *Universal FLA License*. This is suitable for any Traffic Manager at version 10.1 with an active REST API. In all other cases, a *Legacy FLA License* is required. That is:

- The Traffic Manager version is 10.0 or earlier.
- The Traffic Manager (any version) has its REST API disabled.

You can install a Legacy FLA License using the Services Director VA, after which you can install either of these Traffic Manager types.

1. Access your Services Director VA from a browser, using its Service Endpoint IP Address.
2. Log in as the administration user. The **Home** page appears.

3. Click the **Catalogs** menu, and then click **Licensing: FLA Licenses**. The **Flexible Licenses** page appears.

   NOTE
   When the Services Director is first installed, only the pre-installed Universal FLA License is shown on this page; no Legacy FLA Licenses are present.

FIGURE 53 Flexible Licenses Page: No Legacy FLA
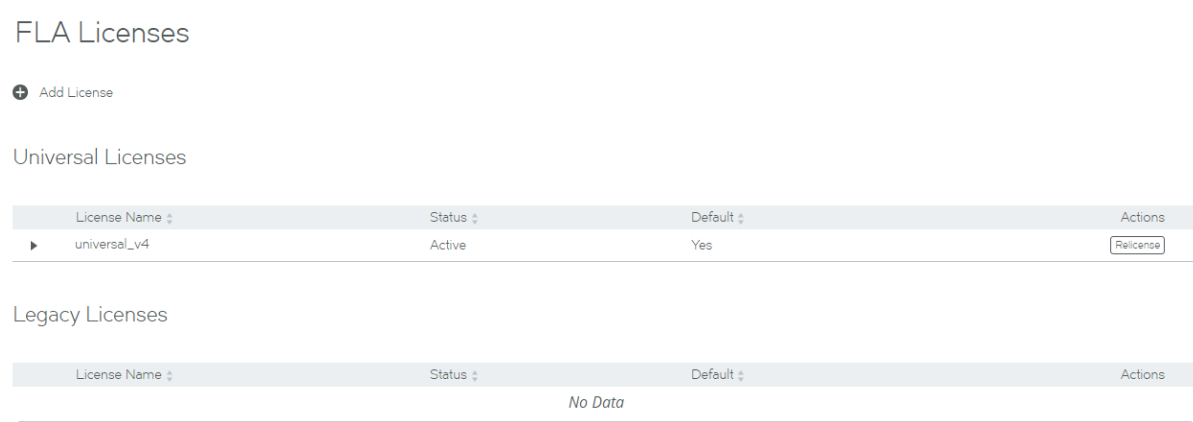
4. Click the **Add License** plus symbol. A licensing pop-up window appears.

FIGURE 54 Add FLA License Page
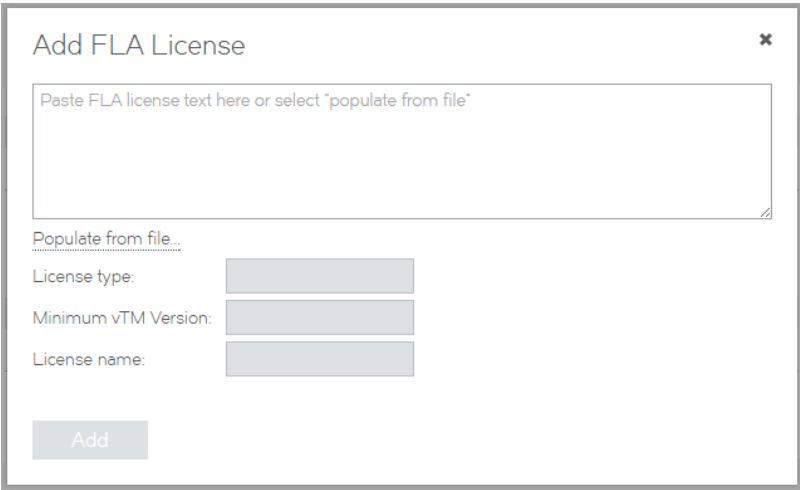
5. Either:

- – Paste the text of the Legacy FLA License into the text box, OR
  - – Click **Populate from File**, select the file and then click **Upload**. This will populate the text box.

The remainder of the fields in the dialog will then update to provide license information:

FIGURE 55 License Information

6. Click **Add**. A relicensing pop-up appears.

7. Click **Later**.

The new license is added to the **FLA Licenses** page.

FIGURE 56 Flexible Licenses Page: Legacy FLA Added

8. Repeat this procedure if you require additional licenses.

9. Both Legacy FLA Licenses and Universal FLA Licenses have a default FLA. If you have more than one FLA license for either type, and want to make it the default license for that type, click **Make Default**.

Uploading a Traffic Manager Image

Before you can deploy a Traffic Manager, you must upload the required Traffic Manager image onto the Services Director VA. During this process, you also specify the required version resource for the image.

1. Access your Services Director VA from a browser, using its Service Endpoint IP Address.

2. Log in as the administration user. The **Home** page appears.

3. Click the **Catalogs** menu, and then click **vTM Images**.

   The **vTM Images** page appears.

4. Click the plus symbol above the table of images. A pop-up window appears:

   FIGURE 57 Adding a vTM Image

5. Click **From file** and then **Choose File**. Then, select the required local Traffic Manager image file. This is in TGZ format.

6. Specify the required **Version Name**. Typically, this is the version number, for example 11.0.

7. Optionally, complete the Info field.

8. Click **Apply**.
   The image file is uploaded to the Services Director VA. Wgen complete, it is displayed in the **vTM Images** page. This may take several minutes.

Configuring Passwordless SSH

The Services Director VA uses passwordless SSH to communicate with your instance host machines. Passwordless SSH enables the Services Director VA to copy files and remotely run commands to deploy, start, stop, upgrade, and delete Traffic Manager instances.

Each Services Director VA in an HA pair has its own unique SSH authentication key pair. When you add a new instance host machine to the Services Director VA, you must set up passwordless SSH between each Services Director VA and the new instance host machine.

The following is a general procedure for configuring the establishing passwordless SSH between the Services Director VA and an instance host machine.

1. Once your VA is configured, licensed, and running, SSH into the VA as **admin**, and generate the SSH key pair that will be used for passwordless SSH access to the instance host machine:

```
> enable
# configure terminal
# ssh client generate identity user admin
# show ssh client private
```

2. Copy the listed public key and paste it into the `/root/.ssh/authorized_keys` file on the instance host machine.

3. Test the password-less connection from the Services Director VA:

```
ssh slogin root@<instance host>
```

Adding the Instance Host Machine to the Services Director VA

Perform the following procedure to add the instance host machine to the Services Director VA.

1. Log into the Services Director VA.

2. Access the **vTM Instance Hosts** page from the menus using **Services** > **vTM Instance Hosts**.

3. Click the plus sign above the table of instance hosts. The **Add vTM Instance Host** dialog appears.

4. On the **Add vTM Instance Host** dialog, enable the **Custom Instance Hosts** check box.

5. Complete the dialog. For example:

FIGURE 58 Adding a vTM Instance Host

NOTE

For instance host machines, the **Instance Host User** and **Instance Host Password** fields are not present on the dialog.

6. Click **Add** to add the instance host machine. The instance host is added to the **vTM Instance Hosts** page

7. After registering the instance host machine, the UI will show the instance host as 'Not running', and with an 'Custom' version.

FIGURE 59 Adding a vTM Instance Host

This is expected, as the Instance Host VA-specific services will not be present on instance host machine. The **Action** column will be empty.

8. To test the registered Host, make a REST request to

```
https://<Services Director>:8100/api/tmcm/2.5/host/<hostname>?status_check=true
```

The response will include a **status_check** property that will be an empty object if the Host is correctly configured for deployments.

If not, **status_check** will have a further **setup_errors** property describing the problems found with the Host.

Deploying a Traffic Manager Instance

Is not possible to use the GUI or CLI to deploy a containerised instance to a custom-built VM, as these depend on certain services being available on the Instance Host image.

However, using the REST API is possible. All start/stop/delete actions are still possible through the UI, and the Instance **Lifecycle**, **Instance Health** and **Licensing Health** columns will operate as expected. Similarly, monitoring and metering function correctly with Instances deployed to custom Hosts.

See Using an Instance Host VA with a Services Director VA on page 67 for more information on how to use the REST API to deploy vTMs.

NOTE

Non-containerised Instances that are deployed via the REST API will show as having an Instance Type of 'Externally Deployed' in the Services Director UI.

The following worked example uses the two-NIC example host configuration from above, to deploy a simple containerised Instance:

1. Create a container configuration file, in the `/root/install` directory, with filename `<container-fqdn-or-ip>.conf`.

This filename must be the value specified for the **container_name** property of the Instance resource. For Instance resources that will not be able to use the Universal FLA License, the hostname/IP part of the container configuration filename must match

the value of the **lxc.utsname** used in the file and the **management_address** specified in the REST request (see the below example).

2. Create contents for the container configuration file. For example:

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0
lxc.network.ipv4 = XX.XX.XX.XX/XX
lxc.network.ipv4.gateway = YY.YY.YY.YY
lxc.utsname = djones-04.cam.demo.com
```

3. Use an HTTP client (for example, POSTMAN) to PUT the Instance resource:

```
curl -k -u admin:password https://djones-02.cam.demo.com:8100/api/tmcm/2.5/instance/r04-conf -X PUT -
H "Content-type: application/json"
-d '{"management_address": "djones-06.cam.demo.com", "container_name": "djones-04.cam.demo.com",
"owner": "me", "bandwidth": 100,
"stm_feature_pack": "STM-400_full", "stm_version": "10.1", "host_name": "djones-06.cam.demo.com",
"cpu_usage": "0"}'
```

It may take a long time to perform the first deployment, as the vTM Image needs to be copied to the Host, un-tarred, and installed.

4. You can now use the user interface can be used to start/stop/delete the Instance.

FIGURE 60 Deployed vTM Instance

The **Instance Type** will be correctly described as Container, but the container properties will be blank or, in the case of **Data1 IP/Subnet** and **Data2 IP/Subnet**, Undefined.

Unfortunately, this also prevents changing any non-container properties through the UI - any attempted update will throw the error 'Failed to update Instance (Unknown container flavor ())'.

Instead, use the REST API or CLI can for any required property updates.

> **NOTE**
> The **Extra Options** property is for optional advanced configuration options. These are described under the **config_options** property in .

NOTE
Any updates to the container options through the CLI will fail.

Registering Externally-Deployed Traffic Managers

This section describes how to register externally-deployed Traffic Managers from a software form-factor Services Director.

> **NOTE**
> You cannot register an externally-deployed Traffic Manager that is in a private network behind a NAT device.

To register Traffic Managers from the Services Director VA, see the *Brocade Services Director Getting Started Guide*.

When the Services Director accesses an externally-deployed instance, it is only able to manages licenses and provide metering capabilities for billing; it does not support Services Director life-cycle operations on externally-deployed instances.

> **NOTE**
> Deployment options are also supported:
>
> - To deploy Traffic Managers to an instance host machine from a software form-factor Services Director, see Using an Instance Host Machine with a Software Services Director on page 47.
> - To deploy Traffic Managers to an instance host VA from a Services Director VA, see Using an Instance Host VA with a Services Director VA on page 67.
> - To deploy Traffic Managers to an instance host machine from a Services Director VA, see Using an Instance Host Machine with a Services Director VA on page 97

See also Services Director Form-Factors on page 14 and Instance Host Form-Factors on page 15.

Properties for an Externally-Deployed Instance

The table below describes the REST API properties for an externally-deployed instance.

Property	Description
owner	The **owner** resource for this instance. See owner Resource on page 163.
stm_feature_pack	The name of the **feature_pack** resource associated with the Traffic Manager instance. This represents the set of features that are available for the instance.
license_name	The name of the FLA **license** resource you want to use for this instance. For an externally-deployed instance, this property will not update the licenses on the Traffic Manager instance.
rest_enabled	Should set to true. This is required to enable monitoring and to allow licenses to be pushed to instances.
bandwidth	The maximum allowed bandwidth for the Traffic Manager instance (in units of Mbps).

Property	Description
tag	A text property which provides an alternative way of referring to an instance. Unlike the unique ID for an instance, the **tag** value can be changed or reused, subject to some restrictions. See Understanding the Tag Property on page 125.
management_address	The hostname used to address the Traffic Manager instance. The hostname must be a fully-qualified domain name, or an IP address where Universal Licensing is used.
	You can modify this property only for an externally-deployed Traffic Manager instance (or in a database-only request).
	If you modify this property, the host component of the **rest_address** , **ui_address** , and **snmp_address** properties is also updated. These values must be fully-qualified domain name, or an IP address where Universal Licensing is used.
ui_address	The address (host or IP address plus port number) of the Traffic Manager instance Administration UI. If you do not enter a value, the UI address defaults to :9090.
	If you use a hostname instead of an IP address, you must use a fully-qualified domain name, or an IP address where Universal Licensing is used. You can modify this property only for an externally-deployed Traffic Manager instance (or in a database-only request).
admin_username	The user name for the admin account for the externally-deployed instance.
	This is essential for communication between the Services Director and the externally-deployed instance.
admin_password	The password for the admin account for the externally-deployed instance.
	This is essential for communication between the Services Director and the externally-deployed instance.
rest_address	The hostname or IP address and port number of the Traffic Manager instance configuration REST API. If left blank, it defaults to :9070. The **rest_address** property must match the instance hostname.
	If you use a hostname instead of an IP address, you must use a fully-qualified domain name.
	The **rest_address** property must be unique and accurate to identify a Traffic Manager instance for licensing purposes.
	You can modify this property only for an externally-deployed Traffic Manager instance (or in a database-only request).
	This is essential for communication between the Services Director and the externally-deployed instance.
snmp_address	The hostname or IP address and port number of the Traffic Manager instance SNMP responder. This setting enables you to set the SNMP address used for metering. See Enabling Metering for an Externally-Deployed Instance on page 120.
	If you use a hostname instead of an IP address, you must use a fully-qualified domain name.
	You can modify this property only for an externally-deployed Traffic Manager instance (or in a database-only request).
config_options	A space-separated string used to define configuration options. See Specifying Configuration Options on page 61.
	A single configuration option is supported:
	• **snmp!community** - The SNMP v2 community setting for this externally-deployed instance. This must be set to the same

Property	Description
	value as the equivalent **snmp!community** property on the **instance** resource (default: "public"). **NOTE** The **config_options** property is visible in the graphical user interface of the Services Director VA. It is listed for an expanded instance on the **vTM Instances** page as **Extra Options**. Refer to the *Brocade Services Director Getting Started Guide*. **NOTE** Whenever the **config_options** property is set, all currently modified options must be specified again in the REST call. Any options that are not specified will lose their current value and be reset to their default value. **NOTE** Unlike deployed instances, externally-deployed instances do not restart when **config_options** are changed. **NOTE** When you specify this property, do not specify the **config_options_json** property.
config_options_json	A JSON data structure that is equivalent to the **config_options** property. See Specifying Configuration Options on page 61. **NOTE** When you specify this property, do not specify the **config_options** property.
access_profile	Identifies an **access_profile** resource (see access_profile Resource on page 126). The **authenticator** and **permission_group** resources referenced by this resource are then applied to the Traffic Manager instance to set its user authentication. See Applying User Authentication to a vTM on page 127.

Registering an Externally-Deployed Instance

When you register an externally-deployed instance using the REST API, you POST an instance resource.

> **NOTE**
> You cannot register an externally-deployed Traffic Manager that is in a private network behind a NAT device.

The request supports a URL parameter **?managed=false**. Include this to indicate that the new instance is an externally-deployed instance. For example:

```
https://<sdhost>:<port>/api/tmcm/2.5/instance/stm1?managed=false
```

For details of all **instance** properties, see instance Resource on page 146.

For details of the instance properties, see Properties for an Externally-Deployed Instance on page 117.

Making Database-Only Updates to an Externally-Deployed Instance

You can register externally-deployed instances with the Services Director with a database-only update or a direct update to the Traffic Manager.

If you issue a database-only REST API PUT request (either when initially creating an instance record or when modifying one), you can set a number of properties that are otherwise managed directly by the Services Director. These properties are:

- **rest_address**
- **admin_username**
- **admin_password**
- **snmp_address**
- **rest_enabled**
- **ui_address**

Setting any of these properties through a database-only REST API PUT request does not result in changes being passed to the Traffic Manager instance; only the Services Director database is affected. You can use these techniques to resolve this monitoring issue:

- Issue a GET REST API request for the Traffic Manager**instance** resource, including the URL parameter **status_check=true**. The Services Director actively checks the state of the Traffic Manager instance and updates the stored status accordingly.

- Issue a PUT REST API request to modify the Traffic Manager**instance** resource and set the **status** property to the known correct state with URL parameter **deploy=false**. The Services Director updates the status of the Traffic Manager instance in the inventory database but does not attempt to start or stop the instance itself.

The **admin_username** , **admin_password** , and **rest_address** properties are essential for communication between the Services Director and the externally-deployed instance.

Enabling Monitoring and the REST API Proxy for an Externally-Deployed Instance

To enable both monitoring and the REST API proxy for an externally-deployed Traffic Manager instance, you must configure:

- The **instance** resources in the Services Director. Provide values for the **admin_username**, **admin password** and **rest_address** properties. See Properties for an Externally-Deployed Instance on page 117.
- The externally-deployed Traffic Manager instance. Enable the REST API.

Enabling Metering for an Externally-Deployed Instance

The Services Director collects metering data from Traffic Manager instances as follows:

- Instances that are at version 9.4 or earlier (or have no REST API enabled) have their metering collected through SNMP.
- Instances that are at version 9.5 or later with the REST API enabled have their metering collected through their REST API. If REST-based metering fails (or is not possible), the Services Director falls back to collecting using SNMP. Any metering issues will be included in the warning logs, as before.

To enable metering for an externally-deployed Traffic Manager instance, you must first configure the *instance* resource in the Services Director. Provide a value for the following properties:

- **snmp_address**.

- **snmp!community**. This **config_options** setting must match the **snmp!community** setting on the instance itself.

See Properties for an Externally-Deployed Instance on page 117.

You must then configure the externally-deployed Traffic Manager instance itself:

- Enable SNMP.

- The **snmp!community** setting must match the **snmp!community** setting in the **config_options** on the **instance** resource.

   NOTE
   When you are configuring an externally-deployed instance, an **Instance Host** resource is not required.

Using the REST API with the Services Director

Introducing REST

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

A REST interface partitions the API into a series of resources, each of which you can access using one or more HTTP methods. With the Services Director, only the GET, PUT, and POST methods are used. (The **action** resource also supports the DELETE method due to the transient nature of the data contained within it. For more details, refer to action Resource on page 128. Each method operates on the Services Director as follows:

* GET – Obtain a representation of the resource without modifying the server state (except perhaps for logging purposes).
* PUT – Create a new resource or apply some change to a resource. If the resource exists, only those properties specified in the request are modified; all others remain unchanged. If a resource object does not exist, a new one is created.
* POST – Create a new resource based on the details contained in the request body. If the resource exists, it is overridden. This method applies to the **controller_license_key** and **bandwidth_pack_license_key** resources only. The **add_on_pack_license_key** resource also supports the POST method for resource creation.

    NOTE
    You cannot delete resources for auditing purposes (with the exception of the **action** resource). Instead, mark a resource as inactive by altering its **status** property. You cannot mark a resource as Inactive if it is in use, and it cannot be altered after you mark it as Inactive (the name cannot be reused).

An **Accept** header, if present, provides a list of acceptable MIME types. If you specify an **Accept** header in your request, it must allow a MIME type of **application/json** for all resource types except **licence**.

```
Accept: application/json
```

The license resource allows a MIME type of either **application/json** or **text/plain**. This can be used to extract the raw text of the FLA license file if required. See license Resource on page 157.

The **Content-type** header when using PUT and POST methods matches the content type expected by each resource, which is typically **application/json**. However, when you POST license keys to **<license type>_license_key** resources, you must set the **Content** header to plain text.

```
Content-Type: text/plain
```

Each resource is uniquely identified with an address or uniform resource identifier (URI). In other words, if you know the URI, you can access the Authentication resource (subject to the authorization and authentication process).

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

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

The full range of HTTP return codes is available in REST, although in practice you can identify and apply a useful subset consistently. For example, the response can tell you whether a request has succeeded or not without any need for parsing the body of the response. However, the Services Director always attempts to provide extra information regarding a failure into the response body.

Authentication

All requests to the Services Director REST API must be authenticated by means of *HTTP Basic Authentication*. You must create an initial Services Director user outside of the REST API, but you can create and manage other users using the REST API. You must access the Services Director REST API through HTTPS. Client certificates are not checked for validity, and HTTPS is used only for encryption and to allow the FLA license to verify the server identity.

URI Root Parts

All inventory database resources are provided through a common base URI that identifies the root of the resource model:

```
https://<host>:<port>/api/tmcm/2.5
```

In this example, <host> is the hostname of the server containing the inventory database, and <port> is the port that the REST API is published on. You can find all inventory resources at this URI. You can perform a GET request on any level of the base URI to obtain a list of the child elements it contains.

Inventory Resources

This table summarizes inventory resources. Each of the inventory resources is located under a specific URI.

Resource URI	Description
/api/tmcm/2.5/action	The list of pending, blocked, or waiting deployment actions. See action Resource on page 128.
/api/tmcm/2.5/add_on_pack_license_key	The list of installed Add-On license keys. See add_on_pack_license_key Resource on page 128.
/api/tmcm/2.5/add_on_sku	The list of supported Add-On SKUs. See add_on_sku Resource on page 129.
/api/tmcm/2.5/config/authentication/access_profile	The list of access profiles to support user authentication on Traffic Manager instances. See access_profile Resource on page 126.
/api/tmcm/2.5/config/authentication/authenticator	The list of authenticators to support user authentication on Traffic Manager instances. See authenticator Resource on page 129.
/api/tmcm/2.5/config/authentication/permission_group	The list of permission groups to support user authentication on Traffic Manager instances. See permission_group Resource on page 164.
/api/tmcm/2.5/backup	The list of cluster backups. See backup Resource on page 135.
/api/tmcm/2.5/bandwidth_pack_license_key	The list of installed Bandwidth Pack license keys. See bandwidth_pack_license_key Resource on page 136.

Resource URI	Description
/api/tmcm/2.5/cluster	The list of defined Services Director clusters. See cluster Resource on page 139.
/api/tmcm/2.5/controller_license	The list of installed Services Director licenses. See controller_license Resource on page 141.
/api/tmcm/2.5/controller_license_key	The list of installed Services Director license keys. See controller_license_key Resource on page 141.
/api/tmcm/2.5/dashboard	A summary view of certain Services Director operations. See dashboard Resource on page 143.
/api/tmcm/2.5/feature_pack	The list of feature packs that you can apply to Traffic Manager instances. See feature_pack Resource on page 144.
/api/tmcm/2.5/host	The list of Traffic Manager instance hosts on which you can deploy Traffic Manager instances. See host Resource on page 145.
/api/tmcm/2.5/instance	The list of Traffic Manager instances. See instance Resource on page 146.
/api/tmcm/2.5/license	The list of license files that you can apply to instances. See license Resource on page 157.
/api/tmcm/2.5/manager	The list of individual Services Director instances that share the same database. The list also contains mode settings that you can manipulate to achieve HA for the Services Director. See manager Resource on page 158.
/api/tmcm/2.5/monitoring	A read-only resource containing monitoring state data on Services Directors and Traffic Manager instances in your deployment. See monitoring Resource on page 160.
/api/tmcm/2.5/ping	This can be used in a GET to confirm the service is running.
/api/tmcm/2.5/registration	This is used during self-registration of Traffic Manager instances. See registration Resource on page 169.
/api/tmcm/2.5/schedule	The list of defined backup schedules. These can be assigned to clusters to create automatic cluster backups. See schedule Resource on page 171.
/api/tmcm/2.5/sku	The list of SKUs that you can use to create feature packs to apply to Traffic Manager instances. See sku Resource on page 178.
/api/tmcm/2.5/status	This enables you to check the status of other inventory resources. See Using the REST API to Check Status on page 182.
/api/tmcm/2.5/user	The set of Services Director administrative users. See user Resource on page 180.
/api/tmcm/2.5/version	The list of Traffic Manager versions you can apply to instances. See version Resource on page 181.

All resource names can be any acceptable URL part. URL encoding allows characters such as spaces. These might not be legitimate user or hostnames in the underlying system, but this is not checked or enforced by the REST API.

Resource Reference

This section contains a full description of the resource objects that you can obtain data from the Services Director REST API. Each resource contains properties and a set of rules governing how you can interact with its properties.

Understanding the Tag Property

Every resource has a unique ID. Many resources also have a **tag** property which provides an alternative way of identifying the resource.

A tag is a user-friendly name which, unlike the unique ID for a resource, can be changed or reused (subject to some restrictions).

A tag is useful in the situation where you want to identify a resource by a consistent name. For example, if an error occurs that requires a Traffic Manager instance to be deleted, the unique ID is no longer available, as the instance persists with a Deleted status. The tag, however, can be reused on a new instance, enabling consistent naming.

Restrictions to tag values are as follows:

- A tag cannot be the same as a unique ID of any resource (except itself).

   NOTE
   This restriction includes Deleted Traffic Manager
   instances.

- A tag must be unique among the tags of all resources.

   NOTE
   This restriction does not include Deleted Traffic Manager
   instances.

- The tag can contain:
  - the characters a-z, A-Z and 0-9.
  - other ASCII character between 32 (space) and 126 (tilde), excluding reserved gen-delims punctuation.
  - the tag can include spaces, but cannot be made entirely of spaces.
  - other characters can be included, but will be displayed using their ANSI escape sequences. That is, the ä character is displayed as \xe4. The same limitation applies to the \ character, which displays as \x5c.

access_profile Resource

An **access_profile** resource combines an **authenticator** resource (see authenticator Resource on page 129) with one or more **permission_group** resources (see permission_group Resource on page 164) for the purposes of user authentication.

This information is applied to a Virtual Traffic Manager (vTM) when the vTM's user authentication is set from the Services Director. See Applying User Authentication to a vTM on page 127.

There are no default access profiles.

   NOTE
   You cannot delete **authenticator** resources or **permission_group** resources that are included in an **access_profile** resource.

The **access_profile** resource is located under the **/api/tmcm/2.5/config/authentication** resource.

The **access_profile** resource will accept the following HTTP methods:

- POST, for creation.
- PUT, for update.
- GET.
- DELETE.

An **access_ profile** resource contains the following properties.

Property	Description
tag	Optional. If unset, this is set to the ID of the access profile.

Property	Description
	The name of the access profile. This must be unique amongst IDs and tags of **access_profile** resources, except when empty or set to its own ID.
authenticator_id	The ID of the authenticator for the access profile.
permission_group_ids	Optional. A list of permission group IDs. Typically, there will be one or more.

Create an **access_ profile** resource using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -D headers.txt -u
admin:adminPassword -X POST -d '{"authenticator_id":"Authenticator-X1UB-AAC1-2WQ6-
X1SN","permission_group_ids":["Permission-Group-0XZS-8EIJ-420T-T59D"]}' https://servicesdirector1.acme.com:
8100/api/tmcm/2.5/config/authentication/access_profile
```

The response body to a POST contains the properties of the created **access_ profile** resource. For example:

```
{"access_profile_id": "Access-Profile-IPLT-UFHD-327L-OVAB", "tag": "Access-Profile-IPLT-UFHD-327L-OVAB",
"permission_group_ids": ["Permission-Group-0XZS-8EIJ-420T-T59D"], "authenticator_id": "Authenticator-X1UB-
AAC1-2WQ6-X1SN"}
```

Applying User Authentication to a vTM

You can specify an **access_profile** when you first register an externally-deployed vTM, see instance Resource on page 146.

You can also define the user authentication for a registered vTM from the Services Director as follows:

1. Create the required **authenticator** resource, see authenticator Resource on page 129.

2. Create any required **permission_group** resources, see permission_group Resource on page 164.

3. Create an **access_profile** resource, combining an authenticator with the required permission groups, see access_profile Resource on page 126.

4. Reference the **access_profile** resource in the required instance resource as follows:

   /api/tmcm/2.5/instance/<instance>?access_profile=<access_profile_id>

   For example:

   /api/tmcm/2.5/instance/my_instance_07?access_profile=Access-Profile-2JSI-1XKF-CZZT-AM9P

   The authenticator and permission groups are then applied to the vTM. Existing authenticators and permission groups may be overwritten, but none will be deleted. All members of a cluster are affected.

   NOTE
   To be eligible for this process, a vTM instance must be marked as Active and have its REST API enabled.

   NOTE
   A registered instance cannot be relicensed and have an access profile applied as a single request; licensing will take priority.

action Resource

An **action** resource describes a deployment action. Whenever a REST request that affects an **instance** resource triggers a deployment action, an **action** resource is created. The resource is removed when the action is completed. An **action** resource can persist for the following reasons:

- If the Services Director experiences a failure or interruption before an action is completed, an **action** resource is retained and is retried when the Services Director recovers.

- If an action fails, the **action** resource is retained and marked as Blocked. It is not automatically retried, but it can be queued for implementation after any underlying problem has been addressed.

- The REST API does not allow direct creation of an **action** resource. However, you can delete an **action** resource by making a DELETE request. You cannot recover a deleted action.

An **action** resource contains the following properties.

Property	Description	Supported Actions
request_user	The name of the user whose request triggered the action.	Read Only
request_ip	The IP address of the request that triggered the action.	Read Only
action_type	A string representation of the action: DEPLOY, START, STOP, UPGRADE, or DELETE.	Read Only
action_args	A string representation of the arguments to the action.	Update
status	The status of the **action** resource: • Waiting – Scheduled and waiting to be implemented. • Pending – Currently being processed. • Blocked – An error occurred.	Update
created	A timestamp string representation of the date and time that the action was created.	Read Only
instance	A structure with the name and **href** of the Traffic Manager instance that the action is intended to change.	Read Only
blocked	A string representation of the date and time when the action was blocked (only applicable when the status is Blocked).	Read Only
block_reason	A description of the reason the action was blocked, intended to aid in debugging and fixing the problem (only applicable when the status is Blocked).	Read Only

You can change the status of an **action** resource from Blocked to Waiting using a PUT request (which causes the request to be reattempted). You can also change its **action_args** property. No other property changes are supported.

add_on_pack_license_key Resource

An **add_on_pack_license_key** resource describes the contents of a decoded Services Director Add-On license key.

An **add_on_pack_license_key** resource contains the following properties.

Property	Description	Supported Actions
bandwidth	The bandwidth limit for this license.	Read Only
controller_license	The optional associated Services Director**controller_license_key** resource.	Read Only
feature_sku	The feature **sku** resource this key applies to.	Read Only
license_key	The license key string.	Read Only
serial	The serial number of this license	Read Only
timestamp	A timestamp encoded in this license.	Read Only
valid	Describes whether the key was successfully validated (with a currently active controller license key): true or false.	Read Only
valid_from	The license start date (Perpetual).	Read Only
valid_until	The license end date (Perpetual).	Read Only

Create a new **add_on_pack_license_key** resource by making a POST request to the resource with the license key text in the request body.

```
POST /api/tmcm/2.5/add_on_pack_license_key HTTP/1.1
Content-Type: text/plain
LK1-ERSSCAPFIPS:1:VXTNN000C5725:20130808T0931351375969495-0000-0000-5-ACB8-AE89-67EE
```

You must include a **Content-Type** header set to text/plain.

add_on_sku Resource

An **add_on_sku** resource defines a set of additional licensable features that can be added to those of an **stm_sku** to extend Traffic Manager or Services Director functionality. You do not apply an **add_on_sku** directly to a Traffic Manager instance, but you can use it in a **feature_pack** to apply additional functionality to the base **stm_sku** of the pack.

