

VMware Software-Defined Data Center Load Balancer as a Service

TECHNICAL WHITE PAPER

MAY 2017

VERSION 1.0



Table of Contents

Load Balancer as a Service Overview	3
Load Balancer as a Service Design Use Cases	3
Provisioning a Single-Site Load Balancer	4
Workflow for Provisioning a Single-Site Load Balancer	4
Provisioning a Multi-Site Load Balancer	6
Workflow for Provisioning a Multi-Site Load Balancer	7
Modifying a Load Balancer	9
Workflow for Modifying a Load Balancer	9
De-provisioning a virtual machine	10
Workflow for De-Provisioning a virtual machine	11
Load Balancer as a Service High-Level Design	12
Load Balancer as a Service Network Design Using NSX	12
Load Balancer as a Service BIG-IP DNS Design	13
Load Balancer as a Service LTM Design	14
Load Balancer as a Service Implementation	15
Provision an LTM virtual server	15
De-Provision a virtual machine	16

Revision History

DATE	VERSION	DESCRIPTION
May 2017	1.0	Initial version

Load Balancer as a Service Overview

Organizations that are making a journey from a traditional datacenter approach to software-defined-datacenter are finding a need to automate load balancer services. VMware offers L4-L7 load balancer services through NSX Enterprise Services Gateway (ESG), which are sufficient for the needs of most applications. VMware also offers a Software Defined Datacenter (SDDC) suite of products to help organizations automate the provisioning of applications and related services. Using VMware SDDC products, such as vRealize Automation and vRealize Orchestrator, you can automate the load balancing services for complex application needs. VMware SDDC products can also be extended to provide automated third-party load balancer services for applications.

This paper describes automation of F5 load balancers using VMware vRealize Automation (vRA), vRealize Orchestrator (vRO), and NSX for vSphere. This paper provides integration among vRA, vRO, F5 Local Traffic Manager (LTM) for providing load balancer services locally in a data center, and F5 BIG-IP DNS (previously known as Global Traffic Manager or “GTM”) for load balancing across two data centers. The paper provides automation of F5 Big-IP DNS between a *Primary* data center and a *Secondary* data center, located in two separate physical locations.

Like NSX load balancers, F5 load balancers offer value added services such as SSL offloading. SSL offloading approach allows secure SSL communication from client browsers to load balancer and non-SSL communication from load balancer to servers in the pool. This paper describes the use of Certificate Authority issued SSL certificates with F5. This paper also describes the use of F5 Big-IP DNS (a.k.a. global load balancer) to provide failover to application services across two data centers. Furthermore, this paper describes the use of weighed round-robin algorithm from F5.

This paper uses VMware XaaS capabilities to build LBaaS. Once built and made available as a catalog item(s), administrators can use LBaaS to provision and de-provision F5 load balancers using user-friendly graphical user interface screens.

Load Balancer as a Service Design Use Cases

A Load Balancer as a Service (LBaaS) includes several steps, including deploying an F5 virtual server, creating a member pool, creating an SSL profile, and creating a virtual IP. This paper assumes that a virtual IP will be assigned from an external DNS server, i.e. Infoblox. This paper describes the design of the following LBaaS.

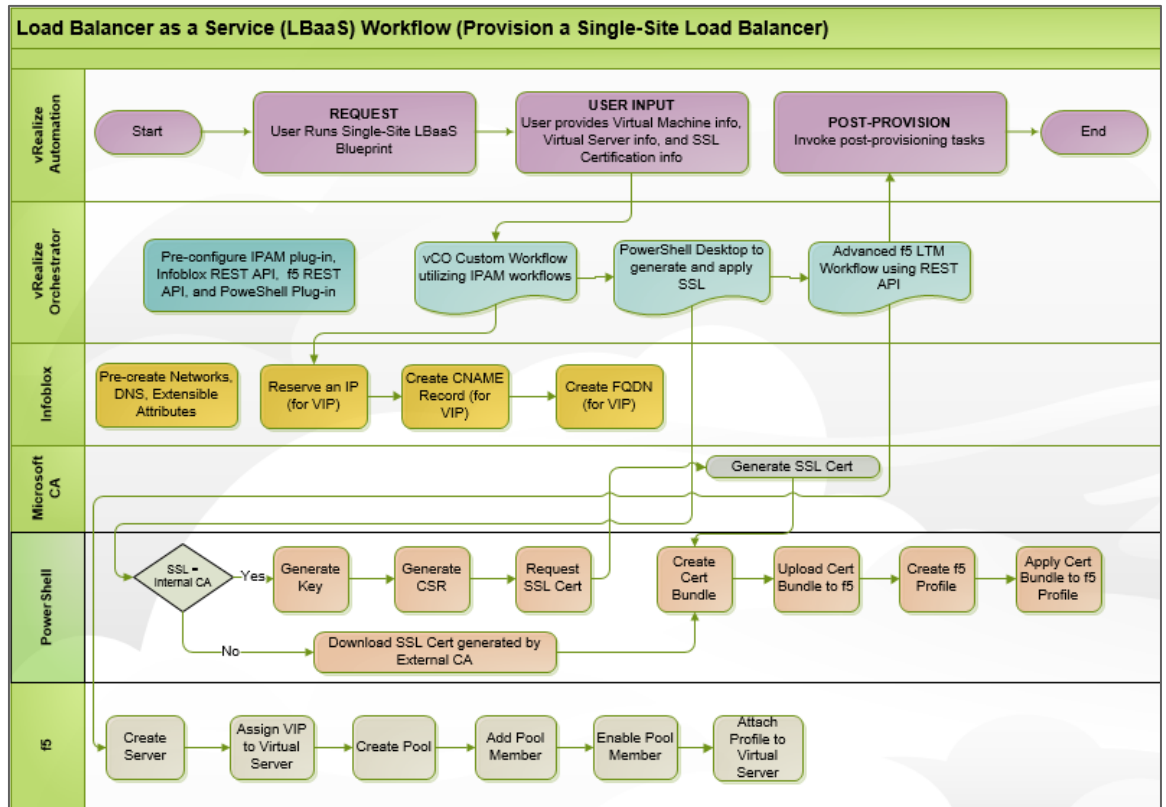
- 1) Provisioning a Single-Site Load Balancer
- 2) Provisioning a Multi-Site Load Balancer
- 3) Modifying a Load Balancer
- 4) De-provisioning a virtual machine

