DEPLOYMENT GUIDE

Load Balancing Meditech RESTful API

Version 1.1.1

loadbalancer.org

Table of Contents

1. About this Guide

This guide details the steps required to configure a load balanced Meditech API environment utilizing Loadbalancer.org appliances. It covers the configuration of the load balancers and also any Meditech API configuration changes that are required to enable load balancing.

For more information about initial appliance deployment, network configuration and using the Web User Interface (WebUI), please also refer to the Administration Manual.

2. Loadbalancer.org Appliances Supported

All our products can be used with Meditech API servers. For full specifications of available models please refer to https://www.loadbalancer.org/products. Some features may not be supported in all cloud platforms due to platform specific limitations, please check with Loadbalancer.org support for further details.

3. Loadbalancer.org Software Versions Supported

- V8.3.8 and later

Note	The screenshots used throughout this document aim to track the latest Loadbalancer.org software version. If using an older software version, note that the screenshots presented here may not match the WebUI exactly.

4. Meditech RESTful API Software Versions Supported

- Meditech RESTful API – all versions

5. Meditech RESTful API

The RESTful API Infrastructure allows MEDITECH and third party vendor software to securely access the MEDITECH EHR through APIs. Interoperability Services (or IOPS) — which is a component of the RESTful API Infrastructure and installed on the same machine(s) — adds a set of APIs to meet Meaningful Use Stage 3 (MU3) and Imaging Appropriate Use Criteria (AUC) requirements. RESTful API is independent from any other web products or interoperability interfaces MEDITECH offers and requires dedicated hardware.

RESTful API is independent from any other web products or interoperability interfaces MEDITECH offers and requires dedicated hardware.

6. Load Balancing Meditech RESTful API

Note	It's highly recommended that you have a working Meditech RESTful API environment first before implementing the load balancer. SSL certificates must be placed either on the load balancer and/or the real servers. DNS entries for the API and Application end-point for each MRI or HIM database are required. SSL termination or SSL bridging are the recommended configurations for load balancing Meditech RESTful API.

Requirements

API Servers(s)

An optimal RESTful API Infrastructure configuration consists of two or more servers running the RESTful API services as well as Interoperability Services. The cluster helps to ensure better performance and failover protection for the Infrastructure. These servers host the web services which clients connect to.

Hardware

- Server Type: Physical or Virtual
- 4 Cores, 2GHz+
- 4GB RAM
- C: Partition: 60GB - used for OS and service installations
- E: Partition: 40GB - used for server logs

Software

- 64-bit Windows Server 2012 Standard Edition

Cache Server

The Redis service, which is installed on the Cache Server, reduces latency and increases performance on requests by reducing the number of hits to the database. The cache is memory-only, meaning it is never persisted to disk. It is suggested that the Redis service run on its own server. However the Redis service can be installed on one of the API servers if additional servers cannot be obtained. If combining the two servers, it is suggested you increase the RAM available to that API server by 4GB, bringing the total to 8GB for that one API server. This server caches responses and also acts as a messaging service between API Servers.

Hardware

- Server Type: Physical or Virtual
- 4 Cores, 2GHz+
- 4GB RAM
- C: Partition: 60GB - used for OS and service installations

Software

- 64-bit Windows Server 2012 Standard Edition

Database Server

The database stores the configuration and run time details of the RESTful services. It does not store patient data nor does it store any other data that is stored in the MEDITECH database.

A supported database is required. The RESTful API Infrastructure supports:

- MSSQL
- MySQL
- MariaDB

HA Load balancers

- Provides high availability and scalability of the Meditech RESTful API services

- It allows the end user to mitigate or prevent SSL vulnerabilities and to configure the SSL parameters according to their regulatory and corporate requirements
- It allows the end user to install and maintain their SSL certificates in a single location instead of across multiple servers/services
- Using DNS round-robin for failover does not provide graceful failover to the client - the client software needs to be smart enough to retry the connection using another IP, depending on the client's technology, this can take seconds or minutes, whereas failover with a load balancer or proxy is nearly instantaneous
- When configured for SSL Termination, this reduces the CPU load on the RESTful API Infrastructure servers

Persistence (aka Server Affinity)

No persistence is required.