The **add_on_sku** resources are pre-installed in the Services Director software, based on sets of features described by existing Traffic Manager template licenses and additional Services Director features. The **add_on_sku** resource is read-only; PUT and POST HTTP requests to create or modify **add_on_sku** resources are not possible.

An **add_on_sku** resource contains the following properties.

Property	Description	Supported Actions
info	An optional descriptive string.	Read Only
features	The features enabled by this **add_on_sku**. The **features** property is a string containing a space-separated list of licensable Traffic Manager or Services Director feature names that the add_on_sku enables. See sku Resource on page 178.	Read Only
status	The status of this resource: Active or Inactive.	Read Only

authenticator Resource

There are two different authenticator API resources:

- An authenticator resource for vTM user authentication, refer to

- An authenticator resource for Services Director user authentication, refer to authenticator Resource (Services Director User Authentication) on page 130.

authenticator Resource (vTM User Authentication)

An **authenticator** resource for vTM user authentication defines an external user authentication service. This information is applied to a Virtual Traffic Manager (vTM) when the vTM's user authentication is set from the Services Director.

> **NOTE**
> An **access_profile** resource (see access_profile Resource on page 126) combines an **authenticator** resource with one or more **permission_group** resources (see permission_group Resource on page 164) for the purposes of user authentication.

Three proprietary authentication services are supported:

- LDAP. See Properties for LDAP Authenticators on page 131.
- RADIUS. See Properties for RADIUS Authenticators on page 133.
- TACACS+. See Properties for TACACS+ Authenticators on page 134.

There are no default authenticators.

The **authenticator** resource for vTM user authentication is located under the **/api/tmcm/2.5/config/authentication** resource, and will accept the following HTTP methods:

- POST, for creation.
- PUT, for update. You cannot update the authenticator type.
- GET.
- DELETE. This can only succeed if the authenticator resource is not in use by an **access_profile** resource (see access_profile Resource on page 126).

To enable testing of authenticators without necessarily creating them, a query parameter ('?test=username:password') is supported. This query parameter turns a request into a synchronous test of the authenticator. When the query parameter is used in combination with a POST request (because the resource does not yet exist), the resource is not created. Instead, the output of the test is logged and a 202 response is sent back with the output of the test. In case of existing resources, the user is expected to update the properties before issuing a GET request using the query parameter. Other requests will silently ignore the query parameter.

authenticator Resource (Services Director User Authentication)

An **authenticator** resource for Services Director user authentication defines an external user authentication service.

Three proprietary authentication services are supported:

- LDAP. See Properties for LDAP Authenticators on page 131.
- RADIUS. See Properties for RADIUS Authenticators on page 133.
- TACACS+. See Properties for TACACS+ Authenticators on page 134.

There are no default authenticators.

The **authenticator** resource for Services Director user authentication is located under the **/api/sd/1.0/authentication** resource, and will accept the following HTTP methods:

- POST, for creation.
- PUT, for update. You cannot update the authenticator type.
- GET.
- DELETE.

To enable testing of authenticators without necessarily creating them, a query parameter ('?test=username:password') is supported. This query parameter turns a request into a synchronous test of the authenticator. When the query parameter is used in combination with a POST request (because the resource does not yet exist), the resource is not created. Instead, the output of the test is logged and a 202 response is sent back with the output of the test. In case of existing resources, the user is expected to update the properties before issuing a GET request using the query parameter. Other requests will silently ignore the query parameter.

Properties for LDAP Authenticators

The following table describes all possible properties for an LDAP authenticator resource. There are some differences between vTM authenticators and Services Director authenticators.

Property	Description
tag	Optional. If unset, this is set to the ID of the authenticator.
	The name of the authenticator. This must be unique amongst IDs and tags of **authenticator** resources, except when empty or set to its own ID.
type	The user authentication service for the authenticator. Set this to 'ldap'. Not allowed on update.
server	The IPv4 address or resolvable hostname/FQDN of the user authentication server.
port	The port used to connect to the user authentication server. Default is 389 for LDAP.
timeout	Optional. The timeout period (in seconds) for a connection to the user authentication server. • For vTM user authorisation, this value must be an integer between 0 and 4294967295. The default is 30. • For Services Director user authorisation, this value must be 30 or less. The default is 10.
fallback_group	Optional. The permissions group to which a valid user will belong if its group is not identified. Set this to the ID of a permission group resource. Required for LDAP unless **group_attribute** is set.
group_attribute	Optional. The LDAP attribute that gives a user's group. For example: "memberOf". If multiple values are returned by the LDAP server the first valid one will be used. Default is an empty string for LDAP. Required unless **fallback_group** is set.
group_field	Optional. The sub-field of the **group_attribute** that gives a user's group. For example: if group_attribute is "memberOf" which delivers "CN=mygroup, OU=groups, OU=users, DC=mycompany, DC=local", set group_field to "CN". The first matching field will be used. Default is an empty string for LDAP.
base_dn	The base DN (Distinguished Name) for directory searches. Cannot be empty.
bind_dn	Optional. A template to construct the bind DN from the username. This is only used (and is required) when the **dn_method** is 'construct'. The string "%u" is replaced by the username. For example: • "%u@mycompany.local", OR • "cn=%u, dn=mycompany, dn=local".

Property	Description
dn_method	Value determines relevance/requirement of **bind_dn** and **search_dn**. Can be set as follows: • 'construct' – the bind DN for a user can be constructed from a known string. See the **bind_dn** field. • 'search' – the bind DN for a user can be searched for in the directory. This is necessary if you have users under different directory paths. See the **search_dn** and **search_password** fields. • 'none'. This setting is not supported for Services Director user authentication.
filter	A filter that uniquely identifies a user located under the **base_dn**. The string "%u" will be substituted with the username. For example: • "sAMAccountName=%u" (Active Directory), OR • "uid=%u" (Unix LDAP) Cannot be empty.
group_filter	Optional. If the user record returned by the filter does not contain the required group information, you can specify an alternative group search filter here. This will typically 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. For example: "(&(memberUid=%u)(objectClass=posixGroup))"
search_dn	Optional. The DN to use when searching the directory for a user's bind DN (see also **search_password**). You can leave this blank if it is possible to perform the bind DN search using an anonymous bind. Only relevant if **dn_method** is 'search'.
search_password	Optional. The password to use when searching the directory for a user's bind DN (see also **search_dn**). You can leave this blank if it is possible to perform the bind DN search using an anonymous bind. Only relevant if **dn_method** is 'search'.
status	(Services Director user authorisation only). Indicates whether the authenticator is enabled or disabled. Only one Services Director authenticator can be enabled. If **status** is enabled, the **status** for all other Services Director authenticators are disabled automatically.

You create an LDAP **authenticator** resource for vTM user authentication using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -D headers.txt -u
admin:adminpassword -X POST -d '{"base_dn":"OU=users, DC=tekton, DC=local", "bind_dn":"%u@takton.local",
"dn_method":"construct", "fallback_group":"admin", "filter":"sAMAccountName=%u",
"group_attribute":"memberOf", "group_field":"CN", "group_filter":"(&(memberUid=%u)
(objectClass=posixGroup))", "port":389, "search_dn":"", "search_password":"", "server":"xx.xx.xx.xx",
"tag":"MyLDAPServer", "timeout":30, "type":"ldap"}' https://servicesdirector1.acme.com:8100/api/tmcm/2.5/
config/authentication/authenticator
```

The response body to this POST contains the properties of the created LDAP **authenticator** resource. For example:

```
{"search_dn": "", "group_field": "CN", "authenticator_id": "Authenticator-I8EX-9PNE-3TR5-ZMW4", "bind_dn":
"%u@takton.local", "dn_method": "construct", "group_attribute": "memberOf", "group_filter": "(&(memberUid=
%u)(objectClass=posixGroup))", "server": "xx.xx.xx.xx", "filter": "sAMAccountName=%u", "tag":
"MyLDAPServer", "timeout": 30, "base_dn": "OU=users, DC=tekton, DC=local", "fallback_group": "admin",
"search_password": "", "type": "ldap", "port": 389}
```

You create an LDAP **authenticator** resource for Services Director user authentication using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -D headers.txt -u
admin:adminpassword -X POST -d '{"base_dn":"OU=users, DC=tekton, DC=local", "bind_dn":"%u@takton.local",
"dn_method":"construct", "fallback_group":"", "filter":"sAMAccountName=%u", "group_attribute":"memberOf",
```

```
"group_field":"CN", "group_filter":"(&(memberUid=%u)(objectClass=posixGroup))", "port":389, "search_dn":"",
"search_password":"", "server":"xx.xx.xx.xx", "tag":"MyLDAPServer", "timeout":30, "type":"ldap"}' https://
servicesdirector1.acme.com:8100/api/sd/1.0/authentication/authenticator
```

The response body to this POST contains the properties of the created LDAP **authenticator** resource. For example:

```
{"status": "enabled", "group_filter": "(&(memberUid=%u)(objectClass=posixGroup))", "search_dn": "", "tag":
"MyLDAPServer", "base_dn": "OU=users, DC=tekton, DC=local", "port": 389, "group_field": "CN",
"authenticator_id": "Authenticator-NMBT-7B4Y-4BE7-HEUS", "bind_dn": "%u@takton.local", "server":
"10.62.169.170", "filter": "sAMAccountName=%u", "group_attribute": "memberOf", "timeout": 30, "dn_method":
"construct", "fallback_group": "", "search_password": "", "type": "ldap"}
```

Properties for RADIUS Authenticators

The following table describes all possible properties for a RADIUS authenticator resource. There are some differences between vTM authenticators and Services Director authenticators.

Property	Description
tag	Optional. If unset, this is set to the ID of the authenticator.
	The name of the authenticator. This must be unique amongst IDs and tags of **authenticator** resources, except when empty or set to its own ID.
type	The user authentication service for the authenticator. Set this to 'radius'. Not allowed on update.
server	The IPv4 address or resolvable hostname/FQDN of the user authentication server.
port	The port used to connect to the user authentication server. Default is 1812 for RADIUS.
timeout	Optional. The timeout period (in seconds) for a connection to the user authentication server.
	• For vTM user authorisation, this value must be an integer between 0 and 4294967295. The default is 30.
	• For Services Director user authorisation, this value must be 30 or less. The default is 10.
fallback_group	Optional. The permissions group to which a valid user will belong if its group is not identified. Set this to the ID of a **permission_group** resource. Required for RADIUS unless **group_attribute** is set.
group_attribute	Optional. The attribute that specifies an account's group. For RADIUS, must be an integer between 0 and 4294967295 inclusive. Default is 1.
	Required unless **fallback_group** is set.
secret	Optional. The secret key shared with the RADIUS server.
group_vendor	Optional. The RADIUS identifier for the vendor of the RADIUS attribute that specifies an account's group. Must be an integer between 0 and 4294967295. Default is 7146. Leave blank if using a standard attribute such as Filter-Id.
nas_identifier	Optional. A string identifying the Network Access Server (NAS) which is requesting authentication of the user. This value is sent to the RADIUS server. If left blank, the address of the interface used to connect to the server will be used.
status	(Services Director user authorisation only). Indicates whether the authenticator is enabled or disabled. Only one Services Director authenticator can be enabled. If **status** is enabled, the **status** for all other Services Director authenticators are disabled automatically.

You create a RADIUS **authenticator** resource for vTM user authentication using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -D headers.txt -u
admin:adminpassword -X POST -d '{ "fallback_group": "admin", "group_attribute": 1, "group_vendor": 1476,
"nas_identifier": "Internal RADIUS", "nas_ip_address": "127.0.0.1", "port": 1812, "secret": "*", "server":
"xx.xx.xx.xx", "tag": "RADIUS Server", "timeout": 30, "type": "radius"}' https://
servicesdirector1.acme.com:8100/api/tmcm/2.5/config/authentication/authenticator
```

The response body to this POST contains the properties of the created RADIUS **authenticator** resource. For example:

```
{"authenticator_id": "Authenticator-IYGN-4BW2-M9FF-FW3O", "nas_ip_address": "127.0.0.1", "group_vendor":
1476, "group_attribute": 1, "server": "xx.xx.xx.xx", "secret": "*", "tag": "RADIUS Server", "timeout": 30,
"fallback_group": "admin", "type": "radius", "port": 1812, "nas_identifier": "Internal RADIUS"}
```

You create a RADIUS **authenticator** resource for Services Director user authentication using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -D headers.txt -u
admin:adminpassword -X POST -d '{ "fallback_group": "", "group_attribute": 1, "group_vendor": 1476,
"nas_identifier": "Internal RADIUS", "nas_ip_address": "127.0.0.1", "port": 1812, "secret": "*", "server":
"xx.xx.xx.xx", "tag": "RADIUS Server", "timeout": 30, "type": "radius"}' https://
servicesdirector1.acme.com:8100/api/sd/1.0/authentication/authenticator
```

The response body to this POST contains the properties of the created RADIUS **authenticator** resource. For example:

```
{"status": "enabled", "group_vendor": 1476, "tag": "RADIUS Server", "port": 1812, "authenticator_id":
"Authenticator-VPCA-QDSS-HS77-38HL", "nas_ip_address": "127.0.0.1", "server": "xx.xx.xx.xx", "secret": "*",
"group_attribute": 1, "timeout": 30, "fallback_group": "", "type": "radius", "nas_identifier": "Internal
RADIUS"}
```

Properties for TACACS+ Authenticators

The following table describes all possible properties for a TACACS+ **authenticator** resource. There are some differences between vTM authenticators and Services Director authenticators.

Property	Description
tag	Optional. If unset, this is set to the ID of the authenticator. The name of the authenticator. This must be unique amongst IDs and tags of **authenticator** resources, except when empty or set to its own ID.
type	The user authentication service for the authenticator. Set this to 'radius'. Not allowed on update.
server	The IPv4 address or resolvable hostname/FQDN of the user authentication server.
port	The port used to connect to the user authentication server. Default is 49 for TACACS+.
timeout	Optional. The timeout period (in seconds) for a connection to the user authentication server. • For vTM user authorisation, this value must be an integer between 0 and 4294967295. The default is 30. • For Services Director user authorisation, this value must be 30 or less. The default is 10.
fallback_group	Optional. The permissions group to which a valid user will belong if its group is not identified. Set this to the ID of a permission group resource. Required for TACACS+ unless **group_service** is set.
secret	Optional. The secret key shared with the TACACS+ server.
auth_type	The TACACS+ authentication type, either 'pap' (default) or 'ascii'.
group_service	The TACACS+ "service" that identifies a user's group field. Required unless **fallback_group** is set.

Property	Description
status	(Services Director user authorisation only). Indicates whether the authenticator is enabled or disabled. Only one Services Director authenticator can be enabled. If **status** is enabled, the **status** for all other Services Director authenticators are disabled automatically.

You create a TACACS+ **authenticator** resource for vTM user authentication using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -D headers.txt -u
admin:adminPassword -X POST -d '{"auth_type": "pap","fallback_group": "admin","group_field":
"mygrp","group_service": "acme","port": 49,"secret": "mysecret","server": "xx.xx.xx.xx","tag":
"TACACS_plus_Server","timeout": 30,"type": "tacacs_plus"}' https:// servicesdirector1.acme.com:8100/api/
tmcm/2.5/config/authentication/authenticator/
```

The response body to this POST contains the properties of the created TACACS+ **authenticator** resource. For example:

```
{"auth_type": "pap", "group_field": "mygrp", "authenticator_id": "Authenticator-X1UB-AAC1-2WQ6-X1SN",
"server": "10.62.164.80", "group_service": "acme", "secret": "mysecret", "tag": "TACACS_plus_Server",
"timeout": 30, "fallback_group": "admin", "type": "tacacs_plus", "port": 49}
```

You create a TACACS+ **authenticator** resource for Services Director user authentication using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -D headers.txt -u
admin:adminPassword -X POST -d '{"auth_type": "pap","fallback_group":"","group_field": "permission-
group","group_service": "zeus","port": 4900,"secret": "mysecret","server": "auth.acme.com","tag":
"acmeservice","timeout": 10,"type": "tacacs_plus"}' https://servicesdirector1.acme.com:8100/api/sd/1.0/
authentication/authenticator/
```

The response body to this POST contains the properties of the created TACACS+ **authenticator** resource. For example:

```
{ "auth_type":"pap", "authenticator_id":"Authenticator-FP2J-MFG0-J8VK-I8QK",
"fallback_group":"","group_field":"permission-group", "group_service":"zeus", "port":
4900,"secret":"mysecret","server":"auth.acme.com","status":"enabled","tag":"acmeservice","timeout":
10,"type":"tacacs_plus"}
```

backup Resource

A **backup** resource is a backup of a cluster configuration. A **backup** resource is created automatically for the Services Director for each **cluster** resource that has an assigned backup **schedule** resource. The schedule for each cluster defines the frequency of backups. You can also create a immediate cluster backup manually. See the *Brocade Services Director Getting Started Guide* for full details of cluster backup and restore.

The **backup** resource is located under the **/api/tmcm/2.5/config** resource.

The **backup** resource will accept the following HTTP methods:

- POST, for creation.
- PUT, for update.
- GET.
- DELETE.

The following table lists properties for **backup**.

Property	Description
backup_id	The UUID of the backup.
cluster_id	The UUID of the cluster that requested the backup.
date	A timestamp for the date/time of the backup.

Property	Description
description	A description of the backup. By default, this is the cluster name plus the **sequence_number**.
retain	A Boolean flag, indicating if the backup is *Retained*. This should be set to true if the backup is to be kept indefinitely, and considered as a candidate for deletion as part of the automated backup rotation.
sequence_num	The sequence number of this backup, incremented for every backup taken
size	The size of the backup.
tag	The user-facing (and reusable) cluster backup resource name. This field can be set by the user at any time.

The response body to a POST contains a JSON structure representing the properties of the created **backup** resource. For example:

```
{
    "backup_id": "Backup-9URJ-UQJM-RP10-8I42",
    "cluster_id": "Cluster-21TN-WWC0-EH2J-Z5RY",
    "date": "2016-07-06 12:15",
    "description": "Radegund Cluster#2",
    "retain": false,
    "sequence_num": 2,
    "size": 133120,
    "tag": "Backup-9URJ-UQJM-RP10-8I42"
}
```

See the cluster Resource on page 139 for details of a cluster that can be backed up.

See the schedule Resource on page 171 for details of backup schedules that control the creation of backups for a cluster.

bandwidth_pack_license_key Resource

A **bandwidth_pack_license_key** resource describes the contents of a decoded Services Director Bandwidth Pack license key.

A **bandwidth_pack_license_key** resource contains the following properties.

Property	Description	Supported Actions
valid	Determines whether the key was successfully validated (with the currently active Services Director license): true or false.	Read Only
status	Determines whether the key is currently in use by the Services Director: Active or Inactive. You can set this property to Active only.	Update
valid_from	The license start date (Perpetual).	Read Only
valid_until	The license end date (Perpetual).	Read Only
stm_sku	The **sku** resource this license applies to.	Read Only
bandwidth	The bandwidth limit for this license.	Read Only
serial	The serial number of the Bandwidth Pack license.	Read Only
timestamp	The timestamp of the Bandwidth Pack license.	Read Only
controller_license_serial	The serial number of the associated Services Director license.	Read Only
controller_licenses	An optional list of associated Services Director **license** resources.	Read Only
license_key	The license key string.	Read Only

Property	Description	Supported Actions
expiry_warning_days	The number of days warning that is given for an impending license expiry.	Read Only

Create a new **bandwidth_pack_license_key** resource by making a POST request to the resource with the license key text in the request body.

```
POST /api/tmcm/2.5/bandwidth_pack_license_key HTTP/1.1
Content-Type: text/plain
LK1-ERSSCTPSTM_B_200:1:VXTNN000C5725:20130808T0931351375969495-0000-0000-5-ACB8-AE89-67EE
```

> **NOTE**
> You must include a **Content-Type** header set to text/plain.

Unlike other REST API resources, the Services Director determines the name of the created resource and the content of each property based on the license key you use. If successful, the Services Director returns a standard POST response of **HTTP/1.1 201 Created** and the resource name in the **Content-Location** header.

The response body contains a JSON structure representing the properties of the created resource:

```
{
    "valid": "true",
    "status": "Active",
    "valid_from": "Perpetual",
    "valid_until": "Perpetual",
    "stm_sku": "STM-B-200",
    "bandwidth": "1",
    "serial": "VXTNN000C5725",
    "timestamp": "2013-08-08T09:31:35.1375969495",
    "controller_license_serial": "00003",
    "controller_license": "ERSSC00003-XXXX-YYYY",
    "license_key": "LK1-ERSSCTPSTM_B_200:1:VXTNN000C5725:20130808T0931351375969495-0000-0000-5-ACB8-
AE89-67EE"
}
```

> **NOTE**
> The request body in a POST request to the **bandwidth_pack_license_key** resource must contain exactly one valid Bandwidth Pack license key.

The **bandwidth** property of the **controller_license_key** resource on which it depends is updated as appropriate.

After a **bandwidth_pack_license_key** resource has been created, the Services Director typically activates it automatically. The exception to this is when activation would cause the current deployment to have insufficient licensed bandwidth; this is the case when an existing Bandwidth Pack license provides bandwidth, in use by the Services Director, that would not be provided by the new Bandwidth Pack license.

Activation also fails in the following circumstances:

- A license has a **valid_from** date in the future.
- A license has a **valid_until** date in the past.
- Activating a license would result in the Services Director deployment having insufficient licensed bandwidth for its current Traffic Manager instance configuration. (This is the case when multiple Bandwidth Packs have been issued with the same license key; activating one pack requires deactivation of whichever pack is currently active.)
- The license is invalid.

You can activate a Bandwidth Pack license manually by setting the **status** property to Active using a PUT request:

```
PUT /api/tmcm/2.5/bandwidth_pack_license_key/VXTNN000C572-1234567890 HTTP/1.1
{"status": "Active"}
```

After the license has been activated, you cannot set this property back to Inactive.

> **NOTE**
> You cannot activate invalid license keys (including those keys that do not validate against any installed Services Director license). These keys have their **valid** property set to false.

The Services Director populates the **controller_license** property with any matching **controller_license** resource this Bandwidth Pack license is validated against. The bandwidth licensed by the Bandwidth Pack is listed in the **controller_license_key** resource when queried. If no matching Services Director license is found, **valid** is set to false and **controller_license** is left blank.

If you attempt to reinstall an existing Bandwidth Pack license, the Services Director license state is unchanged. If you attempt to install a malformed or invalid key, the Services Director responds with **HTTP/1.1 400 Bad Request** and an appropriate error message in the response body.

To view existing licenses, perform a GET request for the **bandwidth_pack_license_key** resource:

```
https://<sdhost>:<port>/api/tmcm/2.5/bandwidth_pack_license_key
```

The response to this request contains a JSON structure representing the list of installed licenses:

```
{
    "children" : [
        {
            "href" : "/api/tmcm/2.5/bandwidth_pack_license_key/VXTNN000C572-1234567890",
            "name" : "VXTNN000C572-1234567890"
        },
        {
            "href" : "/api/tmcm/2.5/bandwidth_pack_license_key/VXTNN000C573-1234567890",
            "name" : " VXTNN000C573-1234567890"
        },
        {
            "href" : "/api/tmcm/2.5/bandwidth_pack_license_key/VXTNN000C574-1234567890",
            "name" : " VXTNN000C574-1234567890"
        }
    ]
}
```

To view the details for an individual Bandwidth Pack license key, perform a GET request for the specific **license** resource:

```
https://<sdhost>:<port>/api/tmcm/2.5/bandwidth_pack_license_key/ VXTNN000C572-1234567890
```

he response to this request contains a JSON structure representing the **license** resource. The properties displayed are the decoded contents of the license key:

```
{
    "valid": "true",
    "status": "Active",
    "valid_from": "Perpetual",
    "valid_until": "Perpetual",
    "stm_sku": "STM-B-200",
    "bandwidth": "1",
    "serial": "VXTNN000C5720",
    "timestamp": "2013-08-08T09:31:35.1375969495",
    "controller_license_serial": "00003",
    "controller_license": "ERSSC00003-XXXX-YYYY",
    "license_key": "LK1-ERSSCTPSTM_B_200:1:VXTNN000C5720:20130808T0931351375969495-0000-0000-5-ACB8-
AE89-67EE"
}
```

To delete a Bandwidth Pack license, use the DELETE request method with the desired **bandwidth_pack_license_key** resource in the URI:

```
DELETE /api/tmcm/2.5/bandwidth_pack_license_key/VXTNN000C572-1234567890 HTTP/1.1
```

If you attempt to delete a Bandwidth Pack license key that is used to provide bandwidth for the Services Director's current configuration of Traffic Manager instances, the request is rejected with an **HTTP/1.1 400 Bad Request** status code.

cluster Resource

A **cluster** resource describes a Traffic Manager cluster. You put the name of the **cluster** resource into the **cluster_id** property of each Traffic Manager**instance** resource you want to add to that cluster.

Creating a **cluster** resource does not automatically create the actual Traffic Manager cluster. You must first create the Traffic Manager**instance** resources (which in turn triggers deployment of your Traffic Manager instances) with the **cluster_id** property set to the name of the **cluster** resource.

The Services Director automatically clusters together all Traffic Manager**instance** resources using the same **cluster_id** property during deployment.

> **NOTE**
> Instances in a cluster must use the same feature pack. Not doing so will lead to the instances running with disparate features across the cluster which could mean that some features will not work as expected.

> **NOTE**
> When adding a Traffic Manager instance to an existing cluster, you must have at least one Traffic Manager instance already running (with a **status** property of Active) within the cluster; otherwise, resource creation fails.

When creating a **cluster** resource, you can specify the following properties.

Property	Description	Supported Actions
cluster_type	This field is populated by Services Director when the cluster resource is created. There are two types: • Discovered –for cluster records which were created by Services Director for the purposes of cluster awareness, • User Created - for cluster records created via the Services Director REST API.	Read-Only
in_use	This Boolean field is populated by the Services Director during response generation. It is true for cluster records which have dependent vTM Instance records (excluding Instance records in status Deleted or Failed_to_deploy).	Read-Only
members	A list of all vTMs in the cluster.	Create/Update
next_backup_time	The time at which the next backup is due, according to the backup_schedule.	Read-Only
number_backups	The number of non-retained backups that should be stored for this cluster. When a new backup is being made, if the number of non-retained backups for the cluster exceeds <number_backups>, one will be deleted.	Create/Update
owner	The owner resource for the instance. See owner Resource on page 163.	Create/Update

Property	Description	Supported Actions
schedule_id	Optional. The automated backup schedule associated with this cluster. See schedule Resource on page 171.	Create/Update
status	The status of this resource: Active or Inactive.	Update
tag	The user-facing (and reusable) cluster resource name. This field can be set by the user at any time.	Create/Update
task	A href and task ID for the most recent user triggered backup/restore/upload action.	Read-Only
user_data	The user data for the cluster. This is replaces the user data from the cloud registration resource when you are creating additional cluster members on AWS. See the *Brocade Services Director Getting Started Guide* for full details of this process.	Read-Only

There are no required properties for this resource, but the request body must be a valid JSON object. The cluster name is the resource name.

The response to a GET request for a **cluster** resource contains an additional read-only list **members** property. This list contains the names of all Traffic Manager instances in the cluster.

```
{
    "cluster_type": "Discovered",
    "in_use": true,
    "members": [
        "Instance-MXP8-575A-VL1A-VU5P",
        "Instance-OKKT-9MU0-DTZH-70YF"
    ],
    "number_backups": 5,
    "owner": "Owner-X287-ZB46-SCYV-HEZT",
    "share_tips": true,
    "status": "Active",
    "tag": "Cluster1",
    "task": {
        "href": "/api/tmcm/2.5/config/backup/task/BackupRestoreTask-SAFA-KCZM-9YEW-DO3G",
        "task_id": "BackupRestoreTask-SAFA-KCZM-9YEW-DO3G"
    },
    "user_data": "Y3x1d3Rlal9ob4FjI...<truncated>...ZpY2AzZGayz9iWWVzIg=="
}
}
```

You can set the **owner** property only when first creating a **cluster** resource. You cannot update this property in an existing **cluster** resource.

You can mark a resource as inactive by changing the **status** property to Inactive.

You can delete an empty resource by sending a DELETE http request with an empty request body to **api/tmcm/2.5/cluster/<cluster id>**.

See owner Resource on page 163 for details of owner resources.

See schedule Resource on page 171 for details of backup schedules that are available to a cluster.

See backup Resource on page 135 for details of backups taken for a cluster.

controller_license Resource

A **controller_license** resource describes the license being used by the Services Director to which you send the REST request. The resource only supports the GET method and is typically used only for identifying the license currently in use by a particular Services Director.

The **controller_license_key** resource is used to install Services Director licenses, but Bandwidth Pack license must be installed using the **bandwidth_pack_license_key** resource.

A **controller_license** resource contains the following properties.

Property	Description	Supported Actions
license_key	The full license key for this Services Director license.	Read Only
license_key_valid_from	The license start date.	Read Only
license_key_valid_until	The license end date.	Read Only

controller_license_key Resource

A **controller_license_key** resource describes the contents of a decoded Services Director license key.

A **controller_license_key** resource contains the following properties.

Property	Description	Supported Actions
add_on_packs	The list of Add-On license keys associated with this Services Director license key. This property is applicable to Enterprise license keys only; for CSP license keys, this property is omitted.	Read Only
bandwidth	The dictionary of licensed SKUs and the bandwidth allowance supplied to each one by this license and associated Add-On and Bandwidth Packs. This property is applicable to Enterprise license keys only; for CSP license keys, this property is omitted.	Read Only
bandwidth_packs	The list of Bandwidth Packs associated with this Services Director license key. This property is applicable to Enterprise license keys only; for CSP license keys, this property is omitted.	Read Only
cluster_bandwidth	The dictionary of licensed SKUs and the total bandwidth allowance supplied for each one by all valid license keys and associated Add-On and Bandwidth Packs. This property is applicable to Enterprise license keys only; for CSP license keys, this property is omitted.	Read Only
license_key	The license key string.	Read Only
license_type	The type of this license: Enterprise or Cloud Service Partner.	Read Only
serial	The serial number of this license	Read Only
status	Determines whether the key is currently in use by the Services Director: Active or Inactive.	Read Only
valid	Indicates whether this license is a valid license. That is, whether the license is in the correct format and the checksum is correct.	Read Only

Property	Description	Supported Actions
valid_from	The license start date (Perpetual).	Read Only
valid_until	The license end date (Perpetual).	Read Only
bandwidth	The list of licensed SKUs and the bandwidth limit applicable to each one under this license. For Enterprise Licensing keys, it also lists dependent Bandwidth Packs.	Read Only
cpsh	Data model changes for a future release.	
cspm	Data model changes for a future release.	