Provisioning a Single-Site Load Balancer



Provisioning a single-site load balancer can be achieved as a Day 2 operation, meaning the load balancer is provisioned after the deployment of virtual machines, which is a Day 1 operation. This automated approach can be used to deploy load balancer one site. vRealize Automation and vRealize Orchestrator are used to create the automation of load balancer deployment. Once deployed, the LBaaS is available as a catalog item from the vRealize Automation request screen. Administrators are prompted with various inputs, which are passed to F5 to create and deploy the load balancer.

Workflow for Provisioning a Single-Site Load Balancer

The following workflow diagram depicts various swim lanes and steps involved in the provisioning of a load balancer for a single site.

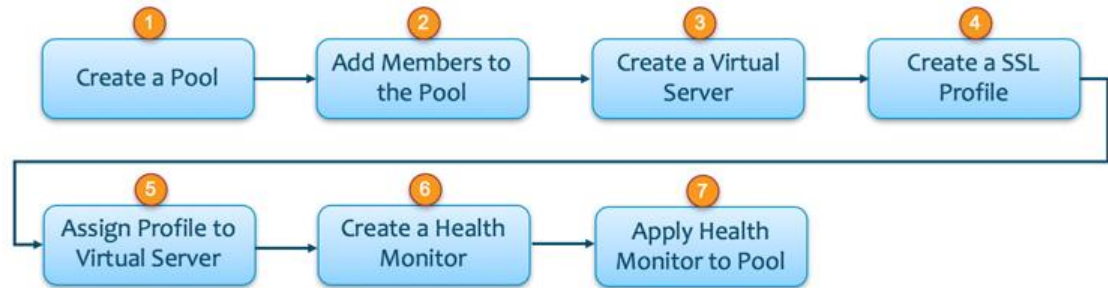


SWIM LANE	ACTION	NOTES
vRealize Automation	User requests XaaS workflow – Provision a virtual server	<ol style="list-style-type: none"> 1. Log in to the vRealize Self-Service Portal. 2. Execute the XaaS Provision a single-site virtual server blueprint.
vRealize Automation	User input	User provides input, including virtual machine info, F5 virtual server information, and SSL certificate information
vRealize Orchestrator	Pre-configure IPAM plug-in, Infoblox REST API, F5 REST API, and PowerShell plug-in	F5 REST plug-in can be utilized to build the vRealize Orchestrator workflows.
vRealize Orchestrator	vRO custom workflows utilizing IPAM workflows	Use of REST APIs for Infoblox significantly reduces rework of the workflows after product upgrades. Use of REST APIs eliminates the need for plug-in installation and configuration and reduces the version

SWIM LANE	ACTION	NOTES
		dependency. REST APIs are known to remain unchanged after major product releases.
vRealize Orchestrator	PowerShell Desktop to generate and apply SSL	<ol style="list-style-type: none"> 1. Open a PowerShell Desktop using a service account that is pre-approved to generate signed SSL certificates. 2. Run a script to generate Private Key. 3. Run a script to generate CSR. 4. Run a script to recreate SSL. Wildcard SSL can be utilized remove the process of generating a CSR.
vRealize Orchestrator	Advanced F5 LTM workflow using REST API	Execute a workflow to create an LTM virtual server using user-provided input and SSL certificate.
Infoblox	Pre-create networks, DNS, Extensible Attributes	 Infoblox DDI must be pre-configured.
Infoblox	Reserve an IP	When vRealize Orchestrator invokes a REST API to reserve an IP for the F5 virtual server, Infoblox performs the task and returns an IP address to vRealize Orchestrator.
Infoblox	Create a CNAME record for VIP	When vRealize Orchestrator invokes a REST API to create a CNAME record for the F5 virtual server VIP, Infoblox performs the task and returns a record to vRealize Orchestrator.
Infoblox	Create FQDN for VIP	When vRealize Orchestrator invokes a REST API to create a FQDN record for the F5 virtual server VIP, Infoblox performs the task and returns a record to vRealize Orchestrator.
Microsoft CA	Generate SSL cert	Microsoft CA is used for generating SSL Certificates for the F5 virtual server profiles. The service account used to generate the SSL certificates has the authority to sign the certificate.
PowerShell	Generate Key	If an SSL needs to be generated from Microsoft CA, a key is generated using a pre-existing PowerShell script. The inputs to the script can be pre-coded and don't need to be obtained from the user. Examples of the input are the name and location of the key file, key type (i.e. RSA), and key length (i.e. 2048).
PowerShell	Generate CSR	Generate a CSR using some values from configuration files and some values from the user. For example: Organization, Organizational Unit, City or Locality, State or Province, Country, Challenge password, and Email ID values can be supplied from a configuration file while the Common Name value can be provided by the user as part of the request form.
PowerShell	Request SSL cert	Once the CSR is generated and saved, a PowerShell script can be called to request a SSL certificate from Microsoft CA.  Details of the automated certificate request process, i.e. URL, certificate type, and credentials can be saved in a configuration file and applied to the script at run-time.
PowerShell	Create cert bundle	Since F5 allows uploading a certificate bundle containing private key and SSL cert, this step creates a bundle as required by F5.
PowerShell	Upload cert bundle to F5	PowerShell script can be utilized to upload a certification bundle to F5.
PowerShell	Create F5 profile	Use a PowerShell script to create a profile in F5.
PowerShell	Apply cert bundle to F5 virtual	F5 allows application of certificates to a profile and then attachment of

SWIM LANE	ACTION	NOTES
	server	the profile to a virtual server later.
		Apply the certification bundle to a F5 profile.