Virtual Service (VIP) Requirements

To provide load balancing and HA for Meditech API and Application, 2 VIPS are required:

- VIP1 : Meditech API
- VIP2 : Meditech Application

Port Requirements

The following table shows the ports that are load balanced:

Port	Protocols	Uses
80	TCP/HTTP	This will respond to API requests that originate from the proxy. Requests directly to the server or from an untrusted proxy will be issued a redirect to HTTPS.
8081	TCP/HTTPS	This will respond to Application requests that originate from the proxy. Requests directly to the server or from an untrusted proxy will be issued a redirect to HTTPS.
443	TCP/HTTPS	This is the connection clients make to access the API and/or Application services proxied by the load balancer/proxy.

Note: Ports displayed are default and can be configured to ports applicable to the customer's environment.

TLS/SSL Termination

There are two suggested configurations concerning the Meditech RESTful API and Application, both of which are supported on the load balancer.

1. **SSL Termination** - this is when the connection to the load balancer is encrypted with SSL but the connection from the load balancer to the RESTful services is not encrypted.
2. **SSL Bridging** - this is when the connection to the load balancer is encrypted with SSL *and* the connection from the load balancer to the RESTful services is also encrypted, sometimes using different certificates.

Note

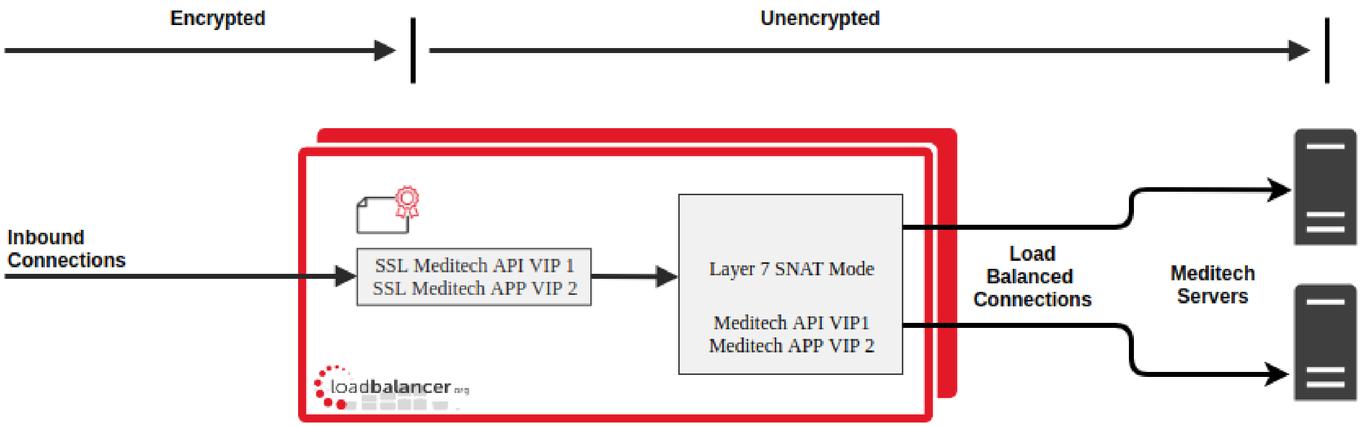
SSL Pass-through is not a recommended configuration as it directly exposes the SSL implementation included in the RESTful services to the Internet.

7. Deployment Concept

Meditech RESTful API can be load balanced in two different ways, as shown in the diagrams that follow:

- **Recommended deployment**: Uses two virtual services to load balance the Meditech API servers which then make a connection to the Cache servers and any other Meditech related server, such as Platform Service, File Library, Monitor Service on the backend.
- **Minimum deployment**: Uses two virtual services to load balance the Meditech API servers which have both the API and Cache services installed on the same server. The server then makes a connection to the other Meditech related servers, such as Platform Service, File Library, Monitor Service on the backend.