Create a new **controller_license_key** resource by performing a POST request to the resource with the license key text (such as LK1-RSSC123456-3E30-3E8A-5-0123-4567-89AB) in the request body.

```
POST /api/tmcm/2.5/controller_license_key HTTP/1.1
Content-Type: text/plain
LK1-RSSC123456-3E30-3E8A-5-0123-4567-89AB
```

> **NOTE**
> You must include a **Content-Type** header set to text/plain.

Unlike other REST API resources, the Services Director determines the name of the created resource and the content of each property based on the license key you use. If successful, the Services Director returns a standard POST response of **HTTP/1.1 201 Created** and the resource name in the **Content-Location** header.

The response body contains a JSON structure representing the properties of the created resource.

For example (Enterprise license):

```
{
        "add_on_packs": [],
        "bandwidth": [],
        "bandwidth_packs": [],
        "license_key": "LK1-XX_XXX_XXXX_XXXXXX_C_01:114-41A1-38BB-5-D4D5-6A97-69BC",
        "license_type": "Enterprise",
        "serial": "123456",
        "status": "Active",
        "valid": true,
        "valid_from": "2013-01-01",
        "valid_until": "2013-12-31",
}
```

For example (CSP license):

```
{
    "csph": xxxxx,
    "cspm": xxxxx,
    "license_key": "LK1-XX_XXX_XXXX_XXXXXX_C_01:114-47B1-42DEF-5-D4D5-6A97-69BC",
    "license_type": "Cloud Service Partner",
    "serial": "123",
    "status": "Active",
    "valid": true,
    "valid_from": "2016-01-01",
    "valid_until": "2016-11-30"
}
```

> **NOTE**
> The request body in a POST request to the **controller_license_key** resource must contain exactly one valid Services Director license key. License keys you have installed via the REST API in a clustered Services Director deployment might need to be validated by a different cluster member and might not be immediately activated as a result.

If you attempt to reinstall an existing controller license key, the Services Director license state is unchanged. If you attempt to install a malformed or invalid license key, the Services Director responds with **HTTP/1.1 400 Bad Request** and an appropriate error message in the response body.

To view existing licenses, perform a GET request for the **controller_license_key** resource:

    https://<sdhost>:<port>/api/tmcm/2.5/controller_license_key

The response to this request contains a JSON structure representing the list of installed licenses:

```
{
    "children" : [
        {
            "href" : "/api/tmcm/2.5/controller_license_key/XYZ123",
            "name" : "XYZ123"
        },
        {
            "href" : "/api/tmcm/2.5/controller_license_key/ABC123",
            "name" : "ABC123"
        }
    ]
}
```

To view the details for an individual controller license key, perform a GET request for the specific **license** resource:

    https://<sdhost>:<port>/api/tmcm/2.5/controller_license_key/ABC123

The response to this request contains a JSON structure representing the **license** resource. The properties displayed are the decoded contents of the license key.

To delete a Services Director license, use the DELETE request method with the desired **controller_license_key** resource in the URI:

    DELETE /api/tmcm/2.5/controller_license_key/ABC123 HTTP/1.1

If you attempt to delete an enterprise license key used to provide licensed bandwidth for the current deployment, the request is rejected with a **HTTP/1.1 400 Bad Request** status code.

You cannot delete cloud service provider license keys, for metering reasons, until their **valid_until** date is at least 180 days in the past.

dashboard Resource

A **dashboard** resource provides a summary view of certain Services Director operations. This is compiled automatically by the Services Director, and is read-only.

Currently, only a summary of the metering health information for the Services Director is included in the **dashboard** resource.

The **dashboard** resource is located under the **/api/tmcm/2.5/** resource.

The **dashboard** resource will only accept GET methods.

A **dashboard** resource contains the following properties.

Property	Description	Supported Actions
metering_health	A parent property for the remaining properties.	Read-Only
alert_level	A summary alert level for the Services Director's instances. This can be 1 (all instances report 1 = "OK") or 3 (one or more instances report 3 = "Warning").	Read-Only
alert_level_short_text	A summary text for the Services Director's instances . This can be either "OK" (all instances	Read-Only

Property	Description	Supported Actions
	report this) or "Warning" (one or more instances report this).	
alert_reason	Only present if **alert_level** is 3. When present, this is always "Possible accounting discrepancy".	Read-Only

The response body to a GET contains a JSON structure representing the properties of the **dashboard** resource. For example, if one or more instances report a metering discrepancy:

```
{
    "metering_health": {
        "alert_level": 3,
        "alert_level_short_text": "Warning"
        "alert_reason": "Possible accounting discrepancy"
    }
}
```

feature_pack Resource

A **feature_pack** resource describes a set of licensable features that you can apply to a Traffic Manager instance. A feature pack is defined relative to a SKU.

A feature pack is defined by a list of features excluded from a SKU (the list is empty). Therefore, the feature pack is always the same as a SKU or a strict subset of a SKU. When you deploy or modify a Traffic Manager instance, the feature pack controls which licensable features are allowed (but does not specify bandwidth limits).

The **add_on_sku** field of the **feature_pack** resource is used to specify a list of one or more Add-On SKUs to the feature pack. Similar to base SKUs (that is, the **stm_sku field**), these are paid for via an Add-On license for Enterprise licensed customers and via metered usage for CSP licensed customers.

When creating a **feature_pack** resource, you can specify the following properties.

Property	Description	Supported Actions
add_on_skus	Optionally, a list of Add-On SKUs used by this feature pack. It might be an empty list if no Add-On SKUs are included.	Create
excluded	A space-separated list of features excluded from the parent SKU. The **excluded** property is an empty list (in which the **feature_pack** includes all features from the parent SKU). The feature names in the excluded list must be only those from the parent SKU **features** property.	Create
info	An optional descriptive string. You can change the **info** property by updating an existing **feature_pack** resource, but you cannot change the **stm_sku** and excluded properties.	Create/Update
status	The status of this resource: Active or Inactive. You can mark the resource as inactive by changing the **status** property to Inactive.	Update
stm_sku	The name of the parent SKU.	Create

The response body to a GET contains a JSON structure representing the properties of the **feature_pack** resource. For example:

```
{
    "add_on_skus": [],
    "excluded": "auto",
    "info": "",
    "status": "Inactive",
    "stm_sku": "STM-400"
}
```

host Resource

A **host** resource describes a Traffic Manager instance host you can use to deploy Traffic Manager instances. The host must have specific directories set up and an SSH user enabled for access from the Services Director servers. These requirements are of the Services Director user.

If you use a host to deploy Traffic Manager instances outside containers, you must name the resource using the FQDN of the Traffic Manager instance host to allow the licensing server to operate correctly, or an IP address where Universal Licensing is used.

> **NOTE**
> If you create **host** resources solely for use with externally-deployed instances, the Services Director does not use the **username** and **install_root** properties. In this case, you do not need to set up passwordless SSH access if the host is solely to be used by externally-deployed instances. For more information about externally-deployed instances, see Properties for an Externally-Deployed Instance on page 117.

When creating a **host** resource, specify the following properties.

Property	Description	Supported Actions
info	An optional descriptive string.	Create/Update
work_location	The absolute path of a temporary directory to which you can copy and create files.	Create
install_root	The absolute path of a directory under which Traffic Manager instances are created. Do not set the path to /var/lib/lxc/.	Create
retained_info_dir	This property is currently not in use.	Create/Update
username	The name of a user that is used by means of passwordless SSH to carry out actions on the host. For several purposes, the user should be root.	Create/Update
usage_info	This property is currently not in use.	Create/Update
size	Defines the size (in instances) of the host	Create/Update
cpu_cores	This property is currently not in use.	Create/Update
status	The status of this resource: Active or Inactive.	Update

The hostname is the name of the **host** resource, and this must be resolvable in the local network environment.

The Services Director does not perform checks on the validity of the **work_location** and **install_root** directories, and there is not a check that the directories are absolute directories. Once defined, you cannot change the **work_location** and **installation_root** directories.

You can mark the resource as inactive by changing the **status** property to Inactive.

The host REST API supports a single query parameter, **status_check=true** or **status_check=false** , on GET requests. The default is false; however **status_check=true** causes a check of the host for network connectivity, user validity, and the state of the **install_root** and **work_location** directory.

If **status_check=true** is set on a GET request for a **host** resource, an extra property is included in the response:

- **status_check** - A string that is empty (if there are no problems) or that contains a description of any problems found.

Brocade recommends that you perform this check after creating a **host** resource.

instance Resource

An **instance** resource describes a Traffic Manager instance. Creating or altering an **instance** resource causes the Traffic Manager instance to be deployed, deleted, or altered.

A REST request to create or alter an instance returns promptly before the action is carried out (to avoid timeouts). You can then use a GET request to verify the status of the **instance** resource.

When you create an instance using the REST API, the request supports a URL parameter **?managed=[true/false]**. This is used as follows:

- Include **?managed=true** to indicate that you are creating a deployed instance. For example:

    ```
    https://<sdhost>:<port>/api/tmcm/2.5/instance/stm1?managed=true
    ```

- Include **?managed=false** to indicate that you are creating an externally-deployed instance. For example:

    ```
    https://<sdhost>:<port>/api/tmcm/2.5/instance/stm1?managed=false
    ```

When creating and deploying an **instance** resource, you can specify the following properties.

Property	Description	Actions
owner	The owner resource for the instance. See owner Resource on page 163.	Create/Update
cluster_id	The optional name of a **cluster** resource to which the instance belongs. If you specify an entry for this property, it must refer to a **cluster** resource. You must also set the **config_options** (or **config_options_json**) property to include **admin_ui=yes** and **start_flipper=yes**.	Create
stm_version	The name of the Traffic Manager**version** resource for the instance. If you modify this property, the Services Director upgrades the Traffic Manager instance to the new version. You can change this property only if the instance **status** is Idle.	Create/Update
observed_version	When an instance resource is created, this is populated automatically from the version of the externally-deployed Traffic Manager. This also occurs during licensing and monitoring operations.	Read Only
host_name	The name of the Traffic Manager instance host on which you deploy the instance. This name must match the FQDN of the instance host that was created, or an IP address where Universal Licensing is used.	Create

Property	Description	Actions
container_name	The name of the LXC container for the Traffic Manager instance. If this is an empty string or set to none, the Traffic Manager is not run inside a container. If you specify a name, you must create an appropriate container configuration file of the form **<containername>.conf** in the install_root directory of the container host. For example, a **container_name** of stm1.example.com requires a container configuration file named stm1.example.com.conf. The container configuration file must set **lxc.utsname** to the container name for the licensing server to operate correctly.	Create
container_configuration	A space-separated string used to set up the default network gateway inside the LXC container. See instance Resource. Use this format: `"{\"gateway\":\"<ip_address>\"}"` For LXC deployments, this is the IP address raised on the bridge interface to which this container is connected. **NOTE** When you specify this property, do not specify the **container_configuration_json** property.	Create/Update
container_configuration_json	A JSON data structure that is equivalent to the **container_configuration** property. See Specifying Container Options on page 61. **NOTE** When you specify this property, do not specify the **container_configuration** property.	Create/Update
config_options	A space-separated string used to define configuration options. See Specifying Configuration Options on page 61. • **default** – This option has no effect and is used to avoid an empty string. If this is option is used, no other options can be specified in the **config_options**. • **admin_ui=yes/no** – Start or bypass the Administration UI for the Traffic Manager instance (default: yes). You must set this to yes if you use the **cluster_id** property. • **maxfds=<number>** – The maximum number of file descriptors (default: 4096). This setting must be consistent between all instances in a cluster. (See Notes, below).	Create/Update

Property	Description	Actions
	• **webcache!size=<number>** - The size of RAM for the web cache (default: 0). This value can be specified in %, MB, GB by appending the corresponding unit symbol to the end of the value when not specifying a value in bytes. For example, 100%, 256MB, 1GB, and so on. This setting must be consistent between all instances in a cluster. (See Notes, below).	
	• **java!enabled=yes/no** - Start or bypass the Java server (default: no). This setting must be consistent between all instances in a cluster. (See Notes, below).	
	• **statd!rsync_enabled=yes/no** - Synchronize historical activity data within a cluster. If this data is unwanted, disable this setting to save CPU and bandwidth (default: yes). This setting must be consistent between all instances in a cluster. (See Notes, below).	
	• **snmp!community** - The SNMP v2 community setting for this **instance** resource. For metering of externally-deployed instances, this must be set to the same value as the equivalent **snmp!community** property on the instance itself (default: "public").	
	• **num_children=<number>** - The number of child processes (default: 1).	
	• **start_flipper=yes/no** - Start or bypass the flipper process (default: yes). You must set this to yes if you use the **cluster_id** property.	
	• **port_offset=<number>** - Offset control port values by a fixed amount. When network isolation is not provided by a container configuration, each Traffic Manager instance on a particular host should be allocated a unique port offset to avoid port clashes. The offset range is from 0 and 499.	
	• **afm_deciders=<number>** - The number of application firewall decider processes. If 0 is specified, the application firewall is not installed (default: 0). You cannot update this option after the instance has been deployed.	
	• **flipper!frontend_check_addrs=<host>** - Check instance front-end connectivity with a specific host. When the Services Director deploys an instance, it checks connectivity to the default gateway of the instance host	

Property	Description	Actions
	by sending ICMP requests to it. If the default gateway is protected by a firewall or blocks ICMP requests, instance deployment can fail. To disable deployment connectivity checks, use **flipper! frontend_check_addrs=""**. This setting must be consistent between all instances in a cluster. (See Notes, below). • **flipper!monitor_interval=<number>** – The interval, in milliseconds, between flipper monitoring actions. (default: 500 ms). For higher density Traffic Manager instance deployments, use a larger value such as 2000ms. This setting must be consistent between all instances in a cluster. (See Notes, below). **NOTE** Any change to the **config_options** settings will cause a restart of the instance. **NOTE** Some configuration options, if specified here, must be consistent between all Traffic Manager instances in a cluster: • **maxfds** • **webcache!size** • **java!enabled** • **statd!rsync_enabled** • **flipper!monitor_interval** • **flipper!frontend_check_addrs** If you set or update the value in one **instance** resource, the Services Director replicates this update automatically to the other **instance** resources. The instance will restart whenever these are changed, but other instances in the cluster must be restarted manually. **NOTE** Whenever the **config_options** property is set, all currently modified options must be specified again in the REST call. Any options that are not specified will lose their current value and be reset to their default value. **NOTE** When you specify this property, do not specify the **config_options_json** property.	

Property	Description	Actions
config_options_json	A JSON data structure that is equivalent to the **config_options** property. See instance Resource. > **NOTE** > When you specify this property, do not specify the **config_options** property.	Create/Update
cpu_usage	A string that describes which CPUs are used for this Traffic Manager instance. If used, you must either: • specify a value in a form that is used by the **taskset** command. For example, "0,3,5-7". • set this property to an empty string. This indicates that the host is not limited in its use of CPU cores (unless it is deployed within an LXC container). This is the default setting for the property if you do not specify a string. > **NOTE** > Any change to the **cpu_usage** settings will cause a restart of the instance.	Create/Update
stm_feature_pack	The name of the **feature_pack** resource associated with the Traffic Manager instance.	Create/Update
bandwidth	How much bandwidth the Traffic Manager instance is allowed (in Mbps).	Create/Update
license_name	The name of the **license** resource you want to use for this instance. When you modify this property, the Services Director updates the license on the Traffic Manager instance.	Create/Update
management_address	The hostname used to address the Traffic Manager instance. The hostname must be an FQDN, or an IP address where Universal Licensing is used. You can modify this property only for an externally-deployed Traffic Manager instance (or in a database-only request). > **NOTE** > If you update this property, the host component of the **rest_address** , **ui_address** , and **snmp_address** properties is also updated. These values must be FQDNs, or IP address where Universal Licensing is used.	Create
status	The status of this resource: • New – An instance that has been created by the REST API but has not yet been successfully deployed. • Idle – An instance that has been deployed but is not currently running.	Update

Property	Description	Actions
	Note that this is the only status from which you can delete the instance. • Active - An instance that is currently running. • Deleted - An instance that has been deleted. • Starting - An instance that is waiting to start. • Failed to start - An instance that has failed to start. • Stopping - An instance that is waiting to stop. • Failed to stop - An instance that has failed to stop. • Deleting - An instance that is waiting to be deleted. • Failed to delete - An instance whose deletion failed. • Failed to deploy - An instance that has failed to deploy. See	
creation_date	A string representation of the date and time of creation of this Traffic Manager**instance** resource.	Read Only
admin_username	The primary admin username for the Traffic Manager instance. You can modify this property only for an externally-deployed Traffic Manager instance (or in a database-only request).	Create/Update
admin_password	The password (unencrypted) of the specified admin user.	Create/Update
service_username	The primary service username for the Traffic Manager instance. This property cannot be modified and is not applicable to externally-deployed instances.	Read Only
service_password	The password (unencrypted) of the specified service user. This property is not applicable to externally-deployed instances.	Create/Update
rest_address	The address (host or IP address plus port number) of the Traffic Manager instance configuration REST API. The **rest_address** property must match the instance hostname. You can modify this property only for an externally-deployed Traffic Manager instance (or in a database-only request). **NOTE** If you use a hostname instead of an IP address, you must use an FQDN.	Create/Update
ui_address	The address (host or IP address plus port number) of the Traffic Manager instance Administration UI. This is blank if the instance does not have an active Administration UI.	Create/Update

Property	Description	Actions
	You can modify this property only for an externally-deployed Traffic Manager instance (or in a database-only request). **NOTE** If you use a hostname instead of an IP address, you must use an FQDN.	
snmp_address	The address (host or IP address plus port number) of the Traffic Manager instance SNMP responder. This enables you to set the SNMP address used for metering. You can modify this property only for an externally-deployed Traffic Manager instance (or in a database-only request). **NOTE** If you use a hostname instead of an IP address, you must use an FQDN.	Create/Update
licensed_date	A string representation of the date and time of the latest successful license validation (if any). This is blank if the Traffic Manager instance has never had a license validated.	Read Only
metrics_date	A string representation of the date and time of the latest successful metrics collection (if any). This is blank if the instance has never had metrics collected.	Read Only
metrics_throughput	The latest collected metrics figure, in bytes, for throughput. This is blank if the Traffic Manager instance has never had metrics collected.	Read Only
metrics_peak_throughput	The latest collected metrics figure, in bytes per second, for peak throughput (for example, the highest figure in the previous hour). This is blank if the Traffic Manager instance has never had metrics collected.	Read Only
metrics_peak_RPS	The latest collected metrics figure for peak RPS (for example, the highest figure in the previous hour). This is blank if the Traffic Manager instance has never had metrics collected.	Read Only
metrics_peak_SSL_TPS	The latest collected metrics figure for peak SSL TPS (for example, the highest figure in the previous hour). This is blank if the Traffic Manager instance has never had metrics collected.	Read Only
pending_action	A structure that contains the name and **href** for the pending or blocked action. This is applicable only if a Traffic Manager instance is associated with a failed or blocked action (caused by deployment, start, stop, and so on).	Read Only
access_profile	Identifies an **access_profile** resource (see access_profile Resource on page 126). The **authenticator** and **permission_group** resources referenced by this resource are then applied to the Traffic Manager instance to set its user	Create/Update

Property	Description	Actions
	authentication. See Applying User Authentication to a vTM on page 127.	
aws_instance_id	The instance ID for an AWS cloud-based instance.	Read Only
aws_vpc_id	The ID for the chosen Virtual Private Cloud for an AWS cloud-based instance.	Read Only

The response from a REST API GET request will include:

- both the **container_config** and **container_config_json** properties. Their values are equivalent.

- both the **config_options** and **config_options_json** properties. Their values are equivalent.

The instance REST API supports a single query parameter, **status_check=true** or **status_check=false** , on GET requests. The default is false; however, **status_check=true** causes an activity check for the instance. The Services Director carries out this check synchronously with the request handling, the duration of which can vary with load. If **status_check=true** is set on a GET request for an **instance** resource, an extra property is included in the response:

- **status_check** - A string that is empty (if there are no problems) or contains a description of any problems found.

The Services Director performs this check automatically after starting an **instance** resource. Any fault found is recorded as a blocking reason for the action.

If you make changes to **feature_pack** or **bandwidth** , the Services Director invokes changes to the licensed behavior of the deployed Traffic Manager instance.

Use the **status** property to track the state of a Traffic Manager instance.If you modify this property, the Services Director starts, stops, or deletes the Traffic Manager instance accordingly.

> NOTE
> The Services Director processes state transitions separately from the REST request, using a separate thread of execution.

You can make a PUT request to change the status of a resource. The Services Director immediately returns an intermediate status, subsequently applies the change, and finally updates the **status** property. You can poll for changes. The change has succeeded when you see the **status** property change to the expected value:

- If you deploy a Traffic Manager instance, this results in an immediate **status** of New. When the Services Director has successfully deployed the instance, **status** changes to Idle.

- You can start an instance when it has a **status** of Idle (or Failed to start or Failed to stop) by updating **status** to Active. The Services Director responds with an immediate **status** of Starting. When the instance has been successfully started, **status** is updated to Active.

- You can stop an instance when it has a **status** of Active (or Failed to start or Failed to stop) by updating **status** to Idle. The Services Director responds with an immediate **status** of Stopping. When the instance has been successfully stopped, its **status** is updated to Idle.

- You can uninstall an instance when it has a **status** of Idle by updating **status** to Deleted. The Services Director responds with an immediate **status** of Deleting. When the instance has been successfully uninstalled, its **status** is updated to Deleted. An instance in this state cannot be changed further.

When the Traffic Manager instance in is one of the failed states (Failed to deploy, Failed to start, Failed to stop, or Failed to delete), you are not able to return it to an Idle state using defined state transitions. You must therefore change the status of the Traffic Manager instance to Idle manually. To do this, issue a PUT request to the REST API with a URL parameter of **deploy=false** and a property of **status=Idle**. The **deploy** setting ensures that this is a database-only change. (See Using INSTALLROOT in This Guide on page 17 for details.)

Once this is complete, you are then able to delete the Traffic Manager instance.

- You can upgrade an instance by changing the **stm_version** property, but only when the instance has a **status** of Idle. The Services Director responds with an immediate **status** of Upgrading. When the instance has been successfully upgraded, its **status** returns to Idle. The following procedure describes how to upgrade an instance in a cluster using the **stm_version** property.

Proxying a Traffic Manager REST API

A running Traffic Manager instance maintains its own REST API for configuration purposes. You can access this directly using the **rest_address** property along with the **admin_username** and **admin_password** properties. However, for convenience, the Services Director provides proxy access to the instance REST API through the URI of the **instance** resource.

> **NOTE**
> You can access the instance REST API only if the instance itself is active. Proxy requests are otherwise rejected with an informative error message.

You can make proxy requests by appending the following to the URI of the Traffic Manager**instance** resource:

```
/tm/2.0/config/active
```

> **NOTE**
> You must not include the initial **/api** component when accessing a proxied instance REST API. This is necessary only if you access the API directly.

For example, the full URI to access the REST API of a running Traffic Manager instance called **stm1** is:

```
https://<host>:<port>/api/tmcm/2.5/instance/stm1/tm/2.0/config/active
```

The Services Director interprets such longer URIs as proxy requests for the named instance. The above example lists the top level configuration objects for the Traffic Manager instance **stm1**. The exact URL depends on the URLs allowed by the Traffic Manager configuration REST API and are not documented here.

In this situation, the Traffic Manager configuration REST API is aware that it is serving a proxied request and automatically adjusts any **href** properties so you can use them directly through the proxy without requiring editing.

The Services Director does not interpret or amend the request or response bodies for proxy requests; however, it does provide appropriate authentication information based on the **admin_username** and **admin_password** properties. If the Traffic Manager REST API returns an error, it is returned unchanged to the client. The exception to this is an authentication error. These are reinterpreted as 500-series errors (with an informative error message), because it implies the Services Director has applied the wrong credentials.

Proxy requests allow additional *content types* or *accept types* , because the Services Director is unable to determine for itself which types might be acceptable to the Traffic Manager REST API.

> **NOTE**
> The Services Director does not attempt to stream large requests or responses. Therefore, you should make such requests or responses directly to the appropriate Traffic Manager REST API.

To Upgrade Instances in a Traffic Manager Cluster

1. Start with all instances in the cluster in Active status.

2. For one instance in the cluster:

- Change the status of the instance to Idle.

- Upgrade the instance by changing the **stm_version** property.

- Change the instance status back to Active.

3. If the upgrade is successful, repeat Step 2 for every other instance in the cluster.

Upgrading a Traffic Manager cluster where all instances are Idle is not possible. At least one instance must be made Active before the other Idle instances are upgraded. In addition, one of the upgraded instances must be made Active so that the remaining instance can be made Idle and upgraded. This action is required because cluster replication must occur as part of a successful upgrade for a cluster member.

> **NOTE**
> Do not make any configuration changes to the Traffic Manager instances during this cluster upgrade procedure.

You can deploy a Traffic Manager instance in a HA scenario by creating a **cluster** resource and then specifying the name of this resource in the **cluster_id** property of each Traffic Manager instance you create. However, you must ensure the following:

- clustered instances must all share the same license and version number.
- clustered instances must have the **admin_ui** and **start_flipper** options set to Yes in **config_options**.

The Traffic Manager instance REST API supports a query parameter on PUT requests, **deploy=true** or **deploy=false**. The default is true; however, **deploy=false** causes the Services Director to apply changes to the inventory database, but no deployment changes are made. This supports testing and database reconciliation. If status is set with **deploy=false** , then that status is applied directly; no actions are carried out and no intermediate status is set. If a new **instance** resource is created with **deploy=false** , then the status is set to Idle on creation.

The Traffic Manager instance REST API supports a query parameter on PUT or POST requests when an instance is created, **managed=true** , or **managed=false**. The default is true. Creating an instance with **managed=false** (an externally-deployed instance) means that the instance is never managed by the Services Director in terms of deployment, starting, stopping, upgrading, reconfiguring, or deleting. The instance can make normal license requests and is metered. You can change its status, but this is stored and no attempt is made to alter the actual instance.

This option is for instances that you manually deploy, but for which you want to use the licensing and metering abilities of the Services Director. After you create an externally-deployed instance, any PUT or POST request is automatically considered to have the **deploy=false** parameter set. This means that you can change the database representation of the instance (and this can affect the features that are enabled through licensing and whether or not the instance is considered active for metering).

Because an externally-deployed instance is not deployed by the Services Director, you must set the **snmp_address** property (which cannot be set for a deployed instance) to enable metering. You can omit the **host_name** , **license_name** , **stm_version** , and **cpu_usage** properties, because these are not required for an Externally-deployed instance. If you do specify any of these properties, they must contain valid values. However, they are stored as empty strings in the inventory database.

> **NOTE**
> You cannot update these properties with empty strings for an existing instance.

Any changes to the properties of an Externally-deployed instance affect only licensing or metering. For example, changes to the version or any configuration options have no effect.

> **NOTE**
> You can modify **admin_username** , **admin_password** , **management_address** , **rest_address** , **ui_address** , and **snmp_address** for an externally-deployed Traffic Manager instance only (or in a database-only request).

Updating **management_address** for an externally-deployed Traffic Manager instance automatically updates the host or IP component of **rest_address** , **ui_address** , and **snmp_addre** ss. These values must be FQDNs, or IP addresses where Universal Licensing is used.

If a deployment, start, stop, or delete action (generated by a PUT request to create a new record or change the **status** property) fails, it can be traced through the **pending_action** property. You can analyze problems based on the **action** resource (see Proxying a Traffic Manager REST API on page 154).

After you fix any underlying problems, you can retry the action in one of two ways:

- You can change the status of the original action to Waiting. This causes the action to be re-queued and retried.
- You can change the status of the Traffic Manager instance to a new value; the system deletes the old action and queues an entirely new action based on the status you set.

Understanding the State Transition Model for Instances

The following diagram summarizes the stable states for an **Instance** resource, and the transient states that connect them.

FIGURE 61 State Transition Model for Instances
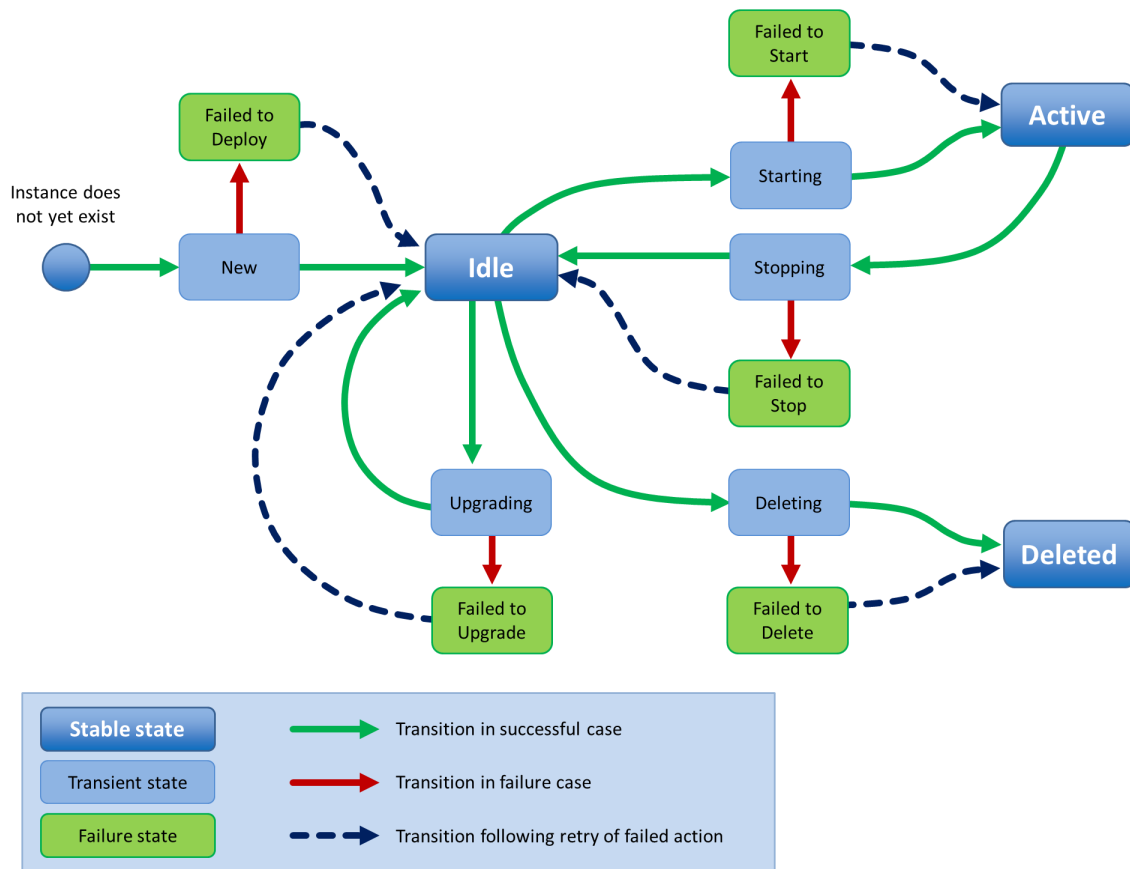
license Resource

A **license** resource describes an FLA license file that you can use to deploy a Traffic Manager instance. Creating, altering, or deleting the **license** resource does not affect the existence of the actual license file. You are responsible for ensuring that the license file is available to all Services Director servers.