The following diagram shows the steps involved in configuring an F5 LTM.



The following steps include details for the tasks shown in the above workflow figure.

1. Create a pool.
LBaaS performs this task automatically. To create a pool manually in the BIG-IP UI, go to **Main > Local Traffic > Pools > Create**, and enter values for the pool parameters.
2. Add members to the pool.
LBaaS performs this task automatically. If you are creating the pool manually, enter node name, address, port, and service type for each member under **New Member**, and click **Add**.
3. Create a virtual server.
LBaaS performs this task automatically. To create a virtual server manually in the BIG-IP UI, go to **Main > Local Traffic > virtual servers > Create**, and enter values for the virtual server parameters.
4. Create a client SSL profile.
LBaaS performs this task automatically if you select HTTPS as the service type for the virtual server. An SSL certificate must be installed using PowerShell scripts, which is automated by the LBaaS workflow. To install a certificate manually in the BIG-IP UI, go to **System > File Management > SSL Certificate List**.
5. Assign a profile to the virtual server.
LBaaS performs this task automatically.
6. Create a health monitor.
LBaaS performs this task automatically if a custom health monitor is needed. To create a health monitor manually in the BIG-IP UI, go to **Main > Local Traffic > Monitors > Create**, and enter values for the monitor parameters.
7. Apply the health monitor to the pool.
LBaaS performs this task automatically.

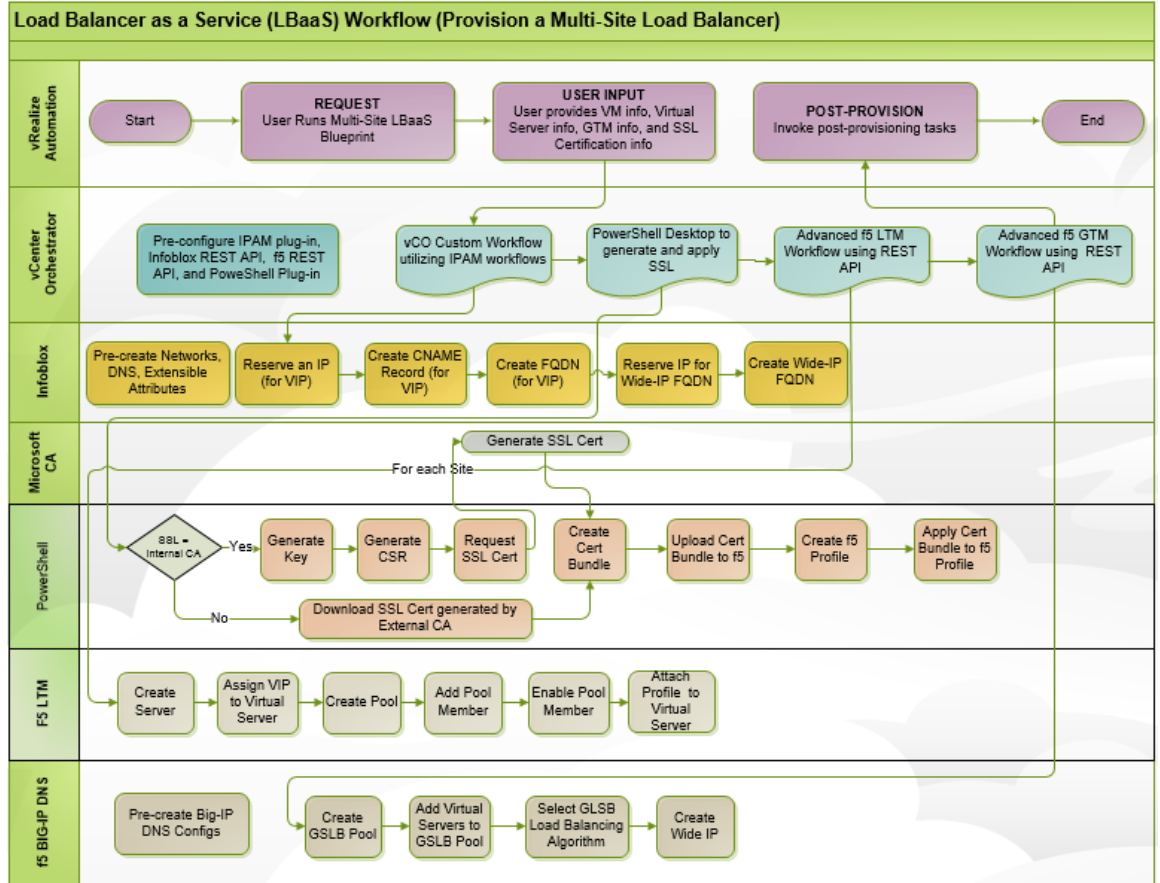
Provisioning a Multi-Site Load Balancer

Provisioning a multi-site load balancer requires more steps than provisioning a single-site load balancer. This use case involves the deployment of two F5 LTMs on two data centers. In addition, a Big-IP DNS is deployed and configured across both data centers to balance traffic across two F5 LTMs. The administrators performing the LBaaS request operation are prompted with fields specific to F5 BIG-IP DNS configurations. The multi-site load balancer workflow can be customized to