Scenario 1 – Recommended Deployment Using SSL Offloading

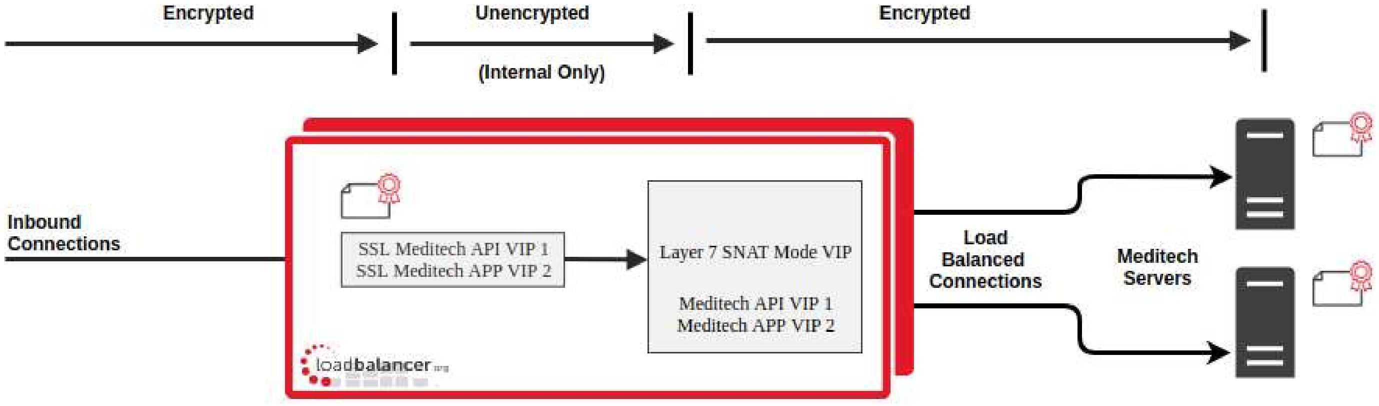


VIPs = **V**irtual **IP** Addresses

Note | The load balancer can be deployed as a single unit, although Loadbalancer.org recommends a clustered pair for resilience & high availability. Please refer to Configuring HA - Adding a Secondary Appliance for more details on configuring a clustered pair.

In this deployment, two virtual services are used in addition to a TLS/SSL termination. The virtual services use layer 7 SNAT mode. An SSL certificate is uploaded and associated to the Virtual Service. Data is encrypted from the client to the load balancer, but is unencrypted from the load balancer to the backend servers as shown above.

Scenario 2 – Recommended Deployment Using SSL Bridging



VIPs = **V**irtual **IP** Addresses

Note — The load balancer can be deployed as a single unit, although Loadbalancer.org recommends a clustered pair for resilience & high availability. Please refer to Configuring HA - Adding a Secondary Appliance for more details on configuring a clustered pair.

In this deployment, two virtual services are used in addition to a TLS/SSL termination. The virtual services use layer 7 SNAT mode with re-encrypt to backend enabled. An SSL certificate is uploaded and associated to the Virtual Service. Data is encrypted from the client to the load balancer and is also encrypted from the load balancer to the backend servers as shown above.