> **NOTE**
> If you create **license** resources solely for use with externally-deployed instances, the Services Director does not require a corresponding license file to exist. For more information about externally-deployed instances, see Properties for an Externally-Deployed Instance on page 117.

Unlike all other resource types, the **Accept** header for a **license** resource allows a MIME type of either **application/json** or **text/plain**. Where **text/plain** is allowed but **application/json** is not, the returned data from a GET or PUT is the raw text of the FLA license file. Where no corresponding FLA license file exists, an error is returned instead.

When creating a **license** resource, you can specify the following properties.

Property	Description	Supported Actions
info	An optional descriptive string.	Create/Update
status	The status of this resource: Active or Inactive.	Update

There are no required properties in this resource, but the request body must be a valid JSON object. The license filename is the resource name.

The response body to a GET contains a JSON structure representing the properties of the **license** resource. For example:

```
{
    "default": true,
    "generic_errors": null,
    "health_check_results": [],
    "health_check_status": "Not yet run",
    "info": "Universal license",
    "last_health_check_time": null,
    "status": "Active",
    "type": "universal"
}
```

You can change the **info** property by updating an existing **license** resource.

You can mark the resource as inactive by changing the **status** property to Inactive.

Reapplying a FLA License

If you want to re-license a Traffic Manager using its assigned FLA license, use the REST API. For example:

```
curl -v -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -u
user:passwd https://x.x.x.x:8100/api/tmcm/2.5/instance/<instance_name>?relicense=true -d '{ }'
```

manager Resource

A **manager** resource represents an instance of the Services Director and contains properties that apply specifically to that instance. You can use the resource properties, known as *mode settings* , to provide HA to a Services Director deployment. The name of a **manager** resource is its hostname.

You do not need to create a **manager** resource to use an installation of the Services Director. If a **manager** resource does not already exist, it is created with reasonable default mode settings when the Services Director starts.

A **manager** resource contains the following properties.

Property	Description	Supported Actions
management	Determines whether or not the named Services Director is accessible through the REST API. This is set to enabled or disabled. To recover from the scenario where every Services Director in your deployment has its **management** property set to disabled, or has failed, Brocade provides a script, **INSTALLROOT/bin/toggle_management** , that is used to read and update the **management** property for the Services Director on which it is run.	Update
metering	Determines whether the named Services Director meters all Traffic Manager instances or none. This is set to none or all.	Update
licensing	Determines whether and how the named Services Director responds to license requests from Traffic Manager instances. You can set this to enabled, disabled, or enabledwithalerts. In enabledwithalerts mode, Services Director alert messages are sent to all configured alert email addresses when the rate at which license requests are being received exceeds a specified threshold setting.	Update
monitoring	Determines the monitoring mode for this Services Director. This mode is one of the following: • all – This Services Director monitors all other Services Directors and all Traffic Manager instances, regardless of the mode setting on any other Services Director in the cluster. • shared – This Services Director monitors all other Services Directors, but together with other Services Directors in the same mode, each one monitors only a proportion of all active Traffic Manager instances. • none – This Services Director does not perform any monitoring.	Update
external_ip	If the Services Director Service Endpoint Address is in a private network behind a NAT device, this string property is set to the external IP address for the Services Director Service Endpoint Address. Otherwise, this property is null.	Update

Deleting Manager Resources

You can delete a **manager** resources in the following ways:

- You can issue a DELETE request via the REST API to a **manager** resource that monitoring has marked as failed. This enables you to de-cluster unhealthy Services Directors. To do this, issue an HTTP DELETE request to:

  ```
  https://<SD-host:port>/api/tmcm/2.5/manager/<manager-ref>
  ```

- You can issue a forced DELETE request via the REST API to a **manager** resource that monitoring has *not* marked as failed. This enables you to de-cluster any Services Directors. To do this, issue an HTTP DELETE request with a **force** query parameter set:

  ```
  https://<SD-host:port>/api/tmcm/2.5/manager/<manager-ref>?force=true
  ```

monitoring Resource

The **monitoring** resource contains stored monitoring state data. This is a read-only resource and accepts only the GET request method.

The **monitoring** resource contains the following properties.

Property	Description	Supported Actions
manager	Contains monitoring state data for all Services Director nodes. A request for this URI returns an array of JSON objects containing properties (see below) that describe the monitoring state for all Services Director nodes in your cluster. Append the name of a specific Services Director to the URI (for example, **/monitoring/manager/sd1**) to retrieve its properties alone.	Read Only
instance	Contains monitoring state data for all Traffic Manager instances. A request for this URI returns an array of JSON objects containing properties (see below) that describe the monitoring state for all Traffic Manager instances in your cluster. Append the name of a specific instance to the URI (for example, **/monitoring/instance/tm1**) to retrieve its properties alone.	Read Only
failures	Contains monitoring state data for all failed Services Director nodes and Traffic Manager instances. This property returns a JSON structure with two elements, **managers** and **instances**. Each of these is an array of objects, one for each currently failed Services Director or Traffic Manager instance.	Read Only

Each object returned from a request contains properties that describe the monitoring state for Services Directors or Traffic Manager instances in your cluster. These properties are listed below.

Property	Description	Supported Actions
name	The name of the Services Director or Traffic Manager instance.	Read Only
monitor_date	The time stamp of the latest monitoring action.	Read Only
monitor_health	A string representation of the health of this Services Director or Traffic Manager instance.	Read Only
gone_down_date	The time stamp of when the Services Director or Traffic Manager instance was first detected to have failed.	Read Only

Property	Description	Supported Actions
	This property is present only for items in the **failures** array.	
notified_down_date	The time stamp of when the Services Director or Traffic Manager instance was considered to have failed. This property is present only for items in the **failures** array.	Read Only
cpu_idle_percent	The time, in percent (%), that CPUs are idle on the Traffic Manager instance's host. This property is present only for items in the **instance** array. It is available only for Traffic Manager instances with a status of Active and running a software version that supports performance monitoring.	Read Only
mem_free_percent	The percentage (%) of free memory available on the Traffic Manager instance's host. This property is present only for items in the **instance** array. It is available only for Traffic Manager instances with a status of Active and running a software version that supports performance monitoring.	Read Only
current_conn	The number of current connections for the Traffic Manager instance. This property is present only for items in the **instance** array. It is available only for Traffic Manager instances with a status of Active and running a software version that supports performance monitoring.	Read Only
total_bytes_in	The total number of connections the Traffic Manager instance is currently handling. This property is present only for items in the **instance** array. It is available only for Traffic Manager instances with a status of Active and running a software version that supports performance monitoring.	Read Only
total_bytes_out	The cumulative total of bytes received by the Traffic Manager instance from its clients. This property is present only for items in the **instance** array. It is available only for Traffic Manager instances with a status of Active and running a software version that supports performance monitoring.	Read Only
throughput_in	The average throughput, in bytes per second, received by the Traffic Manager instance from its clients, calculated between the previous two monitoring events. This property is present only for items in the **instance** array. It is available only for Traffic Manager instances with a status of Active and running a software version that supports performance monitoring.	Read Only
throughput_out	The average throughput, in bytes per second, sent by the Traffic Manager instance to its	Read Only

Property	Description	Supported Actions
	clients, calculated between the previous two monitoring events. This property is present only for items in the **instance** array. It is available only for Traffic Manager instances with a status of Active and running a software version that supports performance monitoring.	
licensing_activity	A summary of licensing activity for the Traffic Manager instance. This includes: • **alert_level** - The reported alert level. This can be 0, 1, 2, 3 or 4. Each setting corresponds to an **alert_level_short_text**. When **alert_level** is either 3 or 4, an **alert_reason** is also included. • **alert_level_short_text** - A text summary. This can be either "N/A", "Licensed", "Pending", "Grace period" or "Expired". Each setting corresponds to an **alert_level**. • **alert_reason** - This is only present when **alert_level** is either 3 or 4.	Read Only
metering_health	A summary of metering health for the Traffic Manager instance, which reflects possible billing discrepancies based on instance activity. This includes: • **alert_level** - The reported alert level. This can be 1 (for "OK") or 3 (for "Warning"). • **alert_level_short_text** - A text summary. This can be either "OK" or "Warning". • **alert_reason** - Only present if **alert_level** is 3. Can be either "Possible under-accounting" or "Possible uptime over-accounting". • **alert_resolution_text** - Only present if **alert_level** is 3. A potential solution for the billing discrepancy. Can be "Enable REST or SNMP connectivity for this instance" (for under-accounting) or "Mark instance as deleted if no longer in use" (for over-accounting). See the *Brocade Services Director Getting Started Guide* for a full description of metering warnings for billing discrepancies.	Read Only
rest_access	A REST access summary for the Traffic Manager instance. This includes: • **alert_level** - The reported alert level. This can be 0, 1, 2, 3 or 4. Each setting corresponds to an **alert_level_short_text**. When **alert_level** is either 3 or 4, an **alert_reason** is also included.	Read Only

Property	Description	Supported Actions
	• **alert_level_short_text** - A text summary. This can be either "N/A", "OK", "Pending", "Unhealthy" or "Failed". Each setting corresponds to an **alert_level**. • **alert_reason** - This is only present when **alert_level** is either 3 or 4.	
id_health	A summary of instance health for the Traffic Manager instance. This includes: • **alert_level** - The reported alert level. This can be 0, 1, 2, 3 or 4. Each setting corresponds to an **alert_level_short_text**. When **alert_level** is either 3 or 4, an **alert_reason** is also included. • **alert_level_short_text** - A text summary. This can be either "N/A", "OK", "Pending", "Unverified" or "Unavailable". Each setting corresponds to an **alert_level**. • **alert_reason** - This is only present when **alert_level** is either 3 or 4. See the *Brocade Services Director Getting Started Guide* for a full description of instance health.	Read Only
traffic_health	A summary of traffic health for the Traffic Manager instance. This includes: • **errors** - a list of errors for the instance. • **failed_nodes** - a list of failed instance nodes. • **license** - the license name for the instance. For example: Universal_v4. • **tip_errors** - a list of tip errors for the instance. • **error_level** - the reported error level. For example: 1 • **virtual_servers** - a list of virtual servers for the instance. See the *Brocade Services Director Getting Started Guide* for a full description of traffic health.	Read Only

owner Resource

There are several Services Director resources that require an owner. This property identifies a person or organisation that is associated with a resource, and optionally includes contact information.

A single owner entry might be used for all resources owned by a Enterprise customer. Alternatively, an owner entry might be created to identify individual customers for resources supplied by a Cloud Service Provider.

The following resources require an owner:

• An externally-deployed vTM Traffic Manager instance. See .

- A vTM Traffic Manager instance that is deployed using an instance host. See Using an Instance Host VA with a Services Director VA on page 67 and Using an Instance Host Machine with a Services Director VA on page 97.
- A vTM Cluster. See cluster Resource on page 139.

The **owner** resource has the following properties.

Property	Description	Supported Actions
clusters	A list of clusters that have this owner.	Read Only
email_address	An optional e-mail address for the owner.	Create/Update
instances	A list of instances that have this owner.	Read Only
owner_id	The ID of the owner.	Read Only
secret	An optional password for the owner.	Create/Update
tag	Optional. If unset, this is set to the ID of the owner. The name of the owner. This must be unique amongst IDs and tags of **owner** resources, except when empty or set to its own ID.	Create/Update
timezone	The timezone occupied by the owner.	Create/Update

The response body to a GET contains a JSON structure representing the properties of the **owner** resource. For example:

```
{
    "clusters": [],
    "email_address": "",
    "instances": [],
    "owner_id": "Owner-00O3-F9E3-UQW5-QONU",
    "secret": "",
    "tag": "Bernie",
    "timezone": "Europe/London"
}
```

permission_group Resource

There are two different permission_group API resources:

- A permission_group resource for vTM user authentication, refer to permission_group (vTM User Authentication) on page 164.
- A permission_group resource for Services Director user authentication, refer to permission_group (Services Director User Authentication) on page 168.

permission_group (vTM User Authentication)

A **permission_group** resource for vTM user authentication defines what a user in the group can do, by combining permission names with access levels. This information is applied to a Virtual Traffic Manager (vTM) when the vTM's user authentication is set from the Services Director.

There are four default permission groups:

- admin - this group has full access to all vTM pages.
- Demo - this group has full access, except to user management / system.
- Monitoring - this group has access only to config summary / monitoring pages.
- Guest - this group has read-only access.

You can update these permission groups, or create additional permission groups.

NOTE
An **access_profile** resource (see access_profile Resource on page 126) combines an authenticator resource (see authenticator Resource on page 129) with one or more **permission_group** resources for the purposes of user authentication.

The **permission_group** resource for vTM user authentication is located under the **/api/tmcm/2.5/config/authentication** resource, and will accept the following HTTP methods:

- POST, for creation.
- PUT, for update.
- GET.
- DELETE. This can only succeed if the **permission_group** resource is not in use by an **access_profile** resource (see access_profile Resource on page 126).

A **permission_group** resource for vTM user authentication contains the following properties.

Property	Description
tag	Optional. If unset, this is set to the ID of the permission group. The name of the permission group. This must be unique amongst IDs and tags of **permission_group** resources, except when empty or set to its own ID.
description	Optional. A description for the permission group.
timeout	Optional. The timeout period (in minutes) for a connection. Must be an integer between 0 and 4294967295. Default is 30.
permissions	A list of paired objects. Each pair is a permission and its access level for users of the permission_group. The pairs are as follows: • **access_level**: a string, set to 'none', 'ro' (read-only) or 'full'. • **name:** the name of a permission. For a full list of permissions, see Permissions for permission_group Resources on page 165.

Create a **permission_group** resource using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -D headers.txt -u
admin:adminPassword -X POST -d '{"description":"VServer admin","permissions":[{"access_level":
"full","name":"Virtual_Servers"},{"access_level":"ro","name": "Pools"},{"access_level": "ro","name":
"Traffic_IP_Groups"},{"access_level":"ro","name":"Monitors"}],"tag":"VSAdmin","timeout":30}' https://
servicesdirector1.acme.com:8100/api/tmcm/2.5/config/authentication/permission_group/
```

The response body to a POST contains the properties of the created **permission_group** resource. For example:

```
{"description":"VServer admin","permission_group_id":"Permission-Group-0XZS-8EIJ-420T-T59D","permissions":
[{"access_level":"ro","name":"Pools"},{"access_level":"full","name":"Virtual_Servers"},
{"access_level":"ro","name":"Traffic_IP_Groups"},
{"access_level":"ro","name":"Monitors"}],"tag":"VSAdmin","timeout":30}
```

Permissions for permission_group Resources

The **permission_group** resource for vTM user authentication can reference any permission that is supported by the Virtual Traffic Manager (vTM). Each such permission can be used as the name in a permissions pair. The list of supported permissions changes between vTM releases, and so each specified name is not verified by the Services Director; the vTM performs this function when the permission group is applied to the vTM. Any name that is not recognised when received by the vTM is ignored.

A current list of permissions is shown below:

```
all
all
```

```
Access_Management
Access_Management!AuthenticationMethods
Access_Management!AuthenticationMethods!Edit
Access_Management!Groups
Access_Management!Groups!Edit
Access_Management!LocalUsers
Access_Management!LocalUsers!Edit
Access_Management!LocalUsers!EditOtherUsers
Access_Management!LocalUsers!PasswordPolicy
Access_Management!Suspended_Users
AFM
AFM!Administration
Alerting
Alerting!Actions
Alerting!Actions!Edit
Alerting!Event_Types
Alerting!Event_Types!Edit
Appliance_Console
Aptimizer
Aptimizer!Aptimizer_Profiles
Aptimizer!Aptimizer_Profiles!Edit
Aptimizer!URL Sets
Aptimizer!URL Sets!Edit
Audit_Log
Audit_Log!Audit_Archive
Backup
Backup!Compare
Backup!Edit
Backup!Partial
Bandwidth
Bandwidth!Edit
Bandwidth!Edit!CopyClass
Catalog
Cloud_Credentials
Cloud_Credentials!Edit
Config_Summary
Connections
Connections!Details
Custom
DateTime
Diagnose
Diagnose!Replicate
DNS_Server
DNS_Server!Zone_Files
DNS_Server!Zones
DNS_Server!Zones!Edit
Draining
Event_Log
Event_Log!Clear
Event_Log!Event_Archive
Extra_Files
Extra_Files!Action_Programs
Extra_Files!ExternProgMonitors
Extra_Files!Miscellaneous_Files
Fault_Tolerance
Fault_Tolerance!BGP_Neighbors
Fault_Tolerance!BGP_Neighbors!Edit
GLB_Services
GLB_Services!Edit
GLB_Services!Edit!DNS_Settings
GLB_Services!Edit!DNSSEC
GLB_Services!Edit!Load_Balancing
GLB_Services!Edit!Locations
GLB_Services!Edit!Request_Logging
GLB_Services!Edit!Rules
Global_Settings
Global_Settings!Restore_Defaults
Help
Java
Java!Edit
Kerberos
Kerberos!Kerberos_Keytabs
```

```
Kerberos!Kerberos_Principals
Kerberos!Kerberos_Principals!Edit
Kerberos!krb5confs
License_Keys
License_Keys!Install_New
License_Keys!Register
License_Keys!Remove
Locations
Locations!Edit
Log_Viewer
Log_Viewer!View
MainIndex
Map
Monitoring
Monitoring!Edit
Monitors
Monitors!Edit
Monitors!Edit!CopyMonitor
Networking
Networking!NAT
Persistence
Persistence!Edit
Persistence!Edit!CopyClass
Pools
Pools!Edit
Pools!Edit!Autoscaling
Pools!Edit!Bandwidth
Pools!Edit!Connection_Management
Pools!Edit!DNSAutoscaling
Pools!Edit!Kerberos_Protocol_Transition
Pools!Edit!Load_Balancing
Pools!Edit!Monitors
Pools!Edit!Persistence
Pools!Edit!SSL
Rate
Rate!Edit
Reboot
Request_Logs
Restart
Rollback
Routing
Rules
Rules!Edit
Rules!Edit!CheckSyntax
Rules!Edit!SaveAs
Rules!GEdit
Rules!GEdit!AddAction
Rules!GEdit!AddCondition
Rules!GEdit!Convert
Security
Service_Protection
Service_Protection!Edit
Service_Protection!Edit!CopyClass
Shutdown
SLM
SLM!Edit
SLM!Edit!CopyClass
SNMP
SOAP_API
SSL
SSL!CAs
SSL!CAs!Edit
SSL!CAs!Import
SSL!Client_Certs
SSL!Client_Certs!Edit
SSL!Client_Certs!Edit!Chain
SSL!Client_Certs!Edit!CopyCert
SSL!Client_Certs!Edit!Sign
SSL!Client_Certs!Import
SSL!Client_Certs!New
SSL!DNSSEC_Keys
SSL!Server_Certs
```

```
SSL!Server_Certs!Edit
SSL!Server_Certs!Edit!Chain
SSL!Server_Certs!Edit!Copy_Certificate
SSL!Server_Certs!Edit!Sign
SSL!Server_Certs!Import
SSL!Server_Certs!New
Statd
Steelhead
Support
Support!TSR
Support_Files
Sysctl
Traffic_IP_Groups
Traffic_IP_Groups!Edit
Traffic_IP_Groups!Networking
Traffic_Managers
Traffic_Managers!AddRemove
Traffic_Managers!Upgrade
Virtual_Servers
Virtual_Servers!Edit
Virtual_Servers!Edit!Aptimizer_Settings
Virtual_Servers!Edit!Classes
Virtual_Servers!Edit!Connection_Management
Virtual_Servers!Edit!Content_Caching
Virtual_Servers!Edit!Content_Compression
Virtual_Servers!Edit!DNS_Server
Virtual_Servers!Edit!GLB_Services
Virtual_Servers!Edit!Kerberos_Protocol_Transition
Virtual_Servers!Edit!Request_Logging
Virtual_Servers!Edit!Request_Tracing
Virtual_Servers!Edit!Rules
Virtual_Servers!Edit!Rules!EnableDisable
Virtual_Servers!Edit!Rules!Move
Virtual_Servers!Edit!Rules!OnceEvery
Virtual_Servers!Edit!Rules!Remove
Virtual_Servers!Edit!SSL_Decryption
Web_Cache
Web_Cache!Clear
Wizard
Wizard!AptimizeService
Wizard!Backup
Wizard!ClusterJoin
Wizard!DisableNode
Wizard!DrainNode
Wizard!EnableRule
Wizard!FreeDiskSpace
Wizard!NewService
Wizard!ReactivateNode
Wizard!RemoveNode
Wizard!Restore
Wizard!SSLDecryptService
```

permission_group (Services Director User Authentication)

A **permission_group** resource for Services Director user authentication defines what a Services Director user in the group can do.

Typically, there is a single permission group for Services Director user authentication with access to all permissions. The name of this permission group must match the group returned by the authenticator.

The **permission_group** resource for Services Director user authentication is located under the **/api/sd/1.0/authentication/** resource, and will accept the following HTTP methods:

- POST, for creation.
- PUT, for update.
- GET.
- DELETE.

A **permission_group** resource for Services Director user authentication contains the following properties.

Property	Description
tag	Optional. If unset, this is set to the ID of the permission group.
	The name of the permission group. This must be unique amongst IDs and tags of **permission_group** resources for Services Director user authentication, except when empty or set to its own ID.
description	Optional. A description for the permission group.

Create a **permission_group** resource using the REST API. For example:

```
curl -k --basic -H "Content-Type: application/json" -H "Accept: application/json" -D headers.txt -u
admin:adminPassword -X POST -d '{"description":"administration group example","tag":"admin_grp"}' https://
servicesdirector1.acme.com:8100/api/sd/1.0/authentication/permission_group/
```

The response body to a POST contains the properties of the created **permission_group** resource. For example:

```
{"permission_group_id": "Permission-Group-9XTL-5BXH-I8CI-D2FB", "tag": "admin_grp", "description":
"administration group example"}
```

registration Resource

A registration resource is created automatically by the Services Director in response to a Traffic Manager instance sending a self-registration request. See *Brocade Services Director Getting Started Guide* for full details of self-registration.

The **registration** resource is located under the **/api/tmcm/2.5/** resource.

The **registration** resource will accept the following HTTP methods:

- POST, for creation.
- PUT, for update.
- GET.
- DELETE. This can only succeed if the **permission_group** resource is not in use by an **access_profile** resource (see access_profile Resource on page 126).

You cannot use the POST method to create a registration resource. The use of POST for an existing registration will be ignored, unless the details of the registration request have changed. If the registration token matches (implying that the vTM has received at least one response from the Services Director) then the user credentials stored by the Services Director are not updated.

The **/api/tmcm/2.5/settings** collection includes a registration end-point resource, refer to settings Resources on page 173.

The following table lists properties for **registrations**.

Property	Description
instance_uuid	The UUID of the instance that made the registration request.
registration_time	The Services Director local time and date of creation of this registration record.
pending_time	The Services Director local time and date of when this record was set to Pending. This is used to calculate when a registration should be automatically blacklisted (see blacklist_timeout, above). Blacklisting can be overridden by user setting this record to Pending once more.
status	The state of the registration request – Pending, Accepted, Blacklisted, Declined.

Property	Description
instance_version	The version number of the instance making the registration request, in the format "<major>.<minor>r<revision>".
cluster_identifier	Cluster ID supplied by the instance making the registration request. Can be empty.
management_ip	The IP address of the instance's management interface.
hostname	The IPv4 address/hostname at which the vTM can be found.
rest_address	The IPv4 address/hostname and port at which the vTM can be contacted by REST.
admin_address	The IPv4 address/hostname and port at which the vTM Admin GUI can be reached.
snmp_address	The IPv4 address/hostname and port at which the vTM can be contacted via SNMP.
snmp_community	The community string for the instance's SNMP server.
email_address	Optional. Email address of the person (typically the instance's Administrator) who is seeking acceptance of this registration.
registration_message	Message from the person (typically the instance's Administrator) who is seeking acceptance of this registration. This is seen by the Services Director Administrator.
declined_reason	Only relevant when a registration transitions to 'Declined'. Optional. Human-readable string describing the reason for declination of this registration.
feature_pack	Required when a registration transitions to 'Accepted'. Must be a valid feature pack resource for the instance.
bandwidth	Required when a registration transitions to 'Accepted'. The bandwidth for the instance.
owner	Required when a registration transitions to 'Accepted'. The owner resource for the instance. See owner Resource on page 163.
valid_owner	Indicates whether the owner is valid.
instance_name	Required when a registration transitions to 'Accepted'. The tag or ID of the instance.
access_profile	Optional. Only used when a registration transitions to 'Accepted'. The user authentication defined by the access profile is applied to the instance. Existing authenticators and permission groups may be overwritten, but none will be deleted. All members of a cluster are affected. See access_profile Resource on page 126.
aws_instance_id	The instance ID for an AWS cloud-based instance. Read Only.
aws_vpc_id	The ID for the chosen Virtual Private Cloud for an AWS cloud-based instance. Read Only.

The response body to a POST contains a JSON structure representing the properties of the created **registration** resource. For example:

```
{
    "admin_address": "10.8.2.10:9090",
    "aws_instance_id": "i-7e23cef1",
    "aws_vpc_id": "vpc-ec1da988",
    "cluster_identifier": "",
    "email_address": "",
    "hostname": "10.8.2.10",
    "instance_version": "11.1b3",
    "management_ip": "10.8.2.10",
    "owner": "Owner-X287-ZB46-SCYV-HEZT",
    "pending_time": "2016-09-28 14:57:23",
    "registration_message": "",
    "registration_time": "2016-09-28 14:57:23",
```

```
        "rest_address": "10.8.2.10:9070",
        "snmp_address": "10.8.2.10:161",
        "snmp_community": "public",
        "status": "Accepted",
        "uuid": "128c4184-b967-3401-8f35-0a0947d1967b",
        "valid_owner": true
    }
```

registration_policy Resource

This resource describes an auto-accept policy. This is used for automatic self-registration of Traffic Managers.

The **registration_policy** resource has the following properties.

Property	Description	Supported Actions
bandwidth	The bandwidth that will be applied to instances that are self-registered using this policy.	Create/Update
feature_pack	The feature pack that will be applied to instances that are self-registered using this policy.	Create/Update
instance_version_range_high	The highest version number that is accepted for instances that are self-registered using this policy.	Create/Update
instance_version_range_low	The lowest version number that is accepted for instances that are self-registered using this policy.	Create/Update
management_ip_subnet	The management subnet to which instances must belong for them to be accepted for self-registeration using this policy.	Create/Update
policy_id	The ID of the registration policy.	Read Only
tag	Optional. If unset, this is set to the policy ID. The name of the registration policy. This must be unique amongst IDs and tags of **registration_policy** resources, except when empty or set to its own ID.	Create/Update

The response body to a GET contains a JSON structure representing the properties of the **registration_policy** resource. For example:

```
    {
        "bandwidth": 1000,
        "feature_pack": "STM-400-FP",
        "instance_version_range_high": "11.0",
        "instance_version_range_low": "11.0",
        "management_ip_subnet": "10.8.0.0/16",
        "policy_id": "Policy-T1HN-946O-EO5T-25Z8",
        "tag": "AWS-Accept"
    }
```

schedule Resource

A **schedule** resource is a definition of when a cluster backup will be created. This includes general frequency (hourly, daily, weekly, monthly, and instant backups) and information to specify an exact backup time. Any cluster backup will be performed for any resource that is associated with a **schedule** resource.

See the *Brocade Services Director Getting Started Guide* for full details of cluster backup and restore.

The **schedule** resource is located under the **/api/tmcm/2.5/config/backup** resource.

The **schedule** resource will accept the following HTTP methods:

- POST, for creation.
- PUT, for update.
- GET.
- DELETE.

The following table lists properties for **schedule**.

Property	Description
backup_time	The time (hh:mm) at which the backup will be taken. This is not needed for daily and user-defined frequencies.
frequency	The frequency of the schedule. This can be: • hourly - this schedule will be performed once every hour. By default, this is on the hour. You can also choose an **offset** to schedule at 15, 30 and 45 minutes past the hour. The default is represented by 0. • daily - this schedule will be performed once per day. By default, this is at midnight (00:00). Alternatively, you choose to schedule at a specific **backup_time**. No **offset** is supported. • weekly - this schedule will be performed once per week. By default, this is on Monday at midnight. Alternatively, you can choose to schedule it on the required **offset** day (Monday to Sunday are numbered 0 to 6) at a different **backup_time**. • monthly - this schedule will be performed once per month. By default, this is on the first at midnight. Alternatively, you can choose to schedule it on the required **offset** day (typically 1-28) at a different **backup_time**. • userdefined - this schedule will be performed at a custom frequency. Instead of specifying an exact time, the first backup will be taken immediately when the schedule is applied to a **cluster** resource, and then at a defined **offset** frequency, expressed in minutes.
info	A text description for the schedule resource.
offset	A difference between the default start time and the actual start time of the backup. Settings for this depend on the schedule's **frequency**.
schedule_id	The UUID of the schedule.
tag	The user-facing (and reusable) schedule resource name. This field can be set by the user at any time.

The response body to a POST for an hourly **schedule** resource contains a JSON structure representing the properties of the resource. The **offset** represents the number of minutes past the hour (0, 15, 30, and 45) at which the backup will be taken. For example:

```
{
    "backup_time": "N/A",
    "frequency": "hourly",
    "info": "Every hour at 15 mins past",
    "offset": 15,
    "schedule_id": "BackupSchedule-QWXN-VU4Z-QHMH-7HS0",
    "tag": "Quarter past the hour"
}
```

The response body to a POST for a daily **schedule** resource contains a JSON structure representing the properties of the resource. The **offset** is a default value (100) that is unused for this schedule. For example:

```
{
    "backup_time": "02:00",
    "frequency": "daily",
    "info": "Backup daily the small hours",
    "offset": 100,
    "schedule_id": "BackupSchedule-SM57-DZJH-O7FB-M86C",
    "tag": "Daily at 2am"
}
```

The response body to a POST for a weekly **schedule** resource contains a JSON structure representing the properties of the resource. The **offset** represents a day, where Monday to Sunday are numbered 0 to 6. For example:

```
{
    "backup_time": "03:00",
    "frequency": "weekly",
    "info": "For medium-churn systems",
    "offset": 1,
    "schedule_id": "BackupSchedule-50DF-FDNA-SVV2-U222",
    "tag": "Tuesday at 3am"
}
```

The response body to a POST for a monthly **schedule** resource contains a JSON structure representing the properties of the resource. The **offset** represents a day number, typically 1 to 28. For example:

```
{
    "backup_time": "23:30",
    "frequency": "monthly",
    "info": "For very low churn setups",
    "offset": 1,
    "schedule_id": "BackupSchedule-BP3X-UPFA-CBFS-GXWZ",
    "tag": "First of the month, 11.30pm"
}
```

The response body to a POST for a user-defined **schedule** resource contains a JSON structure representing the properties of the resource. The **offset** represents the number of minutes between scheduled backups. For example:

```
{
    "backup_time": "N/A",
    "frequency": "userdefined",
    "info": "For high churn systems",
    "offset": 120,
    "schedule_id": "BackupSchedule-FJO6-T9M3-72D4-1D9E",
    "tag": "Frequent periodic"
}
```

See the for details of a cluster that can be backed up.

See for details of backups taken for a cluster.

settings Resources

There is a collection of **settings** end-point resources that control certain aspects of Services Director operation.

The following resources are collected under the **/api/tmcm/2.5/settings** resource:

- deployment
- licensing
- metering
- logging

- **controller_licensing**
- **monitoring**
- **bandwidthpack_licensing**
- **phone_home**
- **fla_check**
- **master_password**
- **registration**
- **backup_scheduler**
- **security**
- **external_ip**

These resources exist automatically whenever a Services Director is installed, and will accept the following HTTP methods:

- GET. for inspection.
- POST or PUT, for updates.

> **NOTE**
> You cannot DELETE a settings resource.

The following table lists properties for all settings resources.

Settings Resource	Property	Description	Supported Actions
deployment	max_instances	The maximum number of Traffic Manager instances that can be deployed from an instance host. The default is 0, which indicates that there is no limit. **NOTE** Brocade recommends that the default value is used.	Update
licensing	alert_threshold	The threshold number of alerts that can be raised before an alarm is raised. The default is 1. **NOTE** This property is only used when the **licensing** property of the manager resource is set to "enabledwithalerts."	Update
	alert_threshold_interval	The minimum interval (in seconds) between threshold alarms. The default is 300.	Update

Settings Resource	Property	Description	Supported Actions
		NOTE This property is only used when the **licensing** property of the manager resource is set to "enabledwithalerts."	
metering	alerts_and_notifications	A Boolean that controls whether metering alerts and notifications are handled. The default value is true, which indicates that the metering health icon will be displayed in the header for the Services Director VA graphical user interface, and that daily emails about metering warnings will be sent.	Update
	meter_interval	The period of time, in seconds, between metering actions. The range is from 1–3600. The default value is 3600 (1 hour).	Update
	snmp_enabled	A Boolean that controls whether SNMP can be used to gather certain types of information (such as metering) from the Traffic Managers in the estate of the Services Director. The default is true. **NOTE** SNMP is only used after attempts to gather information via the REST API have failed.	Update
	log_check_interval	The period of time, in seconds, between checks for log space. The range is from 1–3600. The default is 3600 (1 hour).	Update
logging	authentication_logging	This property should only be used by Brocade Support.	N/A
	monitoring_logging	This property should only be used by Brocade Support.	N/A
	metering_logging	This property should only be used by Brocade Support.	N/A
	inventory_logging	This property should only be used by Brocade Support.	N/A
	license_logging	This property should only be used by Brocade Support.	N/A
	backupschedule_logging	This property should only be used by Brocade Support.	N/A
controller_licensing	expiry_warning_days	The number of days warning that is given for an impending controller license expiry.	Update

Settings Resource	Property	Description	Supported Actions
		The default is 30.	
monitoring	controller_failure_period	The number of seconds that a Services Director must be unavailable before it is flagged as Failed. The default is 180.	Update
	instance_failure_period	The number of seconds that a Traffic Manager instance must be unavailable before it is flagged as Failed. The default is 180.	Update
	host_failure_period	The number of seconds that an instance host must be unavailable before it is flagged as Failed. The default is 180.	Update
	overdue_monitoring_warning_period	The period of time, in seconds, that must elapse before any pending monitoring actions are considered overdue. This might occur during periods of unusually heavy load. If defined, a breach of this interval causes the Services Director to issue an alert. The default is 300.	Update
	instance_monitor_interval	The number of seconds between monitoring cycles for Traffic Manager instances. The default is 60.	Update
	controller_monitor_interval	The number of seconds between monitoring cycles for Services Directors. The default is 60.	Update
	monitor_email_interval	The mimimum number of seconds between email notifications. The default is 60.	Update
	host_monitor_interval	The number of seconds between monitoring cycles for instance hosts. The default is 60.	Update
bandwidthpack_licensing	expiry_warning_days	The number of days warning that is given for an impending bandwidth pack license expiry. The default is 30.	Update
phone_home	username	The user name on the phone home server. The default is sftpuserssc.	Update
	server_address	The phone home server. The default is sftp.brocade.com.	Update

Settings Resource	Property	Description	Supported Actions
	password	The password on the phone home server.	Update
	phone_home_enabled	A Boolean that controls whether phone home is enabled. The default is false.	Update
	server_port	The phone home server port. The default is 22?	Update
fla_check	fla_check_enabled	A Boolean that controls whether FLA checking is enabled. The default is true.	Update
master_password	status	Indicates the status of master password usage. The default is Active.	Update
registration	validate_owners	A Boolean that controls whether self-registration requests must have an **owner_id/tag** and **secret** properties set correctly before registration can succeed. The default is true.	Update
	blacklist_timeout	A timeout period (in seconds) for the automatic transition of a Pending registration request to Blacklisted. The default is 86400, equivalent to two days.	Update
backup_scheduler	scheduler_interval	The number of seconds between checks for backup jobs being due. The default is 300.	Update
security	user_lockout_duration_minutes	A suspension lockout duration (in minutes). If the max_login_attempts threshold limit is reached, the suspension duration lockout is applied. The default is 1 minute, and the maximum is 1440 minutes (equal to one day).	Update
	max_login_attempts	The maximum number of failed Services Director login attempts for a user. The default is 0, which indicates that there is no maximum.	Update
	auth_success_cache_time_seconds	The lifetime, in seconds, of authentication cache entries for the success of logins. The default is 30.	Update
	auth_failure_cache_time_seconds	The lifetime, in seconds, of authentication cache entries for the failure of logins. The default is 10.	Update

Settings Resource	Property	Description	Supported Actions
external_ip	external_ip	When the Services Endpoint Address of the Services Director is in a private network behind a NAT device, this is the external IP address of the Services Endpoint Address. The default is unset.	Update

The response body to a POST contains a JSON structure representing the properties of the updated resource. For example, a **monitoring** resource:

```
{
    "controller_failure_period": 150,
    "instance_failure_period": 150,
    "host_failure_period": 150,
    "overdue_monitoring_warning_period": 200,
    "instance_monitor_interval": 45,
    "controller_monitor_interval": 45,
    "monitor_email_interval": 45,
    "host_monitor_interval": 45
}
```

sku Resource

A **sku** resource contains a SKU that defines a set of licensable features. You do not apply a SKU directly to a Traffic Manager instance, but you can use it as the basis for a feature pack.

The **sku** resources are pre-installed in the Services Director software, based on sets of features described by existing Traffic Manager template licenses. The **sku** resource is read-only; PUT and POST HTTP requests to create or modify **sku** resources are not possible.

A **sku** resource contains the following properties.

Property	Description	Supported Actions
add_on_skus	Add-on SKUs that are compatible with the SKU. For example, for the STM-400 SKU, the ADD-FIPS, ADD-WAF, ADD-WEBACCEL add-on SKUs are supported.	Read Only
csp	Indicates if this SKU is for CSP customers.	Read Only
ent	Indicates if this SKU is for Enterprise customers.	Read Only
features	The features enabled for this SKU. (see below)	Read Only
info	An optional descriptive string.	Read Only
status	The status of this resource: Active or Inactive.	Read Only
feature_tier	A data model change for a future release.	
pricing_model	A data model change for a future release.	
stm_sku	A data model change for a future release.	
resource_unit	A data model change for a future release.	
fixed_resource_usage	A data model change for a future release.	

The **features** property is a string containing a space-separated list of licensable feature names that the SKU enables. This list is supplied by Brocade as part of a contractual definition. The possible feature names are listed below.

Feature	Description
afm	Enable Brocade Application Firewall.
anlyt	Enable Realtime Analytics.
apt	Enable Advanced Web Accelerator.
auto	Enable Autoscaling.
bwm	Enable Bandwidth Management classes.
cache	Enable Web Caching.
comp	Enable Compression.
cr	Do not limit the user to cut-down RuleBuilder content routing for TrafficScript.
evnts	Enable Events and Actions.
glb	Enable Global Load Balancing.
java	Enable Java.
kcd	Enable Kerberos Constrained Delegation support.
loca	Enable Location support.
moni	Enable Active Monitors.
rate	Enable Rate Shaping classes.
rb	Do not limit the user to RuleBuilder for TrafficScript.
rhi	Enable Route Health Injection support.
safpx	Enable Brocade Application Firewall Proxy.
slm	Enable Service Level Monitoring.
ssl	Enable Secure Socket Layer (SSL).
svcprt	Enable Service Protection classes.
ts	Enable TrafficScript.
xml	Enable XML functions in TrafficScript.

The list below provides licensable features for allowed session persistence algorithms.

Feature	Description
spasp	ASP session persistence.
spip	IP-based session persistence.
spj2	J2EE session persistence.
spkip	Application cookie session persistence.
spnam	Named node session persistence.
spsar	Transparent session affinity.
spssl	SSL session ID session persistence.
spuni	Universal session persistence.
spxze	X-Zeus-backend cookie session persistence.

The list below provides licensable features for allowed load balancing algorithms.

Feature	Description
lbrnd	Random.
lbrob	Round robin.

Feature	Description
lbwrob	Weighted round robin.
lbcon	Least connection based.
lbwcon	Weighted least connection based.
lbrsp	Fastest response times.
lbcel	Array of cells.
lbfail	Balance failure class (used only for testing and debugging purposes).
lbone	Always choose first node in a pool (used only for testing and debugging purposes).

The response body to a GET contains a JSON structure representing the properties of the **sku** resource. For example:

```
{
    "add_on_skus": [
        "ADD-FIPS",
        "ADD-WAF",
        "ADD-WEBACCEL"
    ],
    "csp": xxxxx,
    "ent": xxxxx,
    "feature_tier": "STM-400",
    "features": "rb anlyt loca java ts cr comp moni ssl svcprt evnts cache xml glb bwm rate slm auto rhi
kcd spxze spsar spkip spip spssl spuni spnam spj2 spasp lbrnd lbrob lbwrob lbcon lbwcon lbrsp lbcel lbfail
lbone",
    "info": "",
    "pricing_model": "xxxxxx",
    "status": "Active",
    "stm_sku": "STM-400"
}
```

user Resource

A **user** resource describes a Services Director administrative user. There is no system of privileges; all active Services Director users have full read and write access to the inventory database.

The **user** also stores AWS credentials for the user.

The **user** resource supports the following properties:

Property	Description	Supported
aws_access_key	Optional. The user name for AWS operations.	Create/Update
aws_secret_access_key	Optional. The password for AWS operations.	Create/Update
password	Mandatory when creating a user. Cannot be empty. The user password string. This is supplied in clear text, although it is stored as a *salted hash* value. NOTE This is not included in output from a GET command.	Create/Update
status	The user status: Active or false. A non-active user cannot be used for requests.	Update

It is unlikely that you will create a user.

The **password** property is mandatory when creating a user, and it must not be empty. There are no other quality constraints.

> **NOTE**
> For internal implementation reasons, **user** resource names are effectively case-insensitive, although other resource names are case-sensitive. Therefore, resources named admin and Admin refer to the same underlying **user** resource.

The response body to a GET contains a JSON structure representing the properties of the **user** resource. For example:

```
{
    "aws_access_key": "XXXXXXXXXXXXXXXXXXXX",
    "aws_secret_access_key": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "status": "Active",
    "username": "admin"
}
```

user_data Resource

A **user_data** resource describes a cloud registration. This includes the user data required to create the first instance in a cluster.

The **user_data** resource supports the following properties:

Property	Description	Supported
date_created	The timestamp for the creation of the cloud registration.	Read Only.
owner	The owner resource for the cloud registration. See owner Resource on page 163.	Create/Update
registration_policy	The registration policy for this cloud registration.	Create/Update
tag	The display name for the cloud registration.	Create/Update
user_data	The user data for the cloud registration, presented with base64 encoding.	Read Only.
user_data_id	The ID for the cloud registration.	Read Only.

The response body to a GET contains a JSON structure representing the properties of the **user_data** resource. For example:

```
{
    "date_created": "2016-09-26 08:53:08",
    "owner": "Owner-X287-ZB46-SCYV-HEZT",
    "registration_policy": "Policy-T1HN-946O-EO5T-25Z8",
    "tag": "ClusterFirst",
    "user_data": "dGltZXpvbmU9IkV0Yy9VVEMi...<truncated>...1VC0yNVo4Ig==",
    "user_data_id": "UserData-42RL-FJA6-11T5-HHCX"
}
```

When a GET is performed, a new password is assigned automatically, and the encoded user data updates.

> **NOTE**
> Brocade recommends that you perform a GET between each deployment, to ensure that the password is not reused.

version Resource

A **version** resource describes a specific Traffic Manager installation file. A **version** resource is specific to a Traffic Manager version and architecture, and you can have multiple **version** resources for the same Traffic Manager version number (corresponding to different architectures). Creating, altering, or deleting the **version** resource does not affect the existence of the actual version file. You are responsible for ensuring that the Traffic Manager installation file is available to the Services Director servers.

> **NOTE**
> If you create **version** resources solely for use with externally-deployed instances, the Services Director does not use the **version_filename** property. In this case, you do not need to make the Traffic Manager installation file available. For more information about externally-deployed instances, refer to Properties for an Externally-Deployed Instance on page 117.

When creating a **version** resource, you can specify the following properties.

Property	Description	Supported Actions
info	An optional descriptive string.	Create/Update
version_filename	The Traffic Manager installation file (not the complete path).	Create/Update
status	The status of this resource: Active or Inactive.	Update
md5sum	The md5sum of the version tarball. This is an optional field that is used to validate STM tarballs during deployment. If you do not provide a value, the Services Director will calculate the md5sum itself during the first deployment of that version, and continue to use that value for later validations.	Create/Update

You can change the **version_filename** and **info** properties by updating an existing **version** resource.

You can mark the resource as inactive by changing the **status** property to Inactive.

> **NOTE**
> When deploying a Traffic Manager instance, if a Traffic Manager installation file with the same filename as the **version_filename** property is already present on the target instance host, it is reused to save both time and bandwidth. The Services Director does not attempt to verify the metadata or content of the installation file. As such, if you replace an installation file with different content under the same filename, you should manually remove any cached installation files with that name from your instance hosts.

Using the REST API to Check Status

You can check the status of various entities through the REST API. Some of these are directly associated with inventory items, while others are more universal:

- **files**
- **threads**
- **email**

Checking the Status of Files

You can check the state of the files associated with the current Services Director deployment by performing a GET request on the following URI:

```
/api/tmcm/2.5/status/files
```

> **NOTE**
> This URI supports only GET requests.

Example output is shown below:

```
    % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                    Dload  Upload   Total   Spent    Left  Speed
100   615   0   615    0    0   2266      0 --:--:-- --:--:-- --:--:--  2383
{ "licenses" : [ { "filename" : "/var/cache/ssc/fla-ssl",
        "href" : "/api/tmcm/2.5/license/fla-ssl",
        "name" : "fla-ssl",
        "present" : true
    } ],
  "versions" : [ { "filename" : "/var/cache/ssc/ZeusTM_101_Linux-x86_64.tgz",
        "href" : "/api/tmcm/2.5/version/ZeusTM_103_Linux-x86_64",
        "name" : "ZeusTM_101_Linux-x86_64",
        "present" : true
      },
      { "filename" : "/var/cache/ssc/ZeusTM_100_Linux-x86_64.tgz",
        "href" : "/api/tmcm/2.5/version/ZeusTM_103_Linux-x86_64",
        "name" : "ZeusTM_100_Linux-x86_64",
        "present" : true
      },
      { "filename" : "/var/cache/ssc/ZeusTM_98_Linux-x86_64.tgz",
        "href" : "/api/tmcm/2.5/version/STM98",
        "name" : "STM98",
        "present" : true
      }
    ]
}
```

This URI produces a response with the following properties:

- **licenses** - An array of objects, one for each active license, each with the following properties:

  - **name** - The name of the license.
  - **href** - The URI of the **license** resource.
  - **filename** - The absolute path of the file associated with the license.
  - **present** - true if the file is found or false if not.

- **versions** - An array of objects, one for each active version, each with the following properties:

  - **name** - The name of the version.
  - **href** - The URI of the **version** resource.
  - **filename** - The absolute path of the file associated with the version.
  - **present** - true if the file is found or false if not.

  > **NOTE**
  > This operation applies to only the specific Services Director on which the REST request is run.

Checking the Status of Threads

You can view a summary of the threads associated with the current Services Director deployment by performing a GET request on the following URI:

```
/api/tmcm/2.5/status/threads
```

> **NOTE**
> This URI supports only GET requests.

Example output is shown below:

```
    % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                    Dload  Upload   Total   Spent    Left  Speed
100    84   0    84    0    0    465      0 --:--:-- --:--:-- --:--:--   491
```

```
{ "active_thread_count" : 0,
  "queue_length" : 0,
  "thread_count" : 54,
  "thread_info" : {  }
}
```

Sending a Test Notification Email

You can send a test email to the configured notification email address by performing a GET request on the following URI:

```
/api/tmcm/2.5/status/email
```

> **NOTE**
> This URI supports only GET requests.

Understanding REST Request Errors

If the Services Director REST service is unable to handle or interpret a request, it returns a HTTP response with an appropriate HTTP error code. The response body contains a JSON data structure that describes the error:

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

For certain error conditions, the **error_info** property can contain a data structure to further describe the error.

Metering and Monitoring the Services Director

Usage Metering and Activity Metrics (CSP Customers Only)

The Services Director automatically meters usage on a regular basis, and it optionally sends this information to Brocade for billing purposes. By default, it records this information once per hour.

If a Traffic Manager instance is active, the Services Director polls it to obtain total throughput and peak activity metrics. The Services Director creates a metrics log file with one line of metrics data for each Traffic Manager instance. Each line of metrics data records the name of the instance, the time elapsed since the resource was created, and the polled metrics. If an instance is not active, only the elapsed time is recorded.

If you want to generate usage or billing information, typically you process all metering log files and aggregate the results. You should use caution when aggregating data results for billing since metering records include failed deployments.

> **NOTE**
> Generating log files has a cumulative impact on disk
> space.

The Services Director collects metering data from Traffic Manager instances as follows:

- Instances that are at version 9.4 or earlier (or have no REST API enabled) have their metering collected through SNMP.
- Instances that are at version 9.5 or later with the REST API enabled have their metering collected through their REST API. If REST-based metering fails (or is not possible), the Services Director falls back to collecting using SNMP. Any metering issues will be included in the warning logs, as before.

The Services Director records the most recent metrics information for each instance in the inventory database. You can obtain this data using the REST API. The REST API does not supply bulk metrics data.

The Metering Log file is structured as follows:

- The first row contains version data for the metering log format. This first line can be ignored by customers. Ignore this line when you aggregate data for billing.
- Each subsequent row records one set of metrics for a Traffic Manager instance, in comma-separated value (CSV) format.
- The final line contains an MD5 hash of the previous lines. Ignore this line when you aggregate data for billing.

Each line of metrics contains the following fields:

Field	Description
Timestamp	The date and time, in UTC format, that the line was written.
Instance ID	The unique instance ID for the Traffic Manager instance.
Instance Tag	This information may be empty but it is included, even if empty.
Owner	Optionally, the owner of the Traffic Manager instance.

Field	Description
Cluster ID	The cluster for the Traffic Manager instance.
Management IP	The management IP address of the Traffic Manager instance.
Instance SKU	The SKU (or SKU combination) assigned to the Traffic Manager instance (at the time of writing to the log).
	The SKU might vary between readings, and variations are not recorded in the metrics log file.
	This property includes a hash of features applicable to the SKU. Ignore these features for billing purposes.
Feature Pack	The feature pack assigned to the Traffic Manager instance (at the time of writing to the log).
Deploy Time	The length of time (in days, hours and minutes) since the instance was deployed.
Throughput	The number of bytes sent by the Traffic Manager instance, as recorded in the SNMP counter.
	This number is cumulative and is reset whenever the Traffic Manager instance is restarted. It is not the throughput since the latest metering action.
	To generate usage or billing information based on throughput, you should set your aggregating script to detect a drop in throughput and designate this as a restart.
	This property is applicable to active Traffic Manager instances only. • For Idle or Inactive instances, it contains a value of 0 (zero) in the log. • For uncontactable instances, it contains a value of -1 in the log.
Peak Throughput	The highest number of bytes sent by the Traffic Manager instance in any second of the previous hour.
	This property is applicable to active Traffic Manager instances only. • For Idle or Inactive instances, it contains a value of 0 (zero) in the log. • For uncontactable instances, it contains a value of -1 in the log.
Peak Requests	The highest number of requests received by the Traffic Manager instance in any second of the previous hour.
	This property is applicable to active Traffic Manager instances only. • For Idle or Inactive instances, it contains a value of 0 (zero) in the log. • For uncontactable instances, it contains a value of -1 in the log.
Peak SSL Requests	The highest number of Secure Socket Layer (SSL) requests received by the Traffic Manager instance in any second of the previous hour.
	This property is applicable to active Traffic Manager instances only. • For Idle or Inactive instances, it contains a value of 0 (zero) in the log. • For uncontactable instances, it contains a value of -1 in the log.
Instance Bandwidth	The bandwidth (in Mbps) allocated to the Traffic Manager instance.
Record Hash	An MD5 or similar hash of the record from the Services Director license file for tamper detection. Ignore this for billing purposes.

If metrics are not collected for a period of time, peaks for the missing time are not recorded. If you reduce the metering interval, the peak values are still relative to the previous hour rather than the time since metrics were last collected.

Creating Metering Logs (CSP Customers Only)

By default, the Services Director retains and archives metering log files in the `/var/log/ssc/` metering directory. You define the log output location when you first configure the Services Director.

To allow the Services Director to send metering logs to Brocade, enable the *phone home* feature through the REST API or Services Director VA. See the *Brocade Services Director Getting Started Guide*.

> **NOTE**
> The metering phone home service requires DNS to find the phone home server. If you do not have DNS enabled on your Services Director, add the IP address for 'sftp.brocade.com' (currently 144.49.208.38), to the VA's static hosts mapping.

To enable the phone home feature through the REST API, perform a PUT request on the **settings/phone_home** resource, setting the **phone_home_enabled** property to true. This property is set to false by default.

> **NOTE**
> A warning e-mail will be sent every 24 hours if phone-home is enabled and Services Director is unable to connect to the phone-home server.

If the phone home feature is disabled, you must manually extract the metering log files and send them to Brocade. Employing phone home on your system ensures that the process is entirely automatic, secure, and without staffing overhead.

The Services Director employs a **cron** job, created during installation, to run a script that prepares a metering data archive file in this format:

```
<sd_hostname><controller_license_name>.zip
```

The script sends this file to a secure SFTP server at Brocade. The **cron** job runs once a month, at a randomly selected date and time. The Services Director authenticates the destination SFTP server against a pre-configured locally stored host key. You can change this key by placing a new value in:

```
~/.ssh/known_hosts
```

If the host key is not found, or does not match the SFTP server, the phone home process stops.

The Services Director makes two attempts to send the log file. If the first attempt fails, the Services Director waits a random period of up to 12 hours and then tries again. If the second attempt fails, the Services Director sends an alert to the email notification list. Equally, if the file is successfully sent, the Services Director sends an alert to the email notification list announcing the success.

Errors and warnings pertaining to the phone home mechanism are all logged to the file **metering_phone_home.log**. This file provides a source of debugging information for problem resolution.

> **NOTE**
> You can also extract metering logs using the Services Director VA. See the *Brocade Services Director Getting Started Guide*.

Metering Settings

You can view and modify various metering settings Services Director in the **/api/tmcm/2.5/settings/metering** resource. These settings do not normally need to be modified from their default values. For details, see .

You set metering settings for the Services Director as follows:

- **alerts_and_notifications** - A Boolean that controls how metering alerts and notifications are handled. The default value is true. This indicates that the metering health icon will be displayed in the header for the Services Director VA graphical user interface, and that daily emails about metering warnings will be sent.

- **meter_interval** - The period of time, in seconds, between metering actions. The range is from 1-3600. The default value is 3600 seconds (1 hour).

- **snmp_enabled** - A Boolean that controls whether SNMP is used. SNMP is used to gather certain types of information (such as metering) from the Traffic Managers in the estate of the Services Director. The default value is true.

- **log_check_interval** - The period of time, in seconds, between checks for log space. The range is from 1-3600. The default value is 3600 seconds (1 hour).

Creating Metering Records Manually

The Services Director includes a command line utility, **prepare_metering_records** , to manually prepare metering records for processing. You can use this script to create the metering archive file ready for transmission to Brocade.

To create metering records

- At the system prompt, enter:

      prepare_metering_records [--help] [--force]

  --force - Runs the script without prompting for user input.

Creating Metering Records Using the Phone Home Script

You can run the **metering_phone_home** script to manually perform a phone home operation. This script is stored in the bin directory of the Services Director installation directory. For example, in:

```
INSTALLROOT/bin
```

To run the metering phone home script

- At the system prompt, enter:

      metering_phone_home -v/--verbose -n/--noretry

  -v/--verbose - Redirects logging to **stdout** instead of a file.

  -n/--noretry - Prevents further attempts at sending.

Health and Performance Monitoring

Each Services Director in your deployment monitors the health of all other peer Services Directors and deployed Traffic Manager instances recorded in the inventory database. If a failure occurs, the Services Director records a warning entry in the event log and sends an email notification to any system administrators declared in the database.

> **NOTE**
> You can configure the "From" e-mail address of alert e-mails. This address can be set in INSTALLROOT/conf/email_config.txt, in the **common** section, as **from_address**. The symbol "$fqdn" will be replaced by the fully-qualified domain name of the instance host, or an IP address where Universal Licensing is used. The other sections in this file should not normally be modified. For Services Director installs on AWS it is likely that you will need to change this setting to be an address that is resolvable to the instance's public IP.

The Services Director also monitors supported versions of deployed Traffic Manager instances for a number of key performance metrics. You can obtain these metrics through the REST API **monitoring** resource.

A series of inter-controller REST API requests are used to identify the status of each Services Director and Traffic Manager instance in your deployment. You must make sure that each Services Director has network access to all other Services Directors and instance hosts. You must also enable the REST service on your running Traffic Manager instances and record the REST access details in the inventory database.

> **NOTE**
> This process is performed automatically for Traffic Manager instances that are deployed by the Services Director. However, you must enable and record REST API credentials for externally-deployed instances manually to allow monitoring to take place for these instances.

Your Services Directors in an Active state are monitored.

Monitoring Settings

By default, all active Services Directors share the task of monitoring. You can alter this behavior by modifying the Services Director's *monitoring* mode settings in the REST API **manager** resource. These settings do not normally need to be modified from their default values. For details, see Using the REST API with the Services Director on page 123.

You select whether each Services Director individually monitors all other Services Directors and Traffic Manager instances in the deployment, shares the responsibility of monitoring a proportion of Traffic Manager instances with other Services Directors, or performs no monitoring actions at all.

You can view and modify various monitoring interval settings for the Services Director in the REST API **monitoring** resource. These settings do not normally need to be modified from their default values. For details, see Using the REST API with the Services Director on page 123.

You set distinct interval values for monitoring the Services Directoras follows:

- **Monitoring Interval** – The period of time, in seconds, that must elapse between health checks. The default value is 60. A setting of 0 forces the Services Director to use a predefined short interval suitable for deployments that require very frequent monitoring.
- **Failure Identification Interval** – The period of time, in seconds, that must elapse between continuous health check failures before your Services Director determines that a service failure has occurred. This setting helps to prevent against transient network errors being incorrectly identified as service outages. The default value is 180.
- **Overdue Monitoring Warning Interval** – The period of time, in seconds, that must elapse before any pending monitoring actions are considered overdue. This might occur during periods of unusually heavy load. If defined, a breach of this interval causes the Services Director to issue an alert. The default value is 300.
- **Warning Email Interval** – The period of time, in seconds, that must elapse before subsequent email alerts are sent. You can use this setting to avoid large numbers of emails being sent, one for each occasion a warning is triggered. A new email is sent only after this interval has passed. This email contains all monitoring events since the previous email was sent.

Retrieving Monitoring Data

You can retrieve stored monitoring state data from the Services Director by using the REST API **monitoring** resource. This resource is read-only and supports only the REST API GET request method. For details, see Using the REST API with the Services Director on page 123.

You can access the following child elements through the REST API **monitoring** resource:

- **/monitoring/manager** - This element contains monitoring state data for all of your Services Directors, whether or not they have failed.
- **/monitoring/host** - This element contains monitoring state data for all of your service hosts, whether or not they have failed.

- **/monitoring/instance** – This element contains monitoring state data and key performance metrics for your Traffic Manager instances, whether or not they have failed.

- **/monitoring/failures** – This element contains a pair of arrays for failed Services Directors and Traffic Manager instances. You can use this element to retrieve a list of currently failed devices without needing to check the status of each one individually.

Upgrading Services Director VAs and Instance Host VAs

Upgrading the Services Director VA (v2.1 and Earlier)

The required procedure depends on whether your current database is internal or external, and whether you want to use High Availability at Services Director v2.3 (or later).

- If you have an external database and want to use High Availability on your upgraded system, you must convert to using an internal database. High Availability is not supported for external databases. See Upgrading your Services Director VA (External DB and HA required) on page 191.

- If you have an external database but do not want to use High Availability, no special steps are required. See Upgrading your Services Director VA on page 192.

- If you have an internal database, no special steps are required. See Upgrading your Services Director VA on page 192.

Upgrading your Services Director VA (External DB and HA required)

This procedure converts your external database to an internal database, and upgrades your Services Director VA. This enables you to use your existing Services Director VA in an HA pair.

To Upgrade your Services Director VA (External DB and HA)

1. Upgrade your Services Director VA (version 2.1 or earlier) to the Services Director VA v2.1r1 release. See Upgrading your Services Director VA on page 192.

2. Upgrade your Services Director VA v2.1r1 to the Services Director VA v2.4 release. To do this, repeat the procedure described in Upgrading your Services Director VA on page 192.

   > NOTE
   > To support the encryption of stored passwords for Traffic Manager instances, Services Director will encrypt passwords during the upgrade of the Virtual Appliance. A default master password of *master1M@* is used to do this. It is strongly recommended that you update the master password after an upgrade. For a Services Director VA installation, this process is performed from the **System** > **Security** page. See also Working with the Master Password on page 201.

3. Take a **mysqldump** of the external database (.sql extension).

4. Log into your upgraded Services Director VA and start the CLI:

    ```
    login as: admin
    Brocade Services Director
    Last login: Tue Apr 8 22:17:09 2014 from 10.32.26.136
    amnesiac > enable
    amnesiac # configure terminal
    amnesiac (config) #
    ```

5. Reconfigure the VA to use an internal database, using the CLI command:

    ```
    amnesiac (config) # ssc database use-local
    ```

6. Import the database dump file with the CLI command:

    ```
    amnesiac (config) # ssc database local db-file import
       scp://<user>:<password>@<host+path to file> force
    ```

7. Log in to the Services Director VA, and navigate to the **Services** > **Manage HA** page.

8. Use the **Create Primary** command to upgrade your upgraded Services Director to a Primary Services Director node. This node is then ready for High Availability. For details, see *Brocade Services Director Getting Started Guide*.

Upgrading your Services Director VA

To upgrade a Services Director VA from 2.1 (or earlier), you must perform two upgrades:

*   First, you must upgrade your Services Director VA to version 2.1 r1.

*   Then, you must upgrade your Services Director VA to the required later release.

    NOTE
    You *cannot* upgrade Services Director VA v2.1 (or earlier) directly to version 2.2 or later.

    NOTE
    During an upgrade to version 2.3 or later, the **cluster_type** property is added to each existing Cluster resource and set to User Created.

    NOTE
    To support the encryption of stored passwords for Traffic Manager instances, Services Director will encrypt passwords during the upgrade of the Virtual Appliance. A default master password of *master1M@* is used to do this. It is strongly recommended that you update the master password after an upgrade. For a Services Director VA installation, this process is performed from the **System** > **Security** page. See also Working with the Master Password on page 201.

The procedure below can be used for both steps.

To Upgrade the Services Director

1. Download the required Services Director VA upgrade image from the Brocade Support site.

2. Place the image file on a server that is accessible with HTTP, SCP, or FTP.

3. Connect to the Services Director VA using a terminal session and start *enable mode* :

    ```
    login as: admin
    Brocade Services Director
    Last login: Tue Apr 8 22:17:09 2014 from 10.32.26.136
    amnesiac > enable
    amnesiac #
    ```

4. Locate the upgrade image, and retrieve it:

   ```
   amnesiac # image fetch http://<host-name>/<path>/<image-name> <local-name>
   ```

   Where:

   - <image-name> is the name of the file you want to fetch.
   - <local-name> is an optional local name for the downloaded image file. This takes the form "<name>.img". For example, "ssc_23.img". If this is not specified, the downloaded image is renamed to "image.img" once it downloads.

5. Apply the upgrade, using the local name from step 4. For example, "image.img":

   ```
   image upgrade image.img
   ```

   The image upgrade may take a long time and the session will remain unresponsive during this step. However, the Services Director itself will remain operational for instance licensing, monitoring, and so on.

   Once the process is complete, a message is displayed:

   ```
   "Installed image 'image.img' on partition 2. The 'reload' command will load software version
   '<version details>'"
   ```

6. Restart the node:

   ```
   reload
   ```

7. Wait for the image to reload.

   > **NOTE**
   > The Services Director VA will restart at this point. This may cause instances to briefly enter the licensing grace period, but they will return to normal operation once the restart is completed.

8. Access the Services Director VA via Web UI or command line interface to check that the node is operational. The version on the title bar should indicate the new software version.

Upgrading an HA Pair of Services Director VAs (v2.2 or Later)

A High Availability pair of Services Director VAs must be upgraded using the procedure below.

> **NOTE**
> The same general procedure can be used for a standalone Services Director VA.

To Upgrade an HA Pair of Services Directors

1. Download the Services Director VA upgrade image from the Brocade Support site.

2. Place the image file on a server that is accessible with HTTP, SCP, or FTP.

3. Connect to the Active node using a terminal session and start *enable mode* :

   ```
   login as: admin
   Brocade Services Director
   Last login: Tue Apr 8 22:17:09 2014 from 10.32.26.136
   amnesiac > enable
   amnesiac #
   ```

4. Locate the upgrade image, and retrieve it:

    ```
    amnesiac # image fetch http://<host-name>/<path>/<image-name> <local-name>
    ```

    Where:

    - <image-name> is the name of the file you want to fetch.
    - <local-name> is an optional local name for the downloaded image file. This takes the form "<name>.img". For example, "ssc_23.img". If this is not specified, the downloaded image is renamed to "image.img" once it downloads.

5. Apply the upgrade, using the local name from step 4. For example, "image.img":

    ```
    image upgrade image.img
    ```

    The image upgrade may take a long time and the session will remain unresponsive during this step. However, the Services Director itself will remain operational for instance licensing, monitoring, and so on.

    Once the process is complete, a message is displayed:

    ```
    "Installed image 'image.img' on partition 2. The 'reload' command will load software version
    '<version details>'"
    ```

6. Restart the node:

    ```
    reload
    ```

7. Wait for the Active node to reload the upgraded image.

    > **NOTE**
    > The node will restart at this point. This may cause instances to briefly enter the licensing grace period, but they will return to normal operation once the restart is completed.

    > **NOTE**
    > During an upgrade, the **cluster_type** property is added to each existing Cluster resource and set to User Created.

    > **NOTE**
    > To support the encryption of stored passwords for Traffic Manager instances, Services Director will encrypt passwords during the upgrade of the Virtual Appliance. A default master password of *master1M@* is used to do this. It is strongly recommended that you update the master password after an upgrade. For a Services Director VA installation, this process is performed from the **System** > **Security** page. See also Working with the Master Password on page 201.

8. Access the Services Director VA (using its Service Endpoint Address) via Web UI or command line interface to check that the node is operational. The version on the title bar should indicate the new software version.

9. Repeat steps 3-7 on the Standby node. *Do not repeat step 8.*

10. Access the Services Director VA (using its IP Address) via Web UI or command line interface to check that the node is operational. The version on the title bar should indicate the new software version.

    The upgrade of the HA pair is complete.

Upgrading Instance Host VAs

You cannot directly upgrade existing instance hosts to the latest instance host version.

Instead, you must create new instance hosts and then migrate the Virtual Traffic Managers on the original instance hosts over to the new instance hosts.

To do this, perform the procedure outlined in

Migrating Instances to a New Instance Host (v2.4 and Later)

The following procedure enables you to migrate Traffic Manager instances from an instance host (v2.3r1 and earlier) to an instance host (v2.4 and later).

> **NOTE**
> You cannot directly upgrade existing instance hosts to the latest instance host version.

> **NOTE**
> If your network involves older versions of the Services Director VA software, or your instance hosts are not all contained in a single subnet, contact Brocade support.

This procedure requires that all instance hosts are in a single subnet, and that the starting Services Director VA software version is 1.3 or above.

For the purposes of providing a worked example, this procedure assumes the following starting configuration (other configurations are supported):

- An HA pair of Services Director VAs (v2.3r1 or earlier).
- Two instance host VAs (from v2.3r1 or earlier).
- Three or more Virtual Traffic Managers (vTMs) deployed across the two instance hosts.

For this procedure, you need the following information:

- The hostname/IP address for each of the original instance hosts (v2.3r1 or earlier).

  In this example, these are **IH-01.cam.demo.com** and **IH-02.cam.demo.com**.
- The interface addresses (eth0, eth1 and eth2) for each of the original instance hosts.

  > **NOTE**
  > The eth0 address will be the same as the host's IP address in a IP-based system.

- A temporary hostname/IP address for each of the new v2.4 instance hosts.

  In this example:

  IH-01.cam.demo.com is replaced by **IH-03.cam.demo.com**.

  IH-02.cam.demo.com is replaced by **IH-04.cam.demo.com**.
- Temporary interface addresses (eth0, eth1, eth2) for each of the new instance hosts.

  > **NOTE**
  > At the end of the migration procedure, each new instance host will use the hostname/IP address and interfaces of the original instance host that it replaces. That is, all temporary properties will be discarded.

- The names of the vTMs on each of the original instance hosts. In this example:

  IH-01.cam.demo.com is host for instances **VTM-01** and **VTM-02**.

  IH-01.cam.demo.com is host for instances **VTM-03** and **VTM-04**.

The general procedure is shown below.

> **NOTE**
> Following this sequence of steps closely is essential if you wish to minimise any loss of service during the migration process.

1. Upgrade the HA pair of Services Director VAs. See Upgrading the HA pair of Services Director VAs on page 196.

2. Deploy two new instance hosts at v2.4. See Deploying New Instance Hosts on page 196.

3. Configure the new instance hosts with the temporary hostname and temporary interface addresses. See Configuring the New Instance Hosts on page 196.

4. Add the new instance hosts to the Services Director VA.

5. Migrate the vTMs on the original instance hosts to the new instance hosts. See Migrating Instances to a New Instance Host (v2.4 and Later).

6. Access the Services Director VA graphical user interface (GUI) to view the migrated vTMs. See Access the Services Director VA on page 200.

After completing this process, you may choose to decommission the old instance hosts.

Upgrading the HA pair of Services Director VAs

Upgrade the Services Director VAs, as described in:

- Upgrading the Services Director VA (v2.1 and Earlier) on page 191.
- Upgrading an HA Pair of Services Director VAs (v2.2 or Later) on page 193.

Upgrade the Active Services Director VA first, then the Standby.

Deploying New Instance Hosts

You must deploy a new v2.4 instance host for each of the original instance hosts.

Each original instance host and its vTMs must be visible through the graphical user interface (GUI) of the Services Director VA. See the *Brocade Services Director Getting Started Guide* for details.

To Deploy a New Instance Host

1. Obtain the latest OVA file for the instance host from Brocade.

2. Install the first new instance host, using the OVA to create a virtual machine using vSphere.

   > **NOTE**
   > The new instance host is assigned an initial IP address by vSphere. You will assign a different IP address during its configuration.

3. Repeat for the second new instance host.

Configuring the New Instance Hosts

You must configure each new instance host in turn, using the standard procedure. That is:

- Configure the hostname on the new instance host. This must be different from the original instance host. For example: **IH-03.cam.demo.com**.

- Configure the eth0 and eth1 interfaces of the new instance host. These must be different from the interfaces for the original instance host.

- Configure the DNS on the new instance host (DNS-based systems only).

- Establish passwordless SSH communication between the new instance host and the Active Services Director VA.

> **NOTE**
> The Active Services Director VA now has access to both the original and new instance hosts.

- Establish passwordless SSH communication between the new instance host and the Standby Services Director VA.

> **NOTE**
> The Standby Services Director VA now has access to both the original and new instance hosts.

This entire configuration process is described in detail in Configuring an Instance Host VA from a Terminal Session on page 68.

Configuring the New Instance Hosts

You can then add the two new instance hosts to the Services Director VA's graphical user interface, as described in Adding a Configured Instance Host to the Services Director VA on page 71.

Migrating the vTMs on an Instance Host

You can now migrate the vTMS on each original instance host in turn.

Do not attempt to migrate both original instance hosts simultaneously. This enables other vTMs in the cluster to handle traffic from transitioning vTMs during the migration, and therefore minimise any loss of service.

To Deploy a New Instance Host

1. Log into the Active Services Director VA (for example, **SD-01.cam.demo.com**) using a terminal session.
2. Start the CLI in the terminal session.
3. Stop all vTMs on the first original instance host (**IH-01**) from the Services Director VA CLI.

   To do this, transition each vTM on the instance host individually to the Idle state using the **ssc instance stop instance-name** CLI command. For example:

   ```
   SD-01 (config) # ssc instance stop instance-name VTM-01
   SD-01 (config) # ssc instance stop instance-name VTM-02
   ```

   See the *Brocade Services Director Command-Line Interface Reference Manual*.

   > **NOTE**
   > Alternatively, the vTMs can be stopped using the Services Director VA GUI. See the *Brocade Services Director Getting Started Guide*.

4. Migrate the vTMs on the first original instance host to the first new instance host.

    For example, to migrate the vTMs on **IH-01.cam.demo.com** to **IH-03.cam.demo.com**:

    ```
    SD-01 (config) # ssc host host-migrate from IH-01.cam.demo.com to IH-03.cam.demo.com
      migrating vTM VTM-01 to IH-03.cam.demo.com ......
      migrated vTM VTM-01 to IH-03.cam.demo.com
      migrating vTM VTM-02 to IH-03.cam.demo.com ......
      migrated vTM VTM-03 to IH-03.cam.demo.com
    Migrated vTMs from IH-01.cam.demo.com to IH-03.cam.demo.com successfully
    ```

    > **NOTE**
    > Ensure that you wait for the process to complete. This may take several minutes.

5. Shut down the original instance host.

6. Start a terminal session on the new instance host.
    In this example, start the terminal session on **IH03.cam.demo.com**. Use the **sscadmin** user. The default password is
    "password".

7. For a DNS-based system, set the hostname and domain of the new instance host to match the hostname and domain of the
    original instance host.

    For example, to **IH-01.cam.demo.com**.

    ```
    $ host hostname IH-01 domain-name cam.demo.com
    ```

    > **NOTE**
    > For an IP-based system, no change is required.

8. Change the management (eth0) IP address, netmask and gateway of the new instance host to match those on the original
    instance host.

    ```
    $ host interface eth0 ip <ip-addr> netmask <netmask> gateway <ip-gateway> auto
    ```

    > **NOTE**
    > Your terminal session will be lost at this point, as the IP address of the instance host has changed. This change also
    > disrupts SSH passwordless communication between the Services Director and the new instance host. You must re-
    > establish these credentials later.

9. Start a new terminal session for the instance host using its new management (eth0) IP address. Use the **sscadmin** user. The
    default password is "password".

10. Change the front-facing (eth1) IP address, netmask and gateway of the new instance host to match those on the original
    instance host.

    ```
    $ host interface eth1 ip <ip-addr> netmask <netmask> gateway <ip-gateway> auto
    ```

11. Log into the Active Services Director VA and access its CLI.

12. You now need to tidy up the list of known-hosts on the Active Services Director. To do this:

    1. Show the known hosts and locate the entries for the first original instance host and the first new instance host. For example:

       ```
       SD-01 (config) # show ssh client known-hosts
       IH-01.cam.demo.com,<IP_address>,
           <rest of entry>
       IH-02.cam.demo.com,<IP_address>,
           <rest of entry>
       IH-03.cam.demo.com,<original_IP_address>,
           <rest of entry>
       IH-04.cam.demo.com,<IP_address>,
           <rest of entry>
       SD-01 (config) #
       ```

       NOTE
       The entry for the first new instance host will reference its initial hostname and IP address, not the new ones established in steps 7-9. In this example, it is **IH-03**.

13. Delete the entries for the first original and first new instance hosts using the following command:

    ```
    SD-01 (config) # no ssh client known-host <identifier from the host entry>
    ```

    Each instance host is identified by the first property in its entry, which might be a hostname or an IP address. In this example:

    ```
    SD-01 (config) # no ssh client known-host IH-01.cam.demo.com
    SD-01 (config) # no ssh client known-host IH-03.cam.demo.com
    ```

14. You must now re-authorize passwordless SSH access to the first new instance host.

    NOTE
    The new instance host (originally **IH-03**) is now using the hostname/IP address of the first original instance host (it is now **IH-01**).

    To do this, slogin to the first new instance host, and confirm the authenticity of this instance host. For example:

    ```
    SD-01 (config) # slogin root@IH-01.cam.demo.com
    The authenticity of host 'IH-01.cam.demo.com (<ip_address>)' can't be established.
    ECDSA key fingerprint is <key>
    Are you sure you want to continue connecting (yes/no)? yes
    Warning: Permanently added 'IH-01.cam.demo.com (<ip_address>)' (ECDSA) to the list of known hosts.
    Brocade Services Director Host
    Last login: <timestamp>
    root@IH-01:~#
    ```

15. Log out of the instance host.

    ```
    root@IH-01:~# logout
    Connection to IH-01.cam.demo.com closed.
    SD-01 (config) #
    ```

16. Repeat steps 11-14 on the Standby Services Director.
    After you finish, both the Active and Standby Services Directors will have replaced the first original instance host (**IH-01**) with an identical v2.4 instance host (also called **IH-01**) that includes all of the original vTMs. Both **IH-03** and **IH-04** will still be present.

17. You must now start each of the vTMs on the new instance host (IH-01) from the Services Director VA CLI. To do this, transition each vTM on the instance host individually to the Active state using the ssc instance start instance-name CLI command. For example:

    ```
    SD-01 (config) # ssc instance start instance-name VTM-01
    SD-01 (config) # ssc instance start instance-name VTM-02
    ```

    See the *Brocade Services Director Command-Line Interface Reference Manual*.

    > **NOTE**
    > Alternatively, the vTMs can be started using the Services Director VA GUI. See the *Brocade Services Director Getting Started Guide*.

18. You must now migrate the vTMs on the second original instance host (**IH-02**) to a second new instance host (**IH-04**).

    To do this, repeat steps 1-16 of this procedure.

    After you finish, both the Active and Standby Services Directors will have replaced the second original instance host (**IH-01**) with an identical new v2.4 instance host (also called **IH-01**) that includes all of the original vTMs.

Access the Services Director VA

After all migration is complete:

1. Access the Active Services Director VA GUI using its Service Endpoint Address in a browser.

2. In the Services Director VA GUI, view the > page.

    This page will include both **IH-01** and **IH-02** instance hosts, but neither **IH-03** nor **IH-04**.

3. Finally, view the > page.

    This page will include all vTMs that were originally deployed using **IH-01** and **IH-02**.

After you have done this, the migration of vTMs from their original instance hosts to new v2.4 instance hosts is complete.

Working with the Master Password

Services Director v2.3 and later uses a master password to encrypt the passwords for Traffic Manager instances.

Storing the Master Password

It is essential that the master password (whether it is the default, chosen yourself or generated automatically) is recorded and can be retrieved. Brocade recommends that this password is recorded in a secure location that is separate from the Services Director.

However, you can also choose to store this password internally:

* If you choose to store the master password internally, the password will be automatically used whenever the Services Director's Virtual Machine restarts. However, you must enter the master password manually when you recover a Services Director from a backup file.
* If you choose to *not* store the password internally, you must enter the master password manually whenever the Services Director's Virtual Machine restarts, and whenever you recover a Services Director from a backup file.

You can change your decision as follows:

* If your software is configured to store the master password, and you wish to change this, delete the file **$SSCHOME/etc/master**.
* If your software is configured NOT to store the master password, and you wish to change this, run one of the following commands:
  - Ubuntu: Run **$SSCHOME/bin/configure_ssc --liveconfigonly**
  - RHEL/CentOS: Run **$SSCHOME/bin/configure_ssc**

In both cases, you are asked whether you wish to store the master password.

Changing the Master Password

NOTE
If you want to *reset* the master password (that is, you do not know what the current master password is), see Resetting the Master Password on page 202.

You can change the master password in the following ways:

* In the Services Director VA for the Active Services Director, from the **Security Settings** page. After you complete this task, you must re-enter the new master password on the Standby Services Director VA. See the *Brocade Services Director Getting Started Guide*.
* In the Command-Line Interface (CLI) on the Active Services Director VA, using the **ssc settings master-password update** command. See the *Brocade Services Director Command Reference*.

NOTE

You must do this on both Services Director nodes in an HA pair, starting with the Active node.

- For a software-only installation, you must issue a PUT request to the Services Director. The body of the PUT request to the REST API has the following format:

```
{
"current_password": "<current password>",
"new_password": "<new password>"
}
```

Where you have additional Services Directors, run the following command on all other Services Directors:

- Ubuntu: **$SSCHOME/bin/configure_ssc –liveconfigonly**
- RHEL/CentOS: **$SSCHOME/bin/configure_ssc**

In both cases, enter the new master password and confirm it when prompted.

NOTE

You do not need to stop and restart Services Directors when modifying the master password on multiple Services Directors.

Resetting the Master Password

NOTE

If you wish to *change* the master password (that is, you know what the current master password is), see Changing the Master Password on page 201.

In the event that a master password is lost, as a final resort there are two ways to reset the master password:

- From the Command-Line Interface (CLI) on each Services Director in an HA pair. See Resetting the Master Password from the Services Director VA CLI on page 202.
- From a software-only installation on Ubuntu or RHEL/CentOS for each Services Director. See Resetting the Master Password on Ubuntu or RHEL/CentOS on page 203.

In both cases, the encrypted administration password for each Traffic Manager is lost. These administration passwords must be set manually after the master password is reset.

Resetting the Master Password from the Services Director VA CLI

From the Services Director VA CLI, you must use the **ssc settings master-password reset password** command on the Active Services Director in the HA pair (or standalone Services Director).

NOTE

This procedure should only be used as a final resort to re-establish a master password. The encrypted administration password for each Traffic Manager known to the Services Director is lost. These individual administration passwords must be set manually after the master password is reset.

To Reset the Master Password from the Services Director VA CLI

1. Start a terminal session on your Active Services Director (or standalone Services Director) using its Service Endpoint Address, and login as the **admin** user.

2. In the Services Director terminal session, start a CLI session:

```
amnesiac > enable
amnesiac # configure terminal
amnesiac (config) #
```

3. Reset the master password for this Services Director:

```
gold-01 (config) # ssc settings master-password reset password New_pass_22 force true
Successfully reset master password
You must manually update all passwords for registered instances
```

> **NOTE**
>
> If you do not include **force true** the reset will fail, as the password cannot be reset on a Services Director with one or more registered/deployed Traffic Managers.

4. Repeats steps 1-3 for your Standby Services Director.

5. On your Active Services Director, reset the administration password for your first Traffic Manager instance. For example, for a Traffic Manager instance called Cobalt-01:

```
gold-01 (config) #  ssc instance update instance-name cobalt-01 admin-password Admin_PW_01
```

The output of this command will show the full status of the Traffic Manager instance (not shown).

> **NOTE**
>
> You can also perform this step by updating the details for the Traffic Manager from the Services Director VA. See the *Brocade Services Director Getting Started Guide* manual.

6. Repeat step 5 for each of the Traffic Managers on the Active Services Director.

The process is complete.

> **NOTE**
>
> You do *not* need to repeat the password reset process on the Standby Services Director.

Resetting the Master Password on Ubuntu or RHEL/CentOS

On a software form-factor installation on Ubuntu or RHEL/CentOS, you must use the **reset_master_password** command on the Services Director.

> **NOTE**
>
> This procedure should only be used as a final resort to re-establish a master password. The encrypted administration password for each Traffic Manager known to the Services Director is lost. These individual administration passwords must be set manually after the master password is reset.

To Reset the Master Password on Ubuntu or RHEL/CentOS

1. Log into your Services Director.

2. Shutdown the Services Director. For example:

```
sudo stop ssc
```

3. Reset the password on the Services Director:

   ```
   $SSCHOME/bin/reset_master_password --force <new password>
   ```

4. Restart the Services Director. For example:

   ```
   sudo start ssc
   ```

5. Set each vTM password via a **PUT** request to its instance resource via the REST API.

   > **NOTE**
   > For Traffic Managers that were deployed by the Services Director, you must include a URL parameter of **deploy=false** to ensure a database-only update. This is not required for externally-deployed Traffic Managers.

6. Log out of the Services Director.

   The process is complete.

   > **NOTE**
   > If you have a more than one Services Directors configured, perform this procedure for each Service Director. However, do not restart any Services Directors until the master password has been reset on all Services Directors.

Appendix: Deploying for Redundancy

Deploying High Availability for the Software Services Director

To create a high availability installation of the software form-factor Services Director, you require:

- Two or more host machines (either virtual or physical) running supported versions of Ubuntu/CentOS Linux. Each will contain a software form-factor Services Director.

- Two or more clustered software form-factor Services Directors, that are installed and licensed. These can operate in either an Active/Passive or an Active/Active configuration.

- A MySQL database for the Services Directors. Replication of this database is recommended, else the database will be a single point of failure.

- Two clustered front-end virtual Traffic Managers (vTMs) in an Active/Passive configuration. These vTMs ensure that manually-issued requests to the REST API of the Services Director will be directed to an Active Services Director. The configured vServer and pools on these vTMs control how traffic reaches your Services Directors.

- A vTM Traffic IP (TIP) address, which acts as the Service Endpoint Address (SEA) for manually-issued REST API requests to the Services Directors.

> **NOTE**
> The TIP/SEA can be in a private network behind a Network Address Translation (NAT) device. Where this is the case, the external SEA is used for all service requests.

Each element of your Services Director installation can be deployed as required within your network, using any combination of virtual machines, physical machines and data centres. Physical diversity of elements (for example, spreading elements across data centres) is likely to increase the resilience of the installation.

The Services Director and its front-end vTMs can be in a private network behind a NAT device. However, the vTMs that will be added to the estate of the Services Director at a later stage cannot be behind a NAT.

Technologies allowing routable connections between native IP addresses in disparate networks (for example, Virtual Private Networks) are supported.

After you have completed your high availability installation of the software form-factor Services Director, you can add Traffic Managers to the estate of your Services Directors, see Populating the Estate of the Services Directors on page 213.

> **NOTE**
> You must ensure that the master password for the Services Directors is recorded securely. This is required during any recovery from failure, see Disaster Recovery for the Software Services Director on page 215.

Example Deployment: Active/Passive Services Directors

A network topology with Active/Passive Services Directors is shown below. Other network topologies are supported.

FIGURE 62 Example Services Director Architecture: Active/Passive Services Directors

In this example, there are:

- Two clustered software form-factor Services Directors, each of which is installed on its own Linux machine.
- A MySQL database that is shared by the clustered Services Directors. This database is replicated.
- Two clustered Virtual Traffic Managers (vTMs), which act as a resilient *front-end* to the Services Directors.
- A known Service Endpoint Address (SEA) for the clustered Services Directors.
- A customer who wants to manually send REST API requests to the clustered Services Directors using the SEA.

    NOTE
    Each element of your Services Director installation can be deployed as required within your network, using any combination of virtual machines, physical machines and data centres. Physical diversity of elements (for example, spreading elements across data centres) is likely to increase the resilience of the installation. Technologies allowing routable connections between native IP addresses in disparate networks (for example, Virtual Private Networks) are supported.
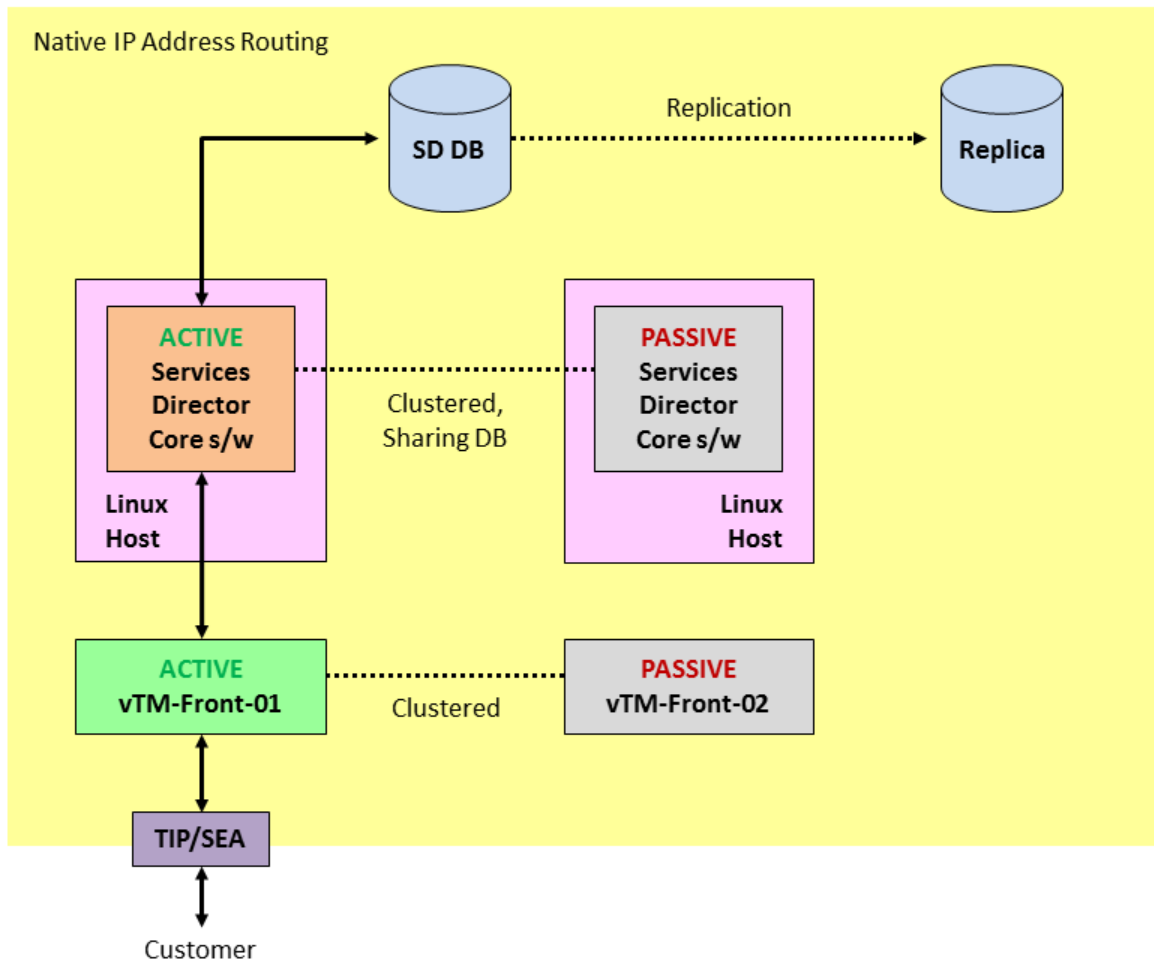
In this example, the Services Directors are required to act as an Active/Passive high availability pair. That is, all requests must go exclusively to the Active Services Director until it fails, at which point the Passive Services Director becomes the Active node. To achieve this Active/Passive configuration, a vServer and two pools are required on the Active front-end vTM:

- A default pool identifies the management address/port for the first Services Director only.
- A failure pool for the first pool identifies the management address/port for the second Services Director only.

    NOTE
    A default monitoring mechanism for the pools is provided. This is ping-based, and operates at a machine level. You may choose to replace this with your chosen monitoring mechanism. Refer to the *Brocade Virtual Traffic Manager* documentation.

The Active front-end vTM raises a Traffic IP (TIP) address, which acts as the SEA for the clustered Services Directors.

The customer sends a request to the SEA, which reaches the Active vTM. This is routed to the Active Services Director to process the request, and a response is returned to the customer.

    NOTE
    The SEA for the clustered Services Director can be in a private network behind a NAT device. However, the vTMs that will be added to the estate of the Services Director at a later stage cannot be behind a NAT.

In this example, both Services Directors are configured to perform licensing, metering and monitoring.

In the event of the failure of the Active Services Director:

- The front-end vTM routes all requests to the Passive Services Director, which becomes the new Active Services Director.
- All requests are routed to the Active Services Director until it fails.
- Licensing, metering and monitoring are performed by the remaining Services Director only.
- The customer continues to manually issue REST API requests to the Services Director using its SEA.

FIGURE 63 Example Services Director Architecture: Active Services Director Failure
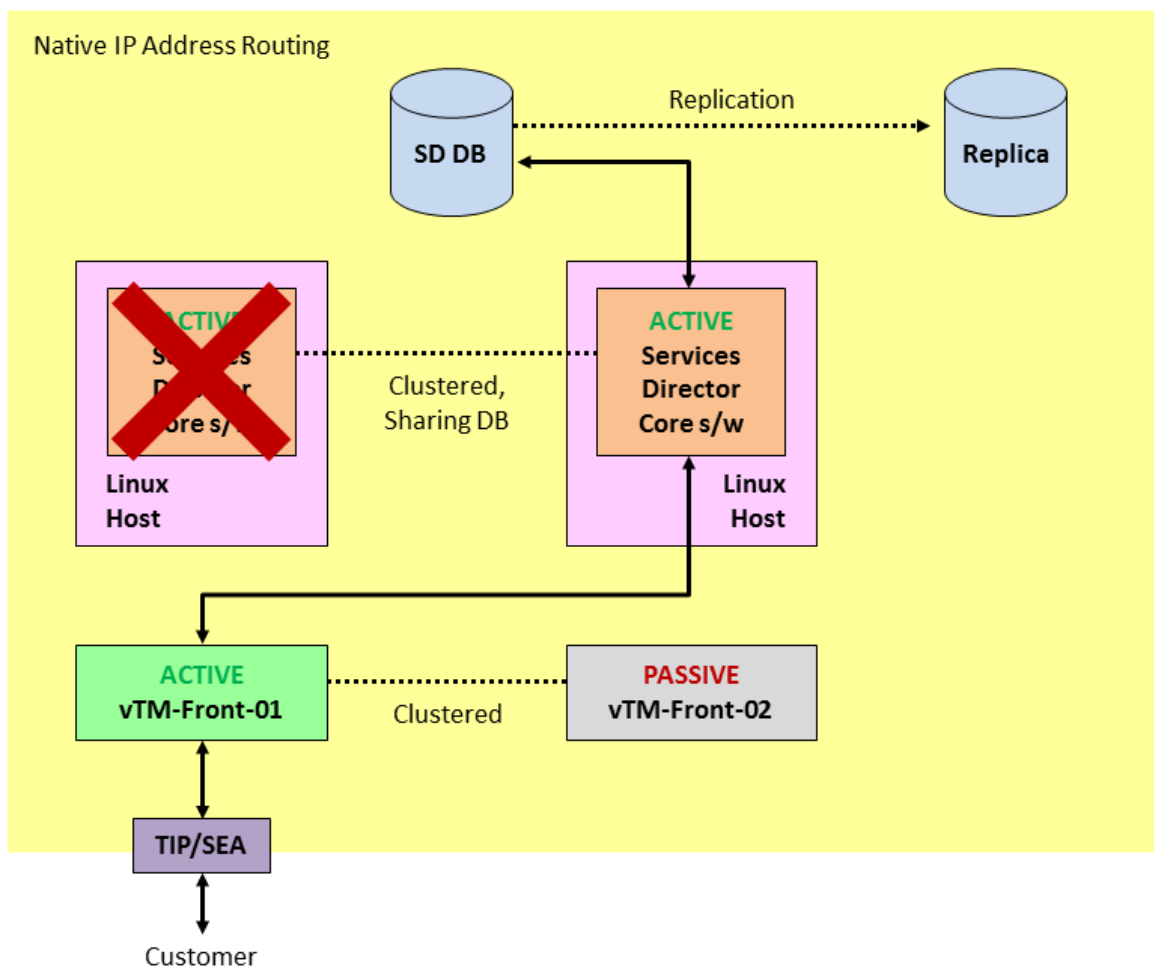
In the event of the failure of the Active front-end vTM:

* The Passive front-end vTM becomes the Active front-end vTM. This is configured with the same vServer pools as the failed front-end vTM.
* The new Active front-end vTM raises a TIP, which matches the SEA for the clustered Services Directors.
* The customer continues to manually issue REST API requests to the Services Director using its SEA.
* The new Active front-end vTM continues to route all requests to the Active Services Director.
* Licensing, metering and monitoring on the Services Directors are unaffected.

FIGURE 64 Example Services Director Architecture: Active vTM Failure
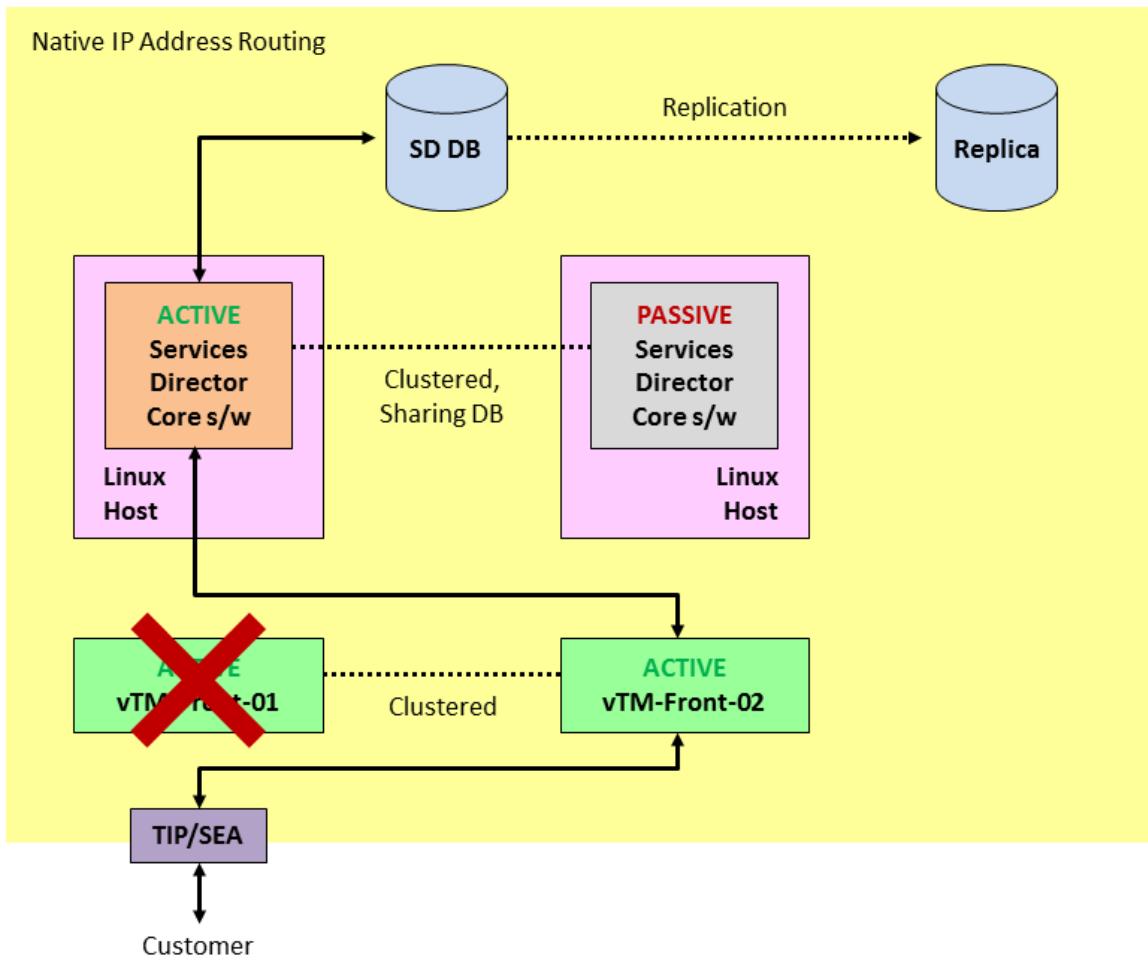
In the event of a failure of the main Services Director database, manually switch the Services Director configuration to use the replica database.

> **NOTE**
> Depending on database activity and the replication schedule, the replica may be behind the main database; re-apply any lost configuration changes.

When the main database becomes available again (or you have replaced it with a new database based on the replica), manually switch the Services Director configuration back, and re-apply any lost configuration changes.

Example Deployment: Multiple Active Services Directors

A network topology with two Active Services Directors is shown below. Additional Active Services Directors can be used if required. Other network topologies are also supported.

FIGURE 65 Example Services Director Architecture: Multiple Active Services Directors
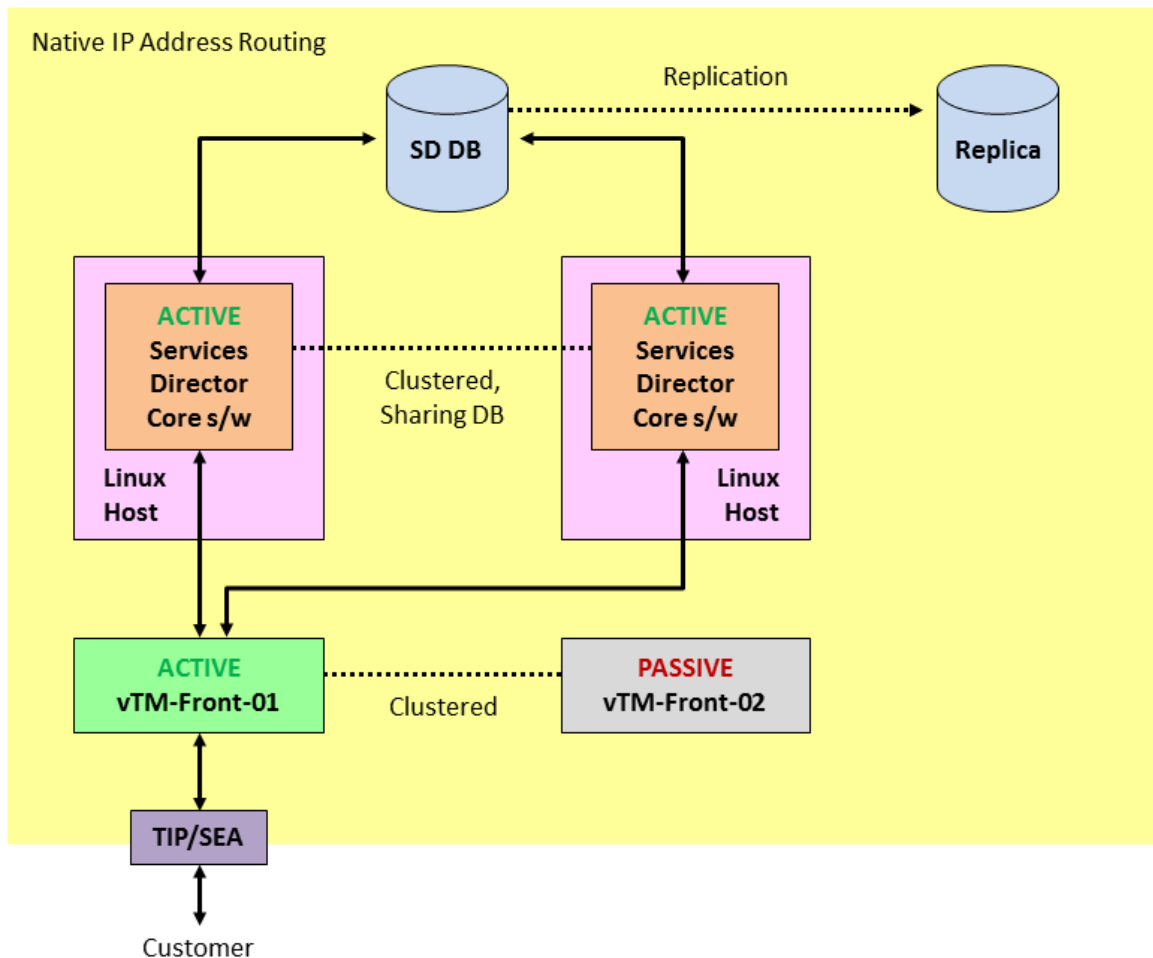
In this example, there are:

- Two clustered software form-factor Services Directors, each of which is installed on its own Linux machine.

  > **NOTE**
  > Additional Services Directors can be added to the cluster if required.

- A MySQL database that is shared by the clustered Services Directors. This database is replicated.
- Two clustered Virtual Traffic Managers (vTMs), which act as a resilient *front-end* to the Services Directors.
- A known Service Endpoint Address (SEA) for the clustered Services Directors.
- A customer who wants to manually send REST API requests to the clustered Services Directors using the SEA.

  NOTE
  Each element of your Services Director installation can be deployed as required within your network, using any combination of virtual machines, physical machines and data centres. Physical diversity of elements (for example, spreading elements across data centres) is likely to increase the resilience of the installation. Technologies allowing routable connections between native IP addresses in disparate networks (for example, Virtual Private Networks) are supported.

In this example, requests are load-balanced between the Services Directors. If one Services Director fails, all requests will be routed to the remaining Services Director. To achieve this configuration, a single vServer and pool is required on the Active front-end vTM. This identifies the management address/ports for both Services Directors, and operates according to the chosen load-balancing algorithm. For example, "round-robin".

> **NOTE**
> A default monitoring mechanism for the pool is provided. This is ping-based, and operates at a machine level. You may choose to replace this with your chosen monitoring mechanism. Refer to the *Brocade Virtual Traffic Manager* documentation.

The Active front-end vTM raises a Traffic IP (TIP) address, which matches the SEA for the clustered Services Directors.

The customer sends a request to the SEA, which reaches the Active vTM. This is routed to a Services Director according to the chosen load-balancing algorithm. The Services Director processes the request, and a response is returned to the customer.

> **NOTE**
> The SEA for the clustered Services Director can be in a private network behind a NAT device. However, the vTMs that will be added to the estate of the Services Director at a later stage cannot be behind a NAT.

In this example, both Services Directors are configured to perform licensing, metering and monitoring.

In the event of the failure of any Services Director:

- The front-end vTM routes all requests away from the failing Services Director. In this case, to the other Services Director.
- Licensing, metering and monitoring are performed by the remaining Services Director only.
- The customer continues to manually issue REST API requests to the Services Director using its SEA.

FIGURE 66 Example Services Director Architecture: Services Director Failure
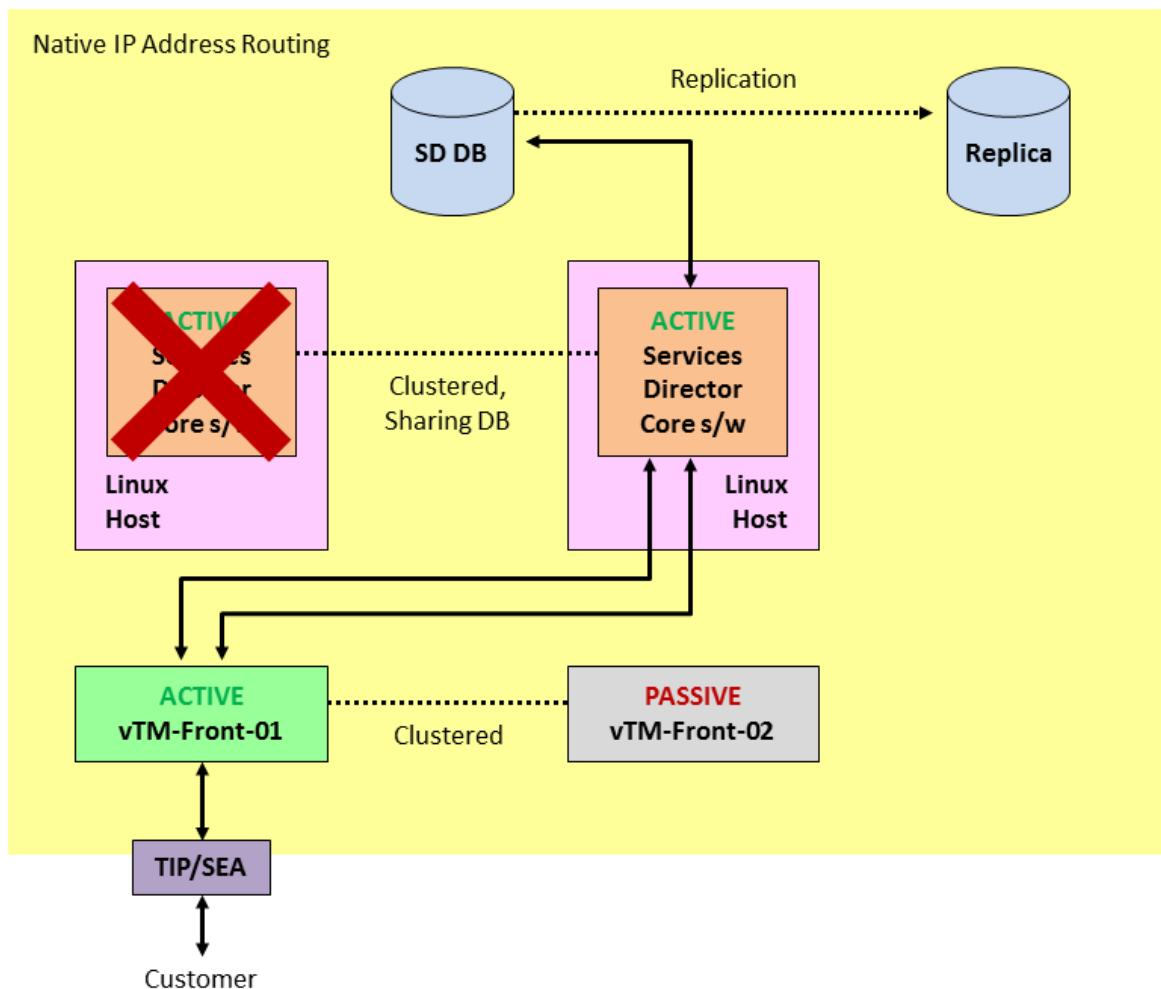
In the event of the failure of the Active front-end vTM:

- The Passive front-end vTM becomes the Active front-end vTM. This is configured with the same vServer pools as the failed front-end vTM.
- The new Active front-end vTM raises a TIP, which matches the SEA for the clustered Services Directors.
- The customer continues to manually issue REST API requests to the Services Director using its SEA.
- The new Active front-end vTM routes all requests to the assigned Services Directors.
- Licensing, metering and monitoring are unaffected.

FIGURE 67 Example Services Director Architecture: Active vTM Failure
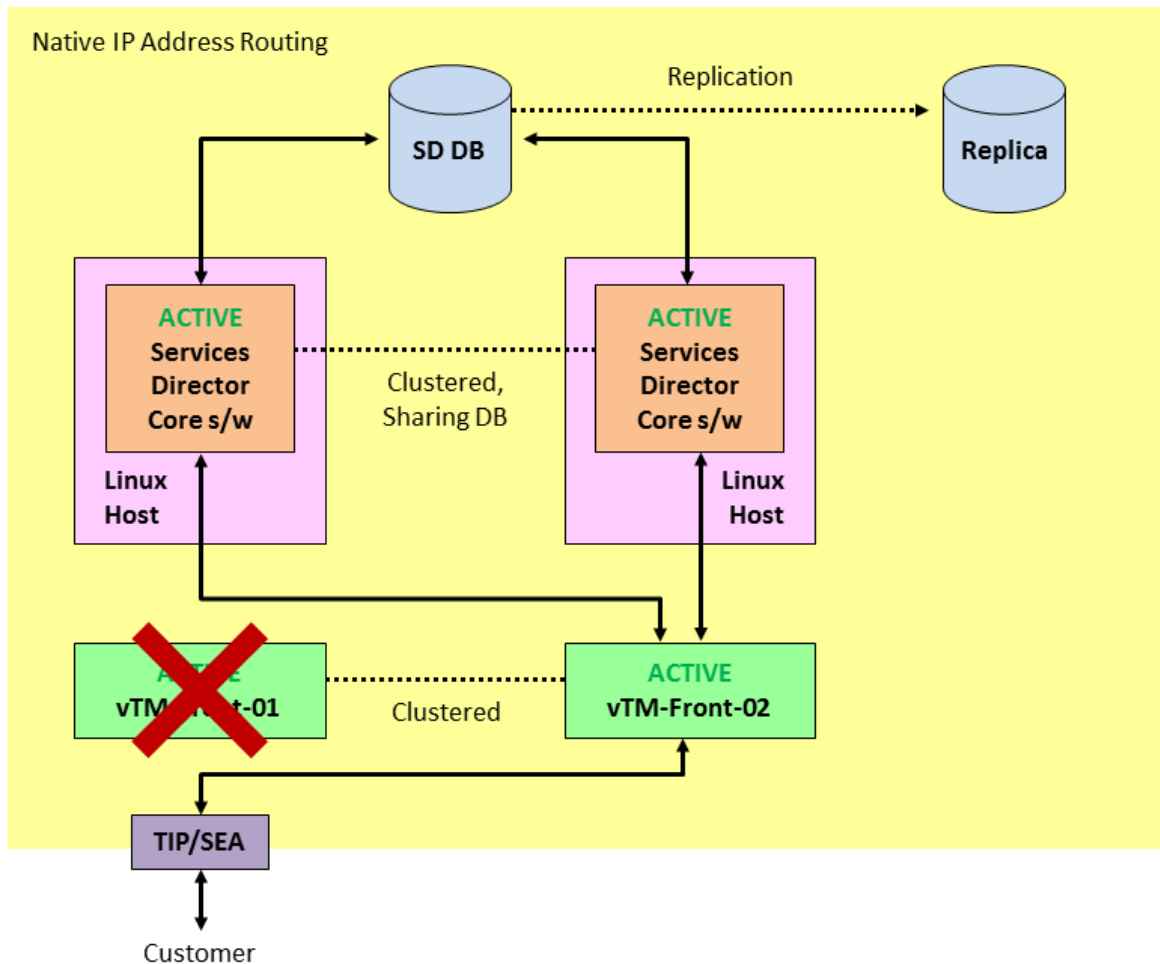
In the event of a failure of the main Services Director database, manually switch the Services Director configuration to use the replica database.

> **NOTE**
> Depending on database activity and the replication schedule, the replica may be behind the main database; re-apply any lost configuration changes.

When the main database becomes available again (or you have replaced it with a new database based on the replica), manually switch the Services Director configuration back, and re-apply any lost configuration changes.

Populating the Estate of the Services Directors

Once your high availability installation is complete, you can start adding vTMs to the estate of the Services Directors. That is:

- Externally-deployed vTMs, see Registering Externally-Deployed Traffic Managers on page 117.
- vTMs that are deployed using an instance host, see Using an Instance Host Machine with a Software Services Director on page 47.
- Cloud-based vTMs, such as those deployed on AWS.

NOTE
Cloud-based vTMs are not supported when the Services Director is in a private network behind a NAT device.

For example (using generalisations to represent your chosen Services Director configuration):

FIGURE 68 The Estate of a Services Director Installation
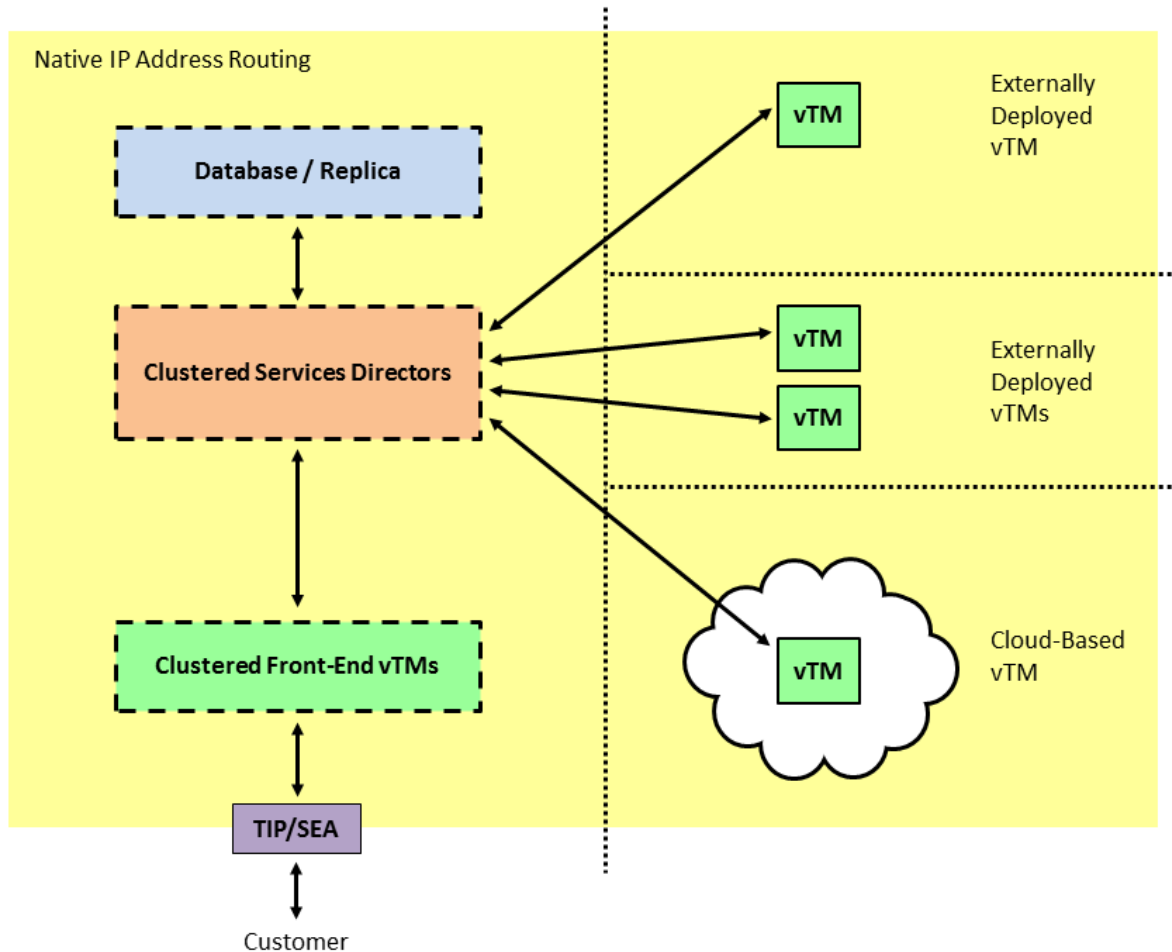
You create the required resources using manually-issued REST API requests to the SEA of the Services Directors.

You can use any combination of virtual machines, physical machines and data centers to achieve this. Physical diversity of elements (for example, spreading elements across data centers) is likely to increase the resilience of the installation.

NOTE
The Services Director and its front-end vTMs can be in a private network behind a NAT device. However, the vTMs that will be added to the estate of the Services Director at a later stage cannot be behind a NAT.

Technologies allowing routable connections between native IP addresses in disparate networks (for example, Virtual Private Networks) are supported.

Disaster Recovery for the Software Services Director

The software form-factor Services Director does not support:

- Scheduled backups of the Services Director configuration.
- Scheduled backups of individual vTM configurations.

If your Services Director installation is lost, you will require the MySQL database (or its replica) to recover the installation. This will enable the Services Director's configuration, including records of all vTMs in its estate, to be recovered.

> **NOTE**
> The master password for the Services Directors is required during any recovery from failure.

The Services Director MySQL database includes:

- Software licenses for the software form-factor Services Directors.
- All information about the vTMs in the estate of the Services Directors, including administration credentials.
- All information for any instance hosts known to the Services Directors.
- Universal FLA licenses for vTMs in your estate.

The Services Director MySQL database does *not* include:

- Any information about the front-end vTMs; you will need to deploy and configure these manually.
- Any Legacy FLA licenses for older vTMs in your estate.
- Any vTM images that you had loaded to enable vTM deployments using an instance host.

For example, if all of your Services Directors are lost:

- Recreate all elements (and clustering) of your original installation except for the MySQL database.
- Configure the Services Directors to use the existing MySQL database (or its replica). It is recommended that you establish a database replication regime for your chosen database.
- Re-install any Legacy FLA licenses as required.
- Configure the front-end vTMs to match your original installation.
- (Optional) Re-install any instance host and vTM images as required. You must do this before further vTMs can be deployed.

The recovered installation will be broadly similar to your original, but may require specific minor configuration.

Deploying High Availability for the Services Director VA

To create a high availability installation of the Services Director VA, you must:

- Install a Primary and a Secondary Services Director VA. See the *Brocade Services Director Getting Started Guide*.

> **NOTE**
> The Services Director VAs will automatically act as an Active/Standby HA pair, and automatically create the required MySQL database.

> **NOTE**
> You must ensure that the master password for the Services Directors is recorded securely. This is required during any recovery from failure, see Disaster Recovery for the Software Services Director on page 215.

- Specify a Service Endpoint Address (SEA) for the HA pair. The Services Director VA is always accessed by its SEA.

  NOTE
  The SEA for the Services Director HA pair can be in a private network behind a NAT device. However, the vTMs that will be added to the estate of the Services Director at a later stage cannot be behind a NAT.

- (Optional) Add one or more Legacy FLA licenses if required.

  NOTE
  Universal FLA licenses are included in the Services Director VA product.

- (Optional) Install an instance host if required.

Your Services Director VAs can be deployed as required within your network. Each element of your Services Director installation must be able to route to each other element using native IP address routing only. The use of Network Address Translation (NAT) between the Services Director and other elements is not supported; technologies allowing routable connections between native IPs in disparate networks (for example, Virtual Private Networks) may be used.

If your Active Services Director VA fails, you must manually failover the Standby Services Director VA so that it becomes Active. The customer continues to access the Services Director VA using its SEA. Once you have performed all repairs, you can revert to the original Services Director VA. See the *Brocade Services Director Getting Started Guide*.

After you have completed your high availability installation of the Services Director VAs, you can add Traffic Managers to the estate of your Services Directors. As with the Services Director VA installation, physical and virtual diversity is likely to increase the resilience of your estate. For example:

FIGURE 69 The Estate of a Services Director VA Installation
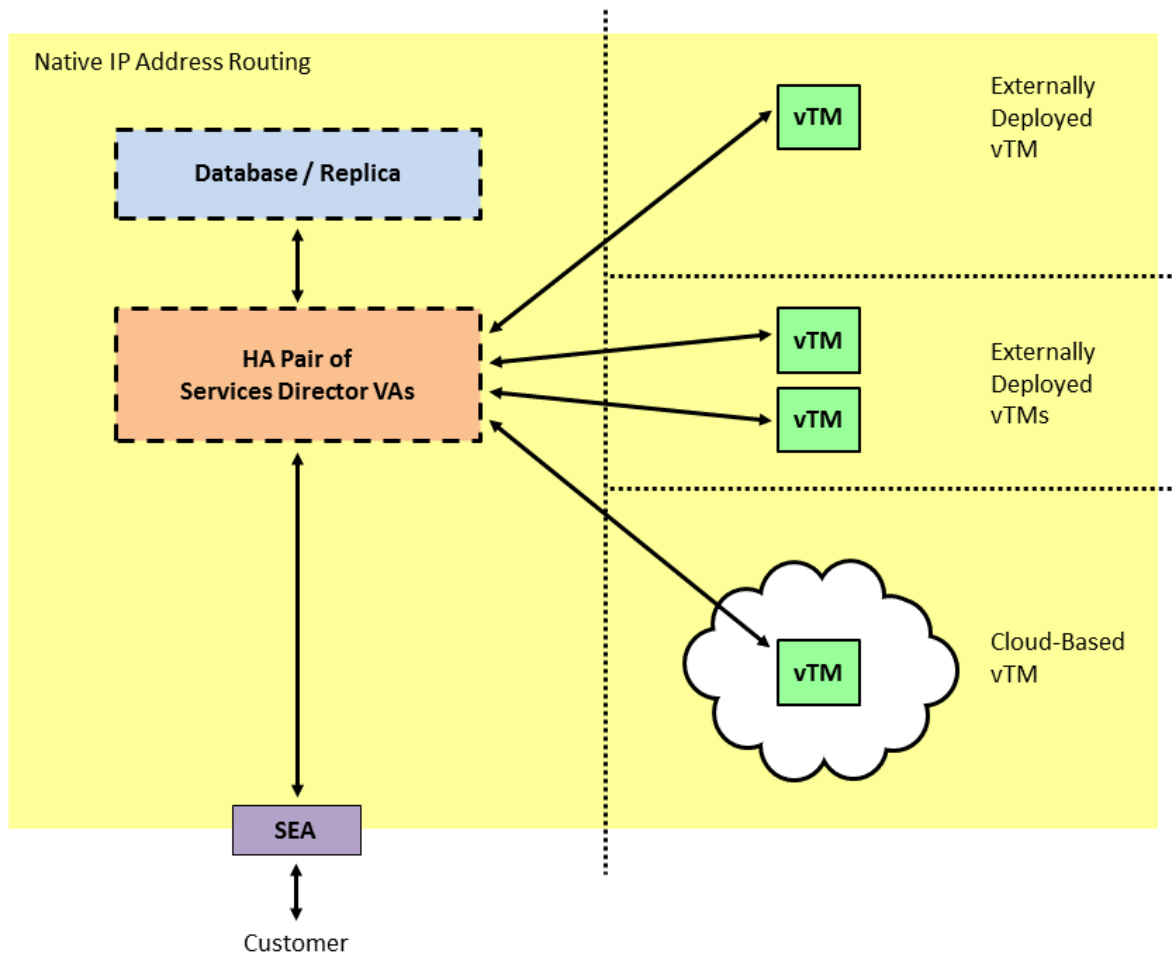
You can create backups for various configurations from the Services Director VA:

- The configuration of the Services Director VA can be backed up automatically to a local or remote location according to a user-defined schedule.
- The configuration of individual vTM clusters can also be backed up automatically according to a user-defined schedule.

You can use these backups to recreate a Services Director VA installation in the event of a failure.

For example, if your Services Director installation is lost:

- Create a new Primary Services Director VA, and choose to create it from a backed up configuration.
- Create a new Secondary Services Director VA, connecting it to the recovered Primary.

> **NOTE**
> Do *not* create this Services Director from a backup; the configuration is automatically retrieved from the Primary Services Director VA.

- Recreate any configuration that is not included in the restore. For example, re-load any required vTM images that were previously used for deployments using an instance host.
- (Optional) Reload any Legacy FLA licenses if required.

- View the vTMs in the estate of the Services Director VA, and ensure they are licensed correctly.

- Recover the configuration of the vTM clusters from their backups.

  NOTE
  The master password for the Services Directors is required during any recovery from failure.

The recovered installation will be broadly similar to your original, but may require specific minor configuration.

See the *Brocade Services Director Getting Started Guide* for details of all processes listed above.

Appendix: Managing the Services Director Using the CLI

Starting the CLI

After you have created the virtual appliance in vSphere, you can administer and manage the Services Director using the CLI or GUI. This chapter describes how to perform configuration tasks using the CLI only.

> **NOTE**
> For the purposes of this chapter, it is assumed that the hostname of your Services Director is **amnesiac** , and that a DNS server is in place.

To Log in to the CLI

1. Open the Services Director in a Telnet or SSH client program such as PuTTy.

2. Log into the Services Director as an administrator:

   ```
   login as: admin
   Brocade Services Director
   admin@amnesiac:<password specified in the graphical Setup Wizard>
   Last login: Tue Aug  4 10:09:03 2015 from <IP-address>
   amnesiac >
   ```

3. Start configuration mode:

   ```
   amnesiac > enable
   amnesiac # configure terminal
   amnesiac (config) #
   ```

   You can now enter CLI commands.

Importing the SSL Certificate, Key, and Licenses

If you did not complete licensing using the graphical Setup Wizard (see the *Brocade Services Director Getting Started Guide*), you must import the following files into the Services Director before you can create instances:

- SSL certificate and key
- Services Director license
- Bandwidth license key
- Legacy FLA license (if you are not using the pre-installed Universal License)
- Traffic Manager images

   NOTE
   If you have not received your license files, contact Brocade Licensing for assistance.

To Import the SSL Certificate Key, Services Director License, and Bandwidth License

1. Log into the Services Director and start the CLI. See Starting the CLI on page 219.

2. To import an SSL certificate and key, you must provide the file path to the certificate and key file. For example, an http, ftp, or scp URL (scp://username:password@host/path).

   ```
   amnesiac (config) # ssc import-cert-key scp://username:pwd@sfhost.example.com/sd_archive/cert_key.pem
   Certificate and Private Key imported successfully
   amnesiac (config) # show ssc certificate
   >>certificate and key is displayed
   ```

3. To import a Services Director (Enterprise) license, you must provide the file path to the license file. For example, an http, ftp, or scp URL (scp://username:password@host/path).

   ```
   amnesiac (config) # ssc import-lic file scp://username:pwd@sfhost.example.com/sd_archive/ent-license
   License imported successfully
   amnesiac (config) # show ssc license-file
   XXX-XXXXXXX-XXXX-X-XXXX-XXXX-XXXX
   ```

4. Import the enterprise bandwidth license key into the Services Director. To do this, provide the license key you obtained from your account representative.

   ```
   amnesiac (config) # ssc license enterprise throughput add XXX-XXXXXX-XXXX-X-XXXX-XXXX-XXXX
   ```

To Import a Legacy FLA License

If you intend to use Traffic Managers whose version is below 10.1, or Traffic Managers for which the REST API is disabled, you must now install a Legacy FLA license, and create a **license** resource for it inside the Services Director.

1. To import a Legacy FLA License (for example **fla-ssl-ssc**), you must provide the file path. For example, an http, ftp, or scp URL (`scp://username:password@host/path`).

   ```
   amnesiac (config) # ssc stm import-lic file scp://username:pwd@sfhost.example.com/sd_archive/fla-ssl-
   ssc
   License imported successfully
   amnesiac (config) # show ssc stm license-file
   >>the license file is displayed
   ```

2. To create a **license** resource:

```
amnesiac (config) # ssc license create license-name fla-ssl-ssc
+--------+--------+
| Field  | Value  |
+--------+--------+
| info   | Active |
| status | Active |
+--------+--------+
```

3. You can confirm that the Services Director process is running:

```
amnesiac (config) # show ssc service
SSC service status: running
```

To Import a Traffic Manager Image

1. You must import the Traffic Manager image (that is, the tarball) and create a **version** resource for the software image. (For example, **stm101**.) To import an image, you must provide the file path. For example, an http, ftp, or scp URL (`scp://username:password@host/path`).

```
amnesiac (config) # ssc stm import-image file scp://root@sf.test.com/sd_archive/ZeusTM_101_Linux-
x86_64.tgz
amnesiac (config) # show ssc stm images
Imported Brocade Traffic Manager Images
---------------------------------------
ZeusTM_101_Linux-x86_64.tgz
```

2. To create a **version** resource:

```
amnesiac (config) # ssc version create version-name stm101 vfilename ZeusTM_101_Linux-x86_64.tgz
vdirectory ZeusTM_101_Linux-x86_64.tgz
+-------------------+-----------------------------+
| Field             | Value                       |
+-------------------+-----------------------------+
| info              | Active                      |
| status            | Active                      |
| version_filename  | ZeusTM_101_Linux-x86_64.tgz |
| version_directory | None                        |
+-------------------+-----------------------------+
```

Where:

- – **version-name** is a unique name for the Traffic Manager image.
  - **vfilename** is the name of the Traffic Manager image.
  - **vdirectory** is the name of directory to which tarball extracts; if none, specify the tarball name.

    NOTE
    To delete an image, run the command: **no ssc stm image-file <image name>**.

Importing an Instance Host OVA into the Services Director

You import the Instance Host OVA and create an SSH key for the administrator user. This enables the automatic creation of virtual instance host on ESXi.

- Import the Instance Host OVA into the Services Director:

```
amnesiac (config) # ssc import-host-ova file scp://root@sf.test.com/sd/rvbd-ssc-host.ova
Host OVA imported successfully.
amnesiac (config) # show ssc host-ova
Imported Host OVA
-------------------------------------
Rvbd-host.ova
```

Enabling Passwordless SSH Communication

You must create a public SSH key to enable passwordless communication between the Services Director and the instance host. This SSH key will be used for all deployed instance hosts.

1. In a terminal session for the Primary Services Director, create an SSH public key for the administrator user to perform passwordless communication to the instance host.

```
amnesiac (config) # show ssh client private
No user identities configured.
SSH authorized keys:
amnesiac (config) # ssh client generate identity user admin
amnesiac (config) # show ssh client private
User Identities:
    User admin:
>>ssh public and private keys are displayed
```

2. In a terminal session for the instance host (user **sscadmin**), inject the SSH public key for the Services Director's **admin** user into the instance host:

```
$ user root sshkey "ssh-rsa <public_key> admin"
Added public SSH key for user (root)
```

3. Repeat steps 1) and 2) for the Secondary Services Director.

Creating a Feature Pack for Instances

You must create a feature pack in the Services Director before you can create instances. A feature pack describes a set of licensable features that you can apply to a Traffic Manager instance. A feature pack is the same as a SKU or a subset of features in a SKU.

When you deploy or modify a Traffic Manager instance, the feature pack controls what licensable features are allowed (but does not specify bandwidth limits). Creating a feature pack in the Services Director requires you to base the pack on a SKU and to give it a unique name.

```
amnesiac (config) # ssc feature-pack create fpname default-fp stm-sku STM-400
+---------- ----------------+
| Field     | Value         |
+------ --------------------+
| info      | Active        |
| status    | Active        |
| stm_sku   | STM-400       |
```

```
| excluded  | None         |
+-----------+--------------+
```

Syntax: **ssc feature-pack create fpname <resource-unique-name> stm-sku <SKU-for-feature-pack>**

Deploying an Instance Host

After you have created a default feature pack, you can deploy the instance host using the CLI.

When you run the **ssc host provision** command, it untars the OVA, repackages the SSH public key created earlier, and provisions it on ESXi. After the instance host is created on ESXi, the public key is assigned to the user that is defined in the **ssc host provision** command (in this example, **root**). After this command is run, you can connect to the instance host without a password using SSH.

> **NOTE**
> Although the Services Director VA runs on vSphere ESXi 5.0+, the **ssc host provision** command works only on ESXi 5.0 and 5.1, not ESXi 5.5.

1. To deploy an instance host, specify a name and size (large or small), and provide the ESXi username, password, location, and datastore location. You also specify the instance host username.

   ```
   amnesiac (config) # ssc host provision name test-demo flavor small esx-uri esx://sf.test.com esx-
   user admin esx-pass brocade storage datastore1 host-user root
   ```

   Syntax: **ssc host provision name <unique-name> flavor [small | large] esx-uri <URI-or-path-to-esxi-host> esx-user <esxi-username> esx-pass <esxi-password> storage <esxi-datastore-name> host-user <host-username> <cr> | [host-info <instance-host-info>]**

Parameter	Description
name <unique-name>	Specify the hostname of this instance host. This is required for all instance hosts. **NOTE** For non-provisioned hosts, this must be the hostname of the existing host.
flavor [small | large]	Specify the size of the instance host: • small - Specifies two CPUs and 4 GB of memory • large - Specifies 8 CPUs and 16 GB of memory
esx-uri <URI-or-path-to-esxi-host>	Specify the URI to the ESXi host where this instance host is to be deployed.
esx-user <esxi-username>	Specify the ESXi host username used for the provisioned instance host.
esx-pass <esxi-password>	Specify the ESXi password used for the provisioned instance host.
storage <esxi-datastore-name>	Specify the ESXi storage pool. For example, **datastore1**.
host-user <host-username>	Specify the name of the user that manages the instance host.
host-info <instance-host-info>	Optionally, specify descriptive information for the new instance host.

2. To test whether the instance host has been provisioned in vSphere, open vSphere, right-click the host, and select **Open Console** to display the new instance host command line in a console window.

3. Log in to the new instance host. By default, the username is **sscadmin** and the password is **password**.

    ```
    $amnesiac login: sscadmin
    Password: password
    $show host interface eth0
    >>interface details are displayed with the new <IP address> <<
    ```

4. On the Services Director, **ping** the new instance host.

    ```
    amnesiac (config) # ping <IP_address>
    64 bytes from <IP_address>: icmp_seq=4 ttl-64 time=0.761 ms
    64 bytes from <IP_address>: icmp_seq=4 ttl-64 time=0.761 ms
    ```

5. If the ping is successful, you can log in to the new instance host as the **sscadmin** user and change its management (eth0) IP address. First, log into the new host.

    ```
    amnesiac (config) # slogin sscadmin@<IP_address>
    ```

    Display host information:

    ```
    $ show host info
    ```

    Then, change the name of the host if required:

    ```
    $ host hostname <new_hostname> domain <domain_name>
    ```

    Finally, change the management (eth0) IP address, netmask and gateway.

    ```
    $ host interface eth0 ip <ip-addr> netmask <netmask> gateway <ip-gateway>
    ```

    Your terminal session will be lost at this point.

6. Log into a terminal session for the instance host using its new management (eth0) IP address.

7. Change the WAN (eth1) and LAN (eth2) IP addresses.

    ```
    $ host interface eth1 ip <ip-addr> netmask <netmask>
    $ host interface eth2 ip <ip-addr> netmask <netmask>
    ```

    After the instance host has been provisioned in vSphere, you must create the instance host entry for deployed instances on the Services Director by assigning a name, and user.

To Configure DNS on the Instance Host

Check the DNS settings on your instance host, and set up if required.

1. In an instance host session (slogin from Services Director or direct login), check DNS:

    ```
    $ show host dns
    +-----------------+-------+
    | Field           | Value |
    +-----------------+-------+
    | nameservers     |       |
    | dhcp_nameservers |      |
    | dns_search      |       |
    +-----------------+-------+
    ```

    In this example, DNS is not defined.

2. Define DNS if required:

    ```
    $ host dns dns-nameservers <IP addresses> dns-search <domain>
    ```

To Enable Passwordless SSH Communication

You must create a public SSH key to enable passwordless communication between the Services Director and instance hosts. This SSH key will be used for all deployed instance hosts.

1. From a Services Director terminal session, create an SSH public key for the administrator user to perform passwordless communication to the instance host.

```
amnesiac (config) # show ssh client private
No user identities configured.
SSH authorized keys:
amnesiac (config) # ssh client generate identity user admin
amnesiac (config) # show ssh client private
User Identities:
     User admin:
>>ssh public and private keys are displayed
```

2. Retrieve the IP address of the Services Director Instance Host and login via SSH as user **sscadmin**.

3. Inject the SSH public key of the admin user of Services Director VA into the instance host:

```
$ user root sshkey "ssh-rsa <public_key> admin"
Added public SSH key for user (root)
$ <ctrl-d>
```

To Create the Instance Host on the Services Director Using the CLI

1. At the prompt, enter:

```
amnesiac (config) # ssc host create host-name <host-FQDN> host-user
sscadmin host-pass password username root
Created host session for (host-FQDN) with user (sscadmin)
+-------------------+----------------+
| Field             | Value          |
+-------------------+----------------+
| install_root      | /root/install  |
| username          | root           |
| retained_info_dir | None           |
| status_check      | None           |
| hw_slot_number    | None           |
| size              | None           |
| info              |                |
| work_location     | /var/cache/ssc |
| hw__node          | None           |
| cpu_cores         | None           |
| status            | Active         |
| usage_info        |                |
+-------------------+----------------+
```

2. Confirm the deployment of the instance host:

```
amnesiac (config) # ssc host list
+--------------+
| Host         |
+--------------+
| <host-FQDN>  |
+--------------+#
```

Once the instance host is deployed, it appears as a configured instance host on the **Instance Hosts** page of the Services Director VA.

Working with User Authentication for a vTM

You can apply user authentication to a vTM from the Services Director VA CLI in two ways:

- During the self-registration of a vTM, see Working with User Authentication for a vTM on page 226 on page 208.
- After registration of the vTM, see Working with User Authentication for a vTM on page 227.

Both of these methods require the user authentication resources to already be configured.

To Define User Authentication Using the CLI

1. Create the required **authenticator** resource using the following CLI commands:

   - **ssc authenticator create ldap**

   - **ssc authenticator create radius**

   - **ssc authenticator create tacacs_plus**

   For a detailed description of **authenticator** resource properties, see authenticator Resource on page 129.

2. Create any required **permission_group** resources using the following CLI command:

   ssc permission-group create

   For a detailed description of **permission_group** resource properties, see permission_group Resource on page 164.

3. Create an **access_profile** resource, combining an authenticator with the required permission groups. To do this, use the following CLI commands:

   - **ssc access-profile create**

   - **ssc access-profile add-perm-group**

   For a detailed description of **access_profile** resource properties, see access_profile Resource on page 126.

   See Brocade Services Director Command-Line Interface Reference Manual for details of all commands.

Working with User Authentication for a vTM

Self-registration is described in the *Brocade Services Director Getting Started Guide*.

Self-registration requests are generated by the vTM itself, and cannot be created using the CLI.

To change the state of a self-registration request, use the following CLI commands:

- **ssc registration update registration-id <reg-id> state decline reason <reason>**
- **ssc registration update registration-id <reg-id> state blacklist**
- **ssc registration update registration-id <reg-id> state pending**

However, when you transition the request to Accepted, in addition to several extra mandatory parameters, you can also specify the required access profile:

- **ssc registration update registration-id <reg-id> state accept instance-name <name> owner <owner> feature-pack <feature_pack> bandwidth <bandwidth> access_profile <access_profile>**

See the *Brocade Services Director Command-Line Interface Reference Manual* for details of all commands.

For a detailed description of all registration resource properties, see registration Resource on page 169.

Working with User Authentication for a vTM

To apply user authentication to a registered vTM, update the instance resource to include the **access_profile** property:

```
ssc instance update instance-name <instance_id> access-profile <access_profile>
```

When you do this, the authenticator and permission groups in the access profile are applied to the vTM. Existing authenticators and permission groups may be overwritten, but none will be deleted. All members of a cluster are affected.

Working with Backup Schedules and Cluster Backups

A Traffic Manager cluster gathers Virtual Traffic Managers (vTMs) together and operates them under a shared cluster configuration.

The configuration of the cluster can be backed up automatically on a regular basis according to a backup schedule.

The following provides an overview of automatic cluster backup operations.

FIGURE 70 Overview of Cluster Backups
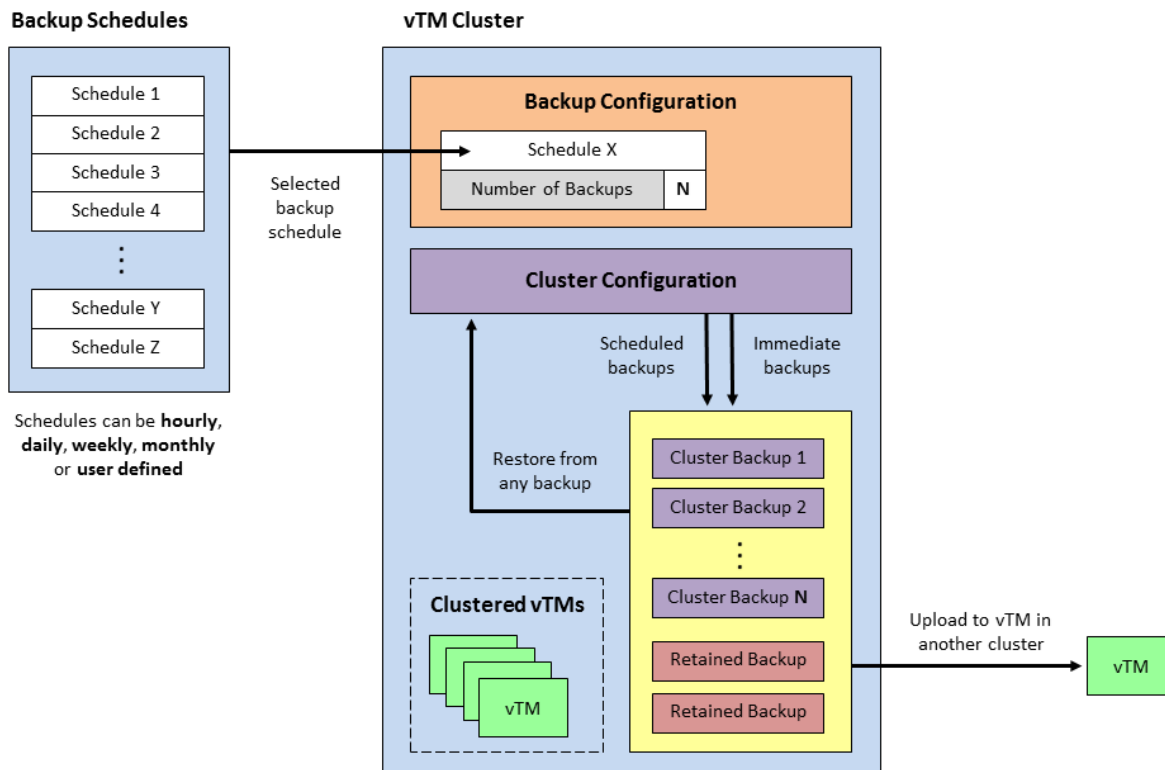
See the *Brocade Services Director Getting Started Guide* for a full description of the above operations.

You can view clusters with the following commands:

- **ssc cluster list** – displays a list of all clusters names.
- **show ssc cluster cluster-name** – displays full details for a specified cluster.

You can create, update and view cluster backup schedules with the following commands:

- **ssc backup-restore cluster create schedule** – creates a new cluster backup schedule.

- **ssc backup-restore cluster update schedule** – updates a specified cluster backup schedule.
- **show ssc backup-restore cluster schedules** – displays a list of all cluster backup schedule names.
- **show ssc backup-restore cluster schedule** – displays a specified cluster backup schedule.

You can associate a cluster with a cluster backup schedule with the following commands:

- **ssc cluster create cluster-name schedule** – creates a new cluster that is associated with the specified cluster backup schedule.
- **ssc cluster update cluster-name schedule** – updates a specified cluster to associate it with the specified cluster backup schedule.

You can view cluster backups with the following commands:

- **show ssc backup-restore cluster cluster-name backups** – displays a list of all backup names for a specified cluster.
- **show ssc backup-restore cluster cluster-name backup-name** – displays a specified backup.

You can perform manual cluster backup operations with the following commands:

- **ssc backup-restore cluster cluster-name backup now** – this requests an immediate manual backup for a cluster.
- **ssc backup-restore cluster cluster-name restore backup-name** – this requests a restore to a specified cluster of a specified backup.
- **ssc backup-restore cluster cluster-name upload backup-name** – this requests an upload of a specified backup to a specified Traffic Manager instance.

Each of these commands creates a backup task. You can view tasks and re-attempt failed tasks with the following commands:

- **show ssc backup-restore cluster cluster-name tasks** – displays a list of all task IDs for a specified cluster.
- **show ssc backup-restore cluster cluster-name task** – displays a specified task.
- **ssc backup-restore cluster cluster-name task retry** – re-attempts a failed specified task.

ESXi vSphere Host Port Mapping

You must map the three interfaces of the instance host VA to the corresponding host ports where the instance host VA is deployed. You map the ports and network using the vSphere client.

> **NOTE**
> Promiscuous mode in the virtual switch must be enabled. For details about configuring promiscuous mode, see the VMware documentation.

You must first verify the mapping of the network adapters to their names given by the kernel. For example, network adapter 1 to eth0, network adapter 2 to eth1, and network adapter 3 to eth2.

You map the ports by starting the instance host VA, checking the MAC address of each interface, and matching that to the MAC address given to each network adapter by the ESXi host. Then you must map each of these network adapters to the correct network on the ESXi host. Brocade assumes that eth0 is always the management network, eth1 is the WAN network, and eth2 is the LAN network. With the instance host VA powered off, on the vSphere client you choose Settings on the instance host VA, select each network adapter, and select the host **Network Connection** > **Network Label** for the appropriate mapping.

The following example represents port mapping on the ESXi vSphere client:

The instance host VA has the following mapping when starting and matching eth0, eth1 and eth2 virtual network cards to the ones given by the ESXi host:

```
network adapter 1 <-> eth0 (mgmtbr0)
network adapter 2 <-> eth1 (wanbr0)
network adapter 3 <-> eth2 (lanbr0)
```

On the Settings panel of the instance host VA, select each network adapter and change the network label to the appropriate network on the ESXi host.

This example assumes that the ESXi host has the following networks: mgmt, wan, and lan:

```
network adapter 1 <-> 'Network label: mgmt'
network adapter 2 <-> 'Network label: wan'
network adapter 3 <-> 'Network label: lan'
```

Exporting a Database

To export the MySQL inventory database from the CLI:

```
ssc database local db-file export
```

The name of the exported database file is chosen automatically, using the following format:

sscdb_dump_<VA_version>_<timestamp>.sql

Generating a Self-Signed SSL Server Certificate

The Services Director is commonly deployed using self-signed certificate/key pairs, using the self-signed server certificate in the Legacy FLA License. Brocade recommends that you do not use a CA-signed certificate.

The Setup Wizard enables you to generate a self-signed certificate. However, you can choose to generate a self-signed SSL certificate before starting the Setup Wizard.

To Generate a Self-Signed SSL Server Certificate

At the Linux prompt, enter:

```
openssl req -x509 -nodes -newkey rsa:2048 -keyout key.pem -out cert.pem -days 3650
```

Parameter	Description
req	Specifies an X509 certificate signing request management.
-x509	Specifies a self-signed certificate rather than a certificate request.
-nodes	Specifies that the private key will not be encrypted (otherwise, the server needs a password to start).
-newkey rsa:2048	Generates a new certificate request and sets the key size.
-keyout key.pem	Sets the target for the new private key.
-out cert.pem	Sets the target for the certificate.
-days 3650	Specifies the duration of the certificate (default is 30 days). A longer period may be desirable as a fresh FLA license will need to be generated and then deployed to all STM instances when the certificate expires.

The FLA license does not accept composite certificates that include a server certificate along with other information or certificates created by **ssh-keygen**.

To Verify the SSL Certificate

1. At the Linux prompt, enter:

    ```
    openssl x509 -in certificate.crt -noout
    ```

    This command either succeeds silently for a valid certificate, or reports errors.

2. To verify a signed certificate, enter:

    ```
    openssl verify <certificate name>
    ```

Generating Metering Logs

To extract metering logs using the Services Director VA CLI, use the following command:

```
amnesiac (config) # ssc log metering generate [backup [yes|no]]
```

In this example, the backup switch indicates whether to regenerate all logs up to the most recent log generation. Any new logs since the most recent log generation will always be included.

A maximum of ten metering logs can be generated by this process.

> **NOTE**
> To generate metering logs using the Services Director VA. See the *Brocade Services Director Getting Started Guide*

Appendix: Email Notifications Generated By Services Director

Notifications/Alerts from the Services Director Core Software

The following notifications are generated from the core Services Director software. They are common to both the software-only and VA form factors of Services Director.

NOTE

Only email subject lines are shown. Email bodies will normally contain further detail related to the reason for the notification/alert.

Functionality	Area Email Subject Line	Event Description
Licensing	Brocade Services Director: License is due to expire on <Services Director host>	Sent on a daily basis when a Services Director controller licence is due to expire within the licence expiry warning period (this warning period is configurable, but defaults to 30 days).
	Brocade Services Director: Required bandwidth/add-on pack is due to expire on <Services Director host>	Sent on a daily basis when a Services Director bandwidth pack licence is due to expire within the bandwidth licence expiry warning period (this warning period is configurable, but defaults to 30 days).
Monitoring	Brocade Services Director Monitoring has detected <failure / recovery events>	Sent as a result of monitoring detecting that connectivity to vTM instances or Services Director controllers have either failed or recovered after a failure.
vTM Self-Registration	Instance <Instance name> has registered itself and requires your attention.	Sent when a self-registering instance has sent a registration request that the Services Director Administrator may wish to manually accept.
	Instance <instance name> has re-registered itself and requires your attention.	Sent when a self-registering instance has been declined but has attempted to self-register again with different parameters.
	Instance <instance name> has registered itself and was auto-accepted by policy <policy name>	Sent when a self-registering instance has been received by Services Director, and automatically accepted by an auto-acceptance policy.
	Instance <instance name> registered itself and failed to be auto-accepted by <policy name>	Sent when a self-registering instance has been received by Services Director, but for some reason failed to be automatically accepted by an auto-acceptance policy.
Metering	Services Director: Metering Warnings have been found for instances on <Services Director host>	Sent periodically if potential metering discrepancies are identified.
	Brocade Services Director: Log space low on <Services Director host>	Sent periodically if available log space has dropped below 512 MB (the period is configurable, but set to one hour by default).

Functionality	Area Email Subject Line	Event Description
	Brocade Services Director: Log space critical on <Services Director host>	Sent periodically if available log space has dropped below 256 MB (the period is configurable, but set to one hour by default).
	Metering phone-home successful	Sent to confirm that Services Director has successfully sent a set of metering logs back to Brocade.
	Metering phone-home failure	Sent if Services Director has been unable to send a set of metering logs back to Brocade.

Notifications/Alerts from the Services Director VA

The following notifications are sent from the Services Director VA. These are in addition to those sent from the Core software.

> **NOTE**
> Only email subject lines are shown. Email bodies will normally contain further detail related to the reason for the notification/alert.

Functionality	Area Email Subject Line	Event Description
Services Director service	Local Services Director Failure Notification	Local Services Director Process Failure (Killed or Crashed)
	Local Services Director Recovery Notification	Local Services Director Process Recovery
Database	Local Database Failure Notification	Local DB Failure
	Local Database Recovery Notification	Local DB Recovery
High Availability	Peer Services Director Connectivity Failure Notification	Peer Services Director Failure
	Peer Services Director Connectivity Recovery Notification	Peer Services Director Recovery
	Local File System Replication Failure Notification	Local Gluster Failure
	Local File System Replication Recovery Notification	Local Gluster Recovery
Disaster Recovery	Backup Generate Operation Failure Notification	Services Director Backup Generation Failure
	Backup Generate Operation Recovery Notification	Services Director Backup Generation Recovery
	Backup Transfer Operation Failure Notification	Services Director Backup Transfer Failure
	Backup Transfer Operation Recovery Notification	Services Director Backup Transfer Recovery
Master Password	Master password Operation Failure Notification	Master Password Missing
	Master password Operation Recovery Notification	Master Password Recovery
System	Core file found for process: <Process Name>	Crash Core Generated
	CPU utilization too high	High CPU Usage
	CPU utilization alarm clearing	High CPU Usage Recovery
	Paging activity too high	Paging Alarm
	Paging activity alarm clearing	Paging Alarm Cleared
	File System %s is full	File Systems Mount Failures
	File System %s is no longer full	File System Mount Recovery

Functionality	Area Email Subject Line	Event Description
	Linkstate alarm triggered	Link State
	Linkstate alarm cleared	Link State Cleared
	Link duplex alarm triggered	Link Duplex
	Link duplex alarm cleared	Link Duplex Clear
	Link I/O errors alarm triggered	Link IO Errors
	Link I/O errors alarm cleared	Link IO Errors Recovery
	Memory Error Detected	Memory Error
	Memory error alarm cleared	Memory Error Cleared
	Unexpected shut down	Unexpected Shutdown
	Secure vault must be unlocked	Secure Vault Locked
	Secure vault has been unlocked	Secure Vault Unlocked
	Process failure: %s	Process Crash
	Process exit: %s	Process Unexpected Exit