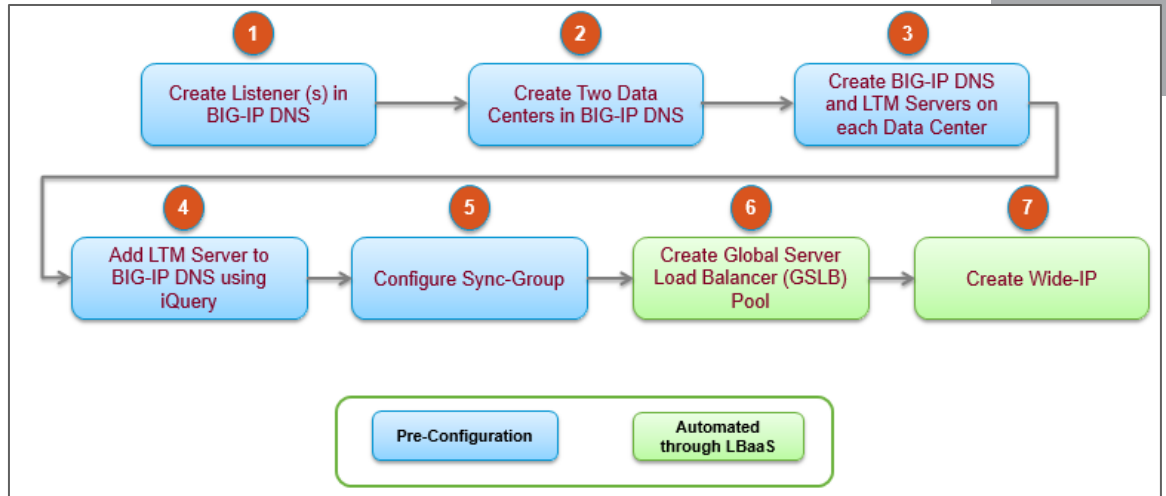
provide administrators a choice to select traffic flow patterns across two sites (i.e. 50/50, 80/20, 60/40, 40/60, 20/80).

Workflow for Provisioning a Multi-Site Load Balancer

The following workflow diagram shows the steps involved in the provisioning of a multi-site load balancer.



When an administrator provisions a multi-site load balancer using vRA blueprint, two identical load balancers are provisioned on each site. In addition, a BIG-IP DNS Wide-IP is created. The administrator is prompted for additional fields including primary site and traffic flow ratio. The following diagram shows tasks involved in configuring BIG-IP DNS Wide-IP, including the tasks needed prior to running the LBaaS workflow.



The following steps include details for the tasks shown in the above workflow figure. The first five steps are prerequisites to running the LBaaS workflow. All tasks are documented for completeness and for reference purposes.

1. In the BIG-IP UI, go to **DNS > Delivery** and provide values for the listener parameters to create a listener. You must create a listener on each site.
This is a prerequisite to the LBaaS workflow.
2. In the BIG-IP UI, go to **DNS > GSLB > Data Centers > Data Center List > New Data Center** and enter a name and other data center parameters to create a data center. Repeat this for the second data center.
This is a prerequisite to the LBaaS workflow.
3. In the BIG-IP UI, go to **DNS > GSLB > servers > server List > New server** to add LTM servers to BIG-IP DNS. Repeat this step for the second data center.
This is a prerequisite to the LBaaS workflow.
4. Configure iQuery.
This is a prerequisite to the LBaaS workflow.
 - a. Log in to the BIG-IP DNS at the primary site.
 - b. Run the command `#bigip_add<rc-ltm-ip>` to add LTM on the primary site to the GTM.
Upon successful communication between BIG-IP DNS and LTMs, the status in the BIG-IP DNS turns green.
5. Configure a Sync-Group
This is a prerequisite to the LBaaS workflow.
 - a. Add both DNS servers to the BIG-IP DNS system.
 - b. Create a Sync-Group.
 - c. Run the command `#gtm_add<rc-ltm-ip>` to add the BIG-IP GTM appliance to the secondary site.
 - d. Run the command `#bigip_add<rc-ltm-ip>` to add the BIG-IP LTM appliance to the secondary site.
6. Create the Global Server Load Balancer (GSLB) Pool.
The pool is created as part of the LBaaS workflow, and the pool name is provided in the XaaS UI. To create the pool manually, go to **GSLB > Pools** and click **Create**.
7. Create Wide-IP.