8. Load Balancer Deployment Methods

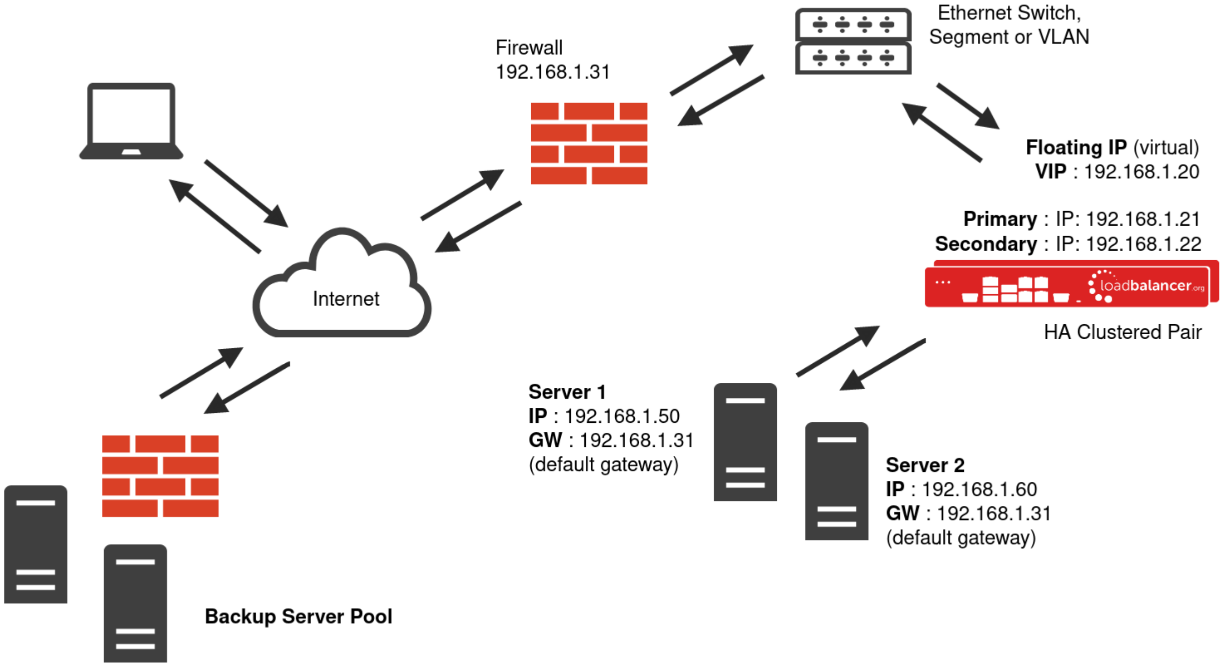
The load balancer can be deployed in 4 fundamental ways: *Layer 4 DR mode*, *Layer 4 NAT mode*, *Layer 4 SNAT mode,* and *Layer 7 SNAT mode*.

For Meditech RESTful API, using layer 7 SNAT mode is recommended due to it being a full proxy meaning the load balanced Real Servers do not need to be changed in any way.

This load balancing mode is described below and is used for the configurations presented in this guide.

Layer 7 SNAT Mode

Layer 7 SNAT mode uses a proxy (HAProxy) at the application layer. Inbound requests are terminated on the load balancer and HAProxy generates a new corresponding request to the chosen Real Server. As a result, Layer 7 is typically not as fast as the Layer 4 methods. Layer 7 is typically chosen when either enhanced options such as SSL termination, cookie based persistence, URL rewriting, header insertion/deletion etc. are required, or when the network topology prohibits the use of the layer 4 methods.



- Because layer 7 SNAT mode is a full proxy, any server in the cluster can be on any accessible subnet including across the Internet or WAN.
- Layer 7 SNAT mode is not transparent by default, i.e. the Real Servers will not see the source IP address of the client, they will see the load balancer's own IP address by default, or any other local appliance IP address if preferred (e.g. the VIP address). This can be configured per layer 7 VIP. If required, the load balancer can be configured to provide the actual client IP address to the Real Servers in 2 ways. Either by inserting a header that contains the client's source IP address, or by modifying the Source Address field of the IP packets and replacing the IP address of the load balancer with the IP address of the client. For more information on these methods please refer to Transparency at Layer 7.
- Layer 7 SNAT mode can be deployed using either a one-arm or two-arm configuration. For two-arm deployments, eth0 is normally used for the internal network and eth1 is used for the external network although this is not mandatory.
- Requires no additional configuration changes to the load balanced Real Servers.
- Port translation is possible with Layer 7 SNAT mode, e.g. VIP:80 → RIP:8080 is supported.
- You should not use the same RIP:PORT combination for layer 7 SNAT mode VIPs and layer 4 SNAT mode VIPs because the required firewall rules conflict.

9. Configuring Meditech RESTful API for Load Balancing

Some changes must be made to the Meditech RESTful API server environment in order for them to be correctly load balanced. **These changes need to be configured by in-house IT staff or a vendor.**

DNS Entries

A DNS entry is needed for the API and Application end-point for each MRI or HIM database (both TEST and LIVE). These records should all point to VIPs on your Load Balancer - you may use one or multiple VIPs when configuring your Load Balancer.

It is suggested that the API services use different VIP(s) from Application services. This makes it easier to restrict access to the Application to only those clients that are on your network. If they shared a VIP, your firewall would need to be able to do deep packet inspection and disallow access to the hostnames for the Application services when accessed over the Internet.