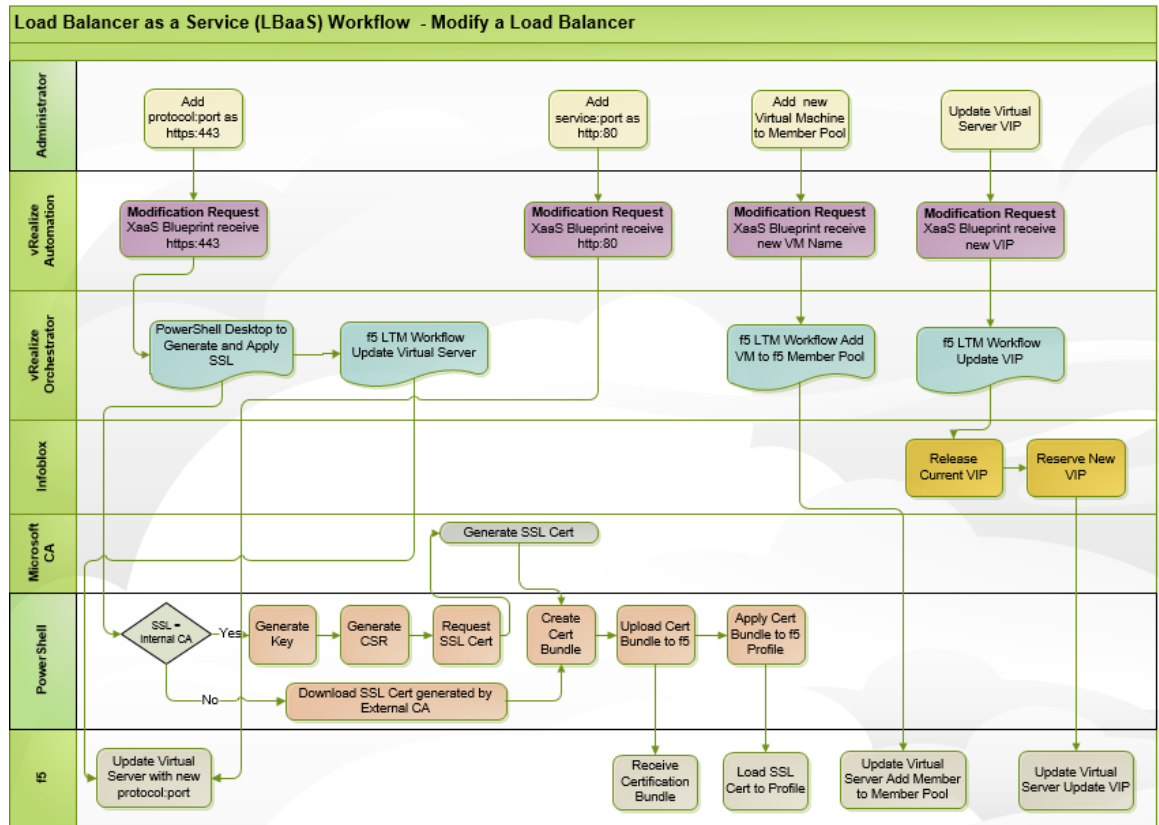
Wide-IP is created as part of the LBaaS workflow. An IP address is assigned to Wide-IP through the vRealize Orchestrator IP reserve workflow from Infoblox. To create Wide-IP manually, go to **GSLB > Wide IPs** and click **Create**.

Modifying a Load Balancer

Modifying a load balancer involves making changes to an existing load balancer deployed on F5. For example, if you need to add a new virtual machine to the F5 virtual server member pool, you can modify an existing load balancer.

Workflow for Modifying a Load Balancer


The following workflow diagram depicts various swim lanes and swim lane specific steps involved in the modifying of a load balancer. Not all modification workflows are represented in the diagram below. It depicts four common load balancer modification requests.



While it is possible to modify everything from an existing virtual server, it is a best practice to delete an existing virtual server and recreate a new one if modification to SSL, VIP, DNS records are required.

The XaaS blueprint allows modification of the following parameters.

#	PARAMETER	NOTES
1	Add/Remove virtual machine	Users should be able to add or remove virtual machine(s) to or from a member pool of a F5 virtual server.
2	virtual server IP	Users should be able to enter a new VIP. In this case the current IP must be returned to the Infoblox and a new IP must be reserved.
3	LB Algorithm	Users should be able to change the current algorithm to a new algorithm using a drop down box.

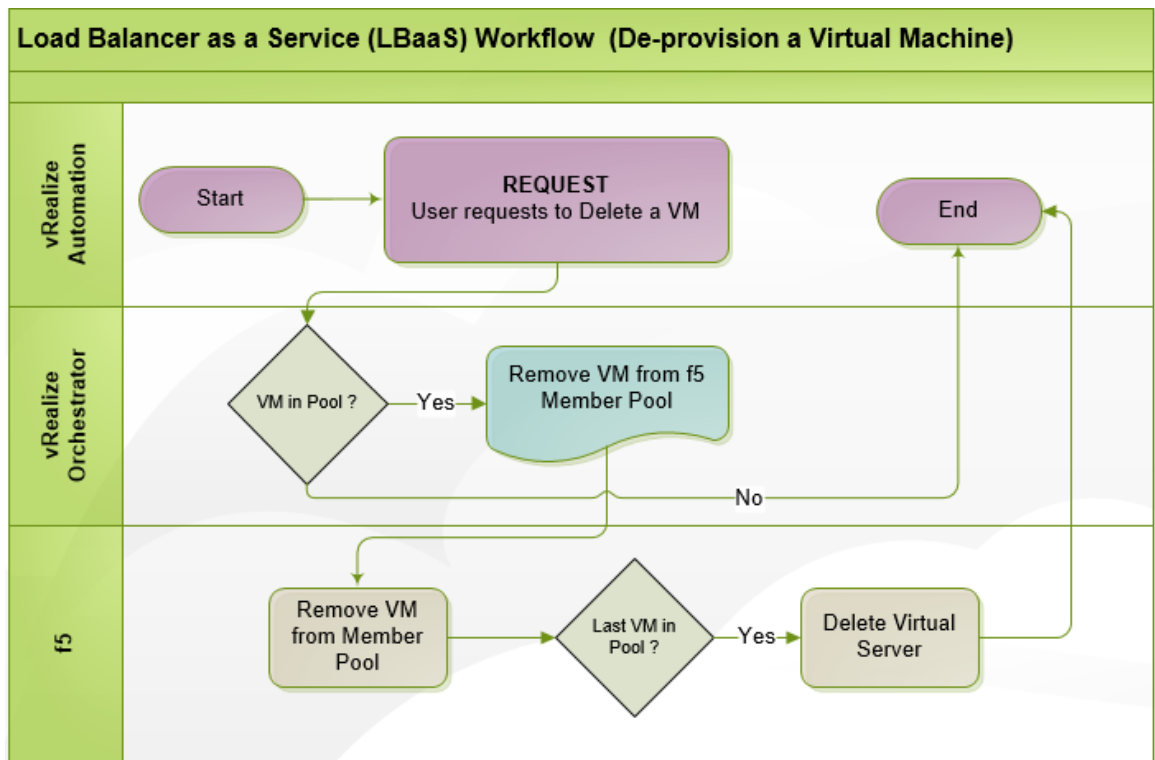
#	PARAMETER	NOTES
4	Add/Remove protocol:port	<p>Users should be able to add/remove service:port(s). For example, if the user wants add http:80 to a virtual server for the sake of testing, she/he should be able to add it using the XaaS blueprint.</p> <p> If https:443 (or any other https port requiring a CA generated SSL) is added to the virtual server, the workflow must also invoke generation of an SSL from the Microsoft CA or import an existing SSL provided by an external CA.</p>
5	virtual server Name	Users will be able to update the virtual server name using the XaaS blueprint.
6	Monitoring Type, Monitoring URL, Max Retries, Monitor Timeout	Users will be able to update the monitoring parameters using the XaaS blueprint.
7	Aliases	Users should be able to update Aliases using XaaS blueprint.
8	Persistence Option	Cookie, Source IP
9	Secondary Pool	If the primary pool is down, use the secondary pool.
10	F5 Profile	HTTP Profile

De-provisioning a Virtual Machine

When a virtual machine of an F5 member pool is de-provisioned using a vRealize Automation workflow, the virtual machine must be removed from the member pool. To remove a virtual machine from an F5 virtual server member pool, a new XaaS workflow must be created and linked to the virtual machine de-provisioning workflow. The XaaS workflow uses vRealize Orchestrator to remove the virtual machine from the F5 virtual server pool. If the virtual machine is the last virtual machine in a member pool, the member pool must be deleted, which in turn means that the F5 virtual server must be deleted.

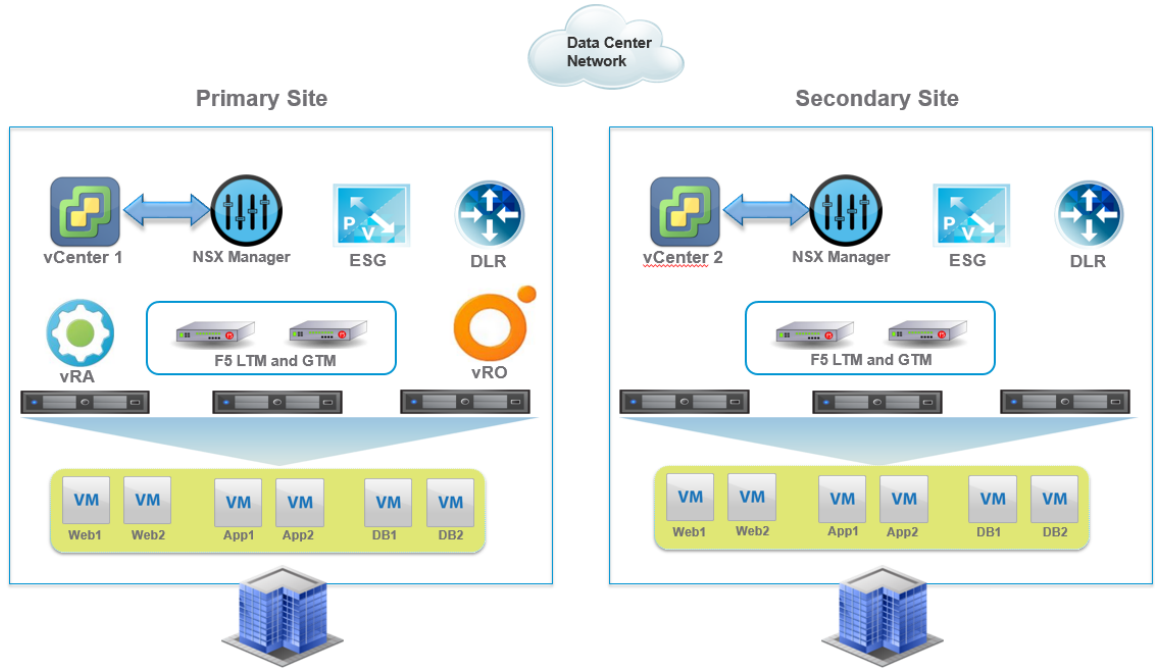
Workflow for De-Provisioning a Virtual Machine

The following workflow diagram depicts the workflow of de-provisioning a virtual server or removing a virtual machine from a F5 virtual server member pool.