Example

If you have 3 LIVE rings with 1 HIM database each and 3 TEST rings with 1 HIM database each, we would expect the following DNS entries:

```
mtrestapis-live01.CUSTOMER-DOMAIN
mtrestapis-live02.CUSTOMER-DOMAIN
mtrestapis-live03.CUSTOMER-DOMAIN
mtrestapis-test01.CUSTOMER-DOMAIN
mtrestapis-test02.CUSTOMER-DOMAIN
mtrestapis-test03.CUSTOMER-DOMAIN
mtrestapps-live01.CUSTOMER-DOMAIN
mtrestapps-live02.CUSTOMER-DOMAIN
mtrestapps-live03.CUSTOMER-DOMAIN
mtrestapps-test01.CUSTOMER-DOMAIN
mtrestapps-test02.CUSTOMER-DOMAIN
mtrestapps-test03.CUSTOMER-DOMAIN
```

When not exposing the Application to the Internet, only the `mtrestapis-*` entries would resolve on your public DNS. The DNS entries allow the infrastructure to differentiate requests as it is possible that an identifier may be reused in one or more databases. Additionally, the API and Application services run on different ports and within different processes because the workloads are significantly different.

10. Loadbalancer.org Appliance – the Basics

Virtual Appliance

A fully featured, fully supported 30 day trial is available if you are conducting a PoC (Proof of Concept) deployment. The VA is currently available for VMware, Virtual Box, Hyper-V, KVM, XEN and Nutanix AHV and has been optimized for each Hypervisor. By default, the VA is allocated 2 vCPUs, 4GB of RAM and has a 20GB virtual disk. The Virtual Appliance can be downloaded here.

Note

The same download is used for the licensed product, the only difference is that a license key file (supplied by our sales team when the product is purchased) must be applied using the appliance's WebUI.

Note

Please refer to Virtual Appliance Installation and the ReadMe.txt text file included in the VA download for additional information on deploying the VA using the various Hypervisors.

Note

The VA has 4 network adapters. For VMware only the first adapter (eth0) is connected by default. For HyperV, KVM, XEN and Nutanix AHV all adapters are disconnected by default. Use the network configuration screen within the Hypervisor to connect the required adapters.

Initial Network Configuration

After boot up, follow the instructions on the appliance console to configure the management IP address, subnet mask, default gateway, DNS Server and other network settings.

Important

Be sure to set a secure password for the load balancer, when prompted during the setup routine.

Accessing the WebUI

The WebUI is accessed using a web browser. By default, user authentication is based on local Apache .htaccess files. User administration tasks such as adding users and changing passwords can be performed using the WebUI menu option: *Maintenance > Passwords*.

> Note — A number of compatibility issues have been found with various versions of Internet Explorer and Edge. The WebUI has been tested and verified using both Chrome & Firefox.

> Note — If required, users can also be authenticated against LDAP, LDAPS, Active Directory or Radius. For more information please refer to External Authentication.