Load Balancer as a Service High-Level Design

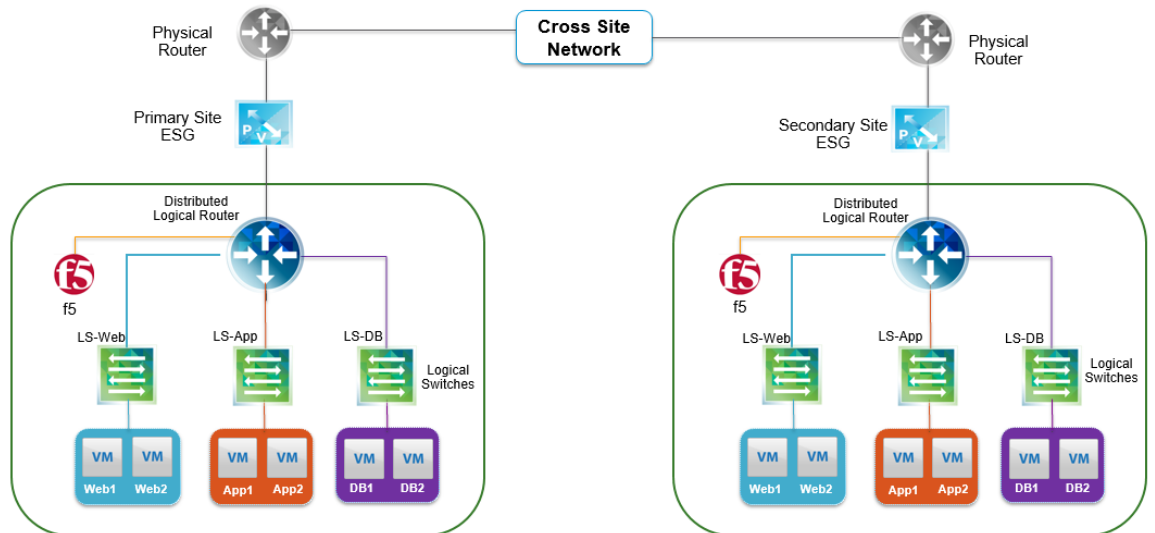
This section describes various design aspects of developing and executing LBaaS. The following diagram shows the products involved in LBaaS design.



Load Balancer as a Service Network Design Using NSX

VMware NSX allows provisioning of virtual networks based on VXLAN. This paper assumes that the application servers, i.e. Web, App, and DB utilize NSX. NSX logical switches provide switching among servers within a server group, i.e. Web. NSX Distributed Logical Router (DLR) provides routing between web, app, and DB servers.

The following diagram depicts logical network designs for the LBaaS workflows.



Load Balancer as a Service BIG-IP DNS Design

Big-IP DNS provides capabilities to improve the availability of the applications by intelligently directing users to the closest or best performing site. Two BIG-IP DNS systems deployed at two sites essentially work as a single unit. The BIG-IP DNSs at *Primary Site* and *Secondary Site* will have primary and secondary authoritative name service responsibilities. The users receive an optimal IP address, which is the LTM virtual server IP (VIP), from *Primary site* or *Secondary site* based on selection policies such as resource availability, performance, load, geolocation, or any other user-defined policies.

BIG-IP DNS supports many types of queries, though “A” type query is most commonly used. When BIG-IP DNS receives a query whose query domain name and type match that of a Wide-IP, it first selects a Global Server Load Balancer Pool to satisfy the response. Then it selects a virtual server from the pool, which will respond with an IP address. Which virtual server (i.e. from *Primary site* or *Secondary site*) is provided from the GSLB pool is based on the load balancing method on each site and the availability of the resources at run-time.

BIG-IP DNS High Level Design

A Wide-IP, which is an FQDN, serves as the entry point for accessing the application URL. Typically, the application URL represents a web application on a group of web servers (e.g. Apache or IIS). The BIG-IP DNS performs query resolution to resolve Wide-IP FQDN to a VIP from a GSLB pool that contains two virtual servers – one from *Primary site* and the other from *Secondary site*. The virtual servers, created on the LTM appliances, contain the appropriate application or web servers. VIP from LTM further resolves the IP to a member IP from its pool.

BIG-IP DNS Component Design

BIG-IP DNS configuration involves the following components.

Listener: A BIG-IP DNS object that processes and responds to DNS queries. Listener configuration on BIG-IP DNS will need to be pre-created prior to running the LBaaS workflow.

Data Center: A container object that represents the location of application delivery components. It contains two LTM virtual servers – one from each site. Data Center configuration on BIG-IP DNS will need to be pre-created prior to running the LBaaS workflow.

Server: A container object that represents a system on which application components are hosted. A server can be a BIG-IP DNS, an LTM server or a Physical server. server configuration on BIG-IP DNS will need to be pre-created prior to running the LBaaS workflow.

Virtual server: Represents the IP address and port of a service that is hosted on a server in a data center. BIG-IP DNS calls the IP addresses and ports. During the intelligent query resolution process, BIG-IP DNS selects the IP address the virtual server that is optimal based on the selection criteria. virtual servers will need to be pre-created on BIG-IP DNS prior to running the LBaaS workflow.

Pool: A logical object configured on the BIG-IP DNS system, virtual servers can be organized into various pools for intelligent resolution. Related virtual servers can be grouped together to a GSLB pool. The pool configuration for LBaaS will be created on-demand using LBaaS workflow.

Wide-IP: A logical container that groups GSLB pools and is represented by a FQDN. It contains a set of related virtual servers. This object will be created and configured as part of the LBaaS workflow. The IP address for the FQDN will be reserved on Infoblox using a vRO workflow.

DIG: DNS Resolution utility is a tool to test Wide-IP configuration. It can be downloaded to a PC. Users can choose to download the utility to the PowerShell desktop to perform the validation of Wide-IP configuration by LBaaS workflow. The command `#dig @listener-ip wide-ip-name` sends the DNS query to the Listener and present the response from BIG-IP DNS.

BIG-IP DNS Load Balancing Algorithm Design

Based on the LBaaS design, three types of Load Balancer Algorithms can be configured dynamically to the BIG-IP DNS GSLB pool as follows.

Global Availability

This load balancing algorithm closely matches the *Active/Standby* requirement. If *Global Availability* is the selected load balancer algorithm, the BIG-IP DNS distributes DNS name resolution requests to the first available virtual server in a pool. BIG-IP DNS starts at the top of a configured list of virtual servers and sends requests to the first available virtual server in the list. Only when the virtual server becomes unavailable does BIG-IP DNS send requests to the next virtual server in the list.

Ratio

Another requirement for LBaaS workflow is to control traffic flow across two sites. For example, when the *Primary* site contains six servers and the *Secondary* site contains four servers of an application, an ideal configuration is to send 60% traffic to the *Primary* site and 40% to the *Secondary* site. BIG-IP DNS *Ratio* load balancing algorithm can meet this requirement.

The *Ratio* load balancing method distributes DNS name resolution requests among the virtual servers in a pool using *weighed round robin*. For the *Ratio* Load Balancing algorithm to work properly, a weight must be assigned to a virtual server. To service the sample use case above, a weight of 60 must be assigned to the primary site virtual server and a weight of 40 must be assigned to the secondary site virtual server.

Round-Robin

Another requirement for LBaaS workflow is to be able to split the inbound requests using *Active/Active* load balancing across both sites. To service this requirement, the BIG-IP DNS round-robin algorithm must be selected for the GSLB pool. With this load balancing method, BIG-IP DNS name resolution requests are executed in a circular and sequential pattern among the virtual servers in a GSLB pool. Over-time, each virtual server receives an equal number of requests.

Load Balancer as a Service LTM Design

LBaaS design assumes that BIG-IP virtual Editions (VE) are deployed on both sites. It is also assumed that Device Service Clustering (DSC) is utilized for each pair of Big-IP appliances using an *Active/Active* configuration. In addition, LBaaS design assumes that a Sync failover device configuration exists on the VE pair so that the device configuration is synched between the members of the pair and so that the devices can failover to one another.

BIG-IP LTM Configuration Objects

The following objects are configured in F5 LTM using an LBaaS workflow:


Node: Defines the IP address of a physical or virtual device on a network, e.g., a web server or an application server. A single node may host many application services, each of which can be serviced by a Pool Member.

Pool Member: Defined by the combination of an IP address and a port number. It represents an application service hosted on a node. A Pool Member is delivered through a BIG-IP system.

Pool: One or more Pool Members can be grouped together into a Pool for the purpose of load balancing. The configuration of a *pool* involves identification of pool members and the load balancing method. The load balancing method along with other criteria determines which pool member will be selected.

Virtual Server: A virtual server is a type of listener that allows matching traffic types (virtual server IP and port) and redirects them to pool members based on a load balancing method. An F5 LTM is a default-deny system; Hence all traffic is blocked by default. virtual servers are conceptual entities that the clients connect to.

Health Monitors: These are monitoring mechanisms to determine the availability of a virtual server, pool, or pool member. Four types of health monitors are available in F5 LTM: Address/Service Check Monitor, Content Check Monitor, Application Check Monitor, and Path Check Monitor. For LBaaS design *Content Check Monitor* is recommended.



LTM Load Balancing Method Design

Several load balancing methods are available for LTM virtual server traffic processing. The users will be presented with a drop-down box to select a load balancing method from the following list.

1. Round Robin (Default)
2. Ratio
3. Least Connections
4. Weighted Least Connections
5. Fastest
6. Observed
7. Predictive
8. Dynamic Ratio
9. Least Sessions

Round-Robin and Ratio are static load balancing methods while the others are dynamic. Users can select *Ratio* as the load balancing method when they want to send traffic to members in a pool based on a weight factor. If the user selects *Ratio* as the load balancing method, additional text fields must be presented on the XaaS form to collect the weight factor for each VM (or member) in the pool.

LTM virtual server Design

F5 LTM offers a several different types of virtual servers. For the LBaaS use case, deployment of a Standard virtual server is recommended. Several features are available with F5 LTM Standard virtual server, including TCP optimization, SSL termination (or offload), HTTP optimization and HTTP compression. NOTE: SSL termination is recommended for the LBaaS workflow.

Load Balancer as a Service Implementation


The following section provides guidance on the development of the user interface and workflow to build LBaaS.

Provision an LTM Virtual Server

A self-service form can be built using vRealize Automation XaaS capability to deploy F5 virtual servers for pre-provisioned Web, App, and DB servers. The blueprint will have the ability to create a virtual server in F5 based on the inputs provided by the requester. This requirement will utilize a pre-existing HA pair of F5 LTM load balancers deployed in each physical data center and mapped to a BIG-IP DNS across the two sites. The virtual servers should get their IP Address assignments from Infoblox from specific networks designated for load balancers only and identifiable by extensible attributes in Infoblox. The networks are identified by the Use attribute with the value specified for F5. There may be more than one of each of these types of networks.

The attribute for port with protocol will need to be set up in a catalog of services with the flexibility to add additional ports and monitors (port 80 with protocol http and 443 with protocol https). These will be used for load balancing, and the default algorithm will be Least Connection. The following form shows a sample UI for creating a single-site load balancer.

New Request



Create a VIP

GTM
LTM

App ID:

Environment:

▼

Protocol:Port:

+

No data selected

Application Name:

Application Description:

LB Algorithm:

▼

Pool Members Rancho:

vmware01_rancho
 vmware02_rancho
 vmware03_rancho

Pool Members Fairfax:

vmware01_fairfax
 vmware02_fairfax
 vmware03_fairfax

vIPAddress:

De-Provision a Virtual Machine

When a virtual machine that is a member of an F5 virtual server member pool is de-provisioned using a vRealize Automation workflow, it must be removed from the F5 virtual server member pool. If the virtual machine is not removed, F5 will issue alerts stating that the virtual machine is unavailable.

To be able to remove a virtual machine from an F5 virtual server member pool, a new XaaS workflow must be created and linked to the VM de-provisioning workflow. There will be no user input to the new XaaS workflow, as it should be able to receive all of the data necessary (i.e. VM name) from the VM de-provisioning workflow. The new XaaS workflow will use vRealize Orchestrator to remove the VM from the F5 virtual server pool. If the VM happens to be the last VM of a member pool, the member pool must be deleted, which in turn means that the F5 virtual server must be deleted.