1. Using a browser, access the WebUI using the following URL:

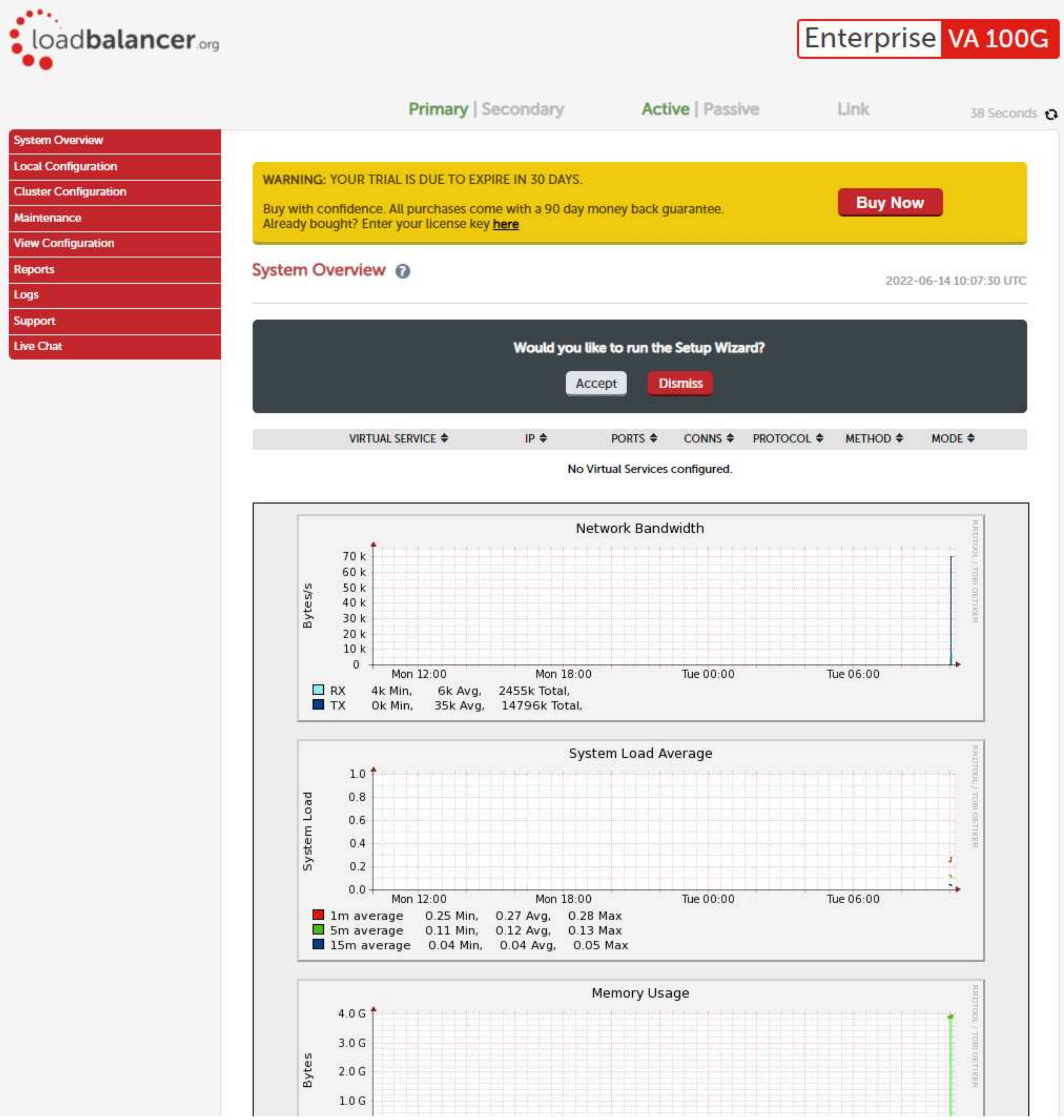
   https://<IP-address-configured-during-network-setup-wizard>:9443/lbadmin/

2. Log in to the WebUI:

   Username: loadbalancer
   Password: <configured-during-network-setup-wizard>

   > Note — To change the password, use the WebUI menu option: *Maintenance > Passwords.*

   Once logged in, the WebUI will be displayed as shown below:



WARNING: YOUR TRIAL IS DUE TO EXPIRE IN 30 DAYS.

Buy with confidence. All purchases come with a 90 day money back guarantee.
Already bought? Enter your license key here

System Overview

2022-06-14 10:07:30 UTC

Would you like to run the Setup Wizard?

VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE

No Virtual Services configured.

> Note — The WebUI for the VA is shown, the hardware and cloud appliances are very similar. The yellow licensing related message is platform & model dependent.

3. You'll be asked if you want to run the Setup Wizard. If you click **Accept** the Layer 7 Virtual Service configuration wizard will start. If you want to configure the appliance manually, simple click **Dismiss**.

Main Menu Options

System Overview - Displays a graphical summary of all VIPs, RIPs and key appliance statistics
Local Configuration - Configure local host settings such as IP address, DNS, system time etc.
Cluster Configuration - Configure load balanced services such as VIPs & RIPs
Maintenance - Perform maintenance tasks such as service restarts and taking backups
View Configuration - Display the saved appliance configuration settings
Reports - View various appliance reports & graphs
Logs - View various appliance logs

Support - Create a support download, contact the support team & access useful links
Live Chat - Start a live chat session with one of our Support Engineers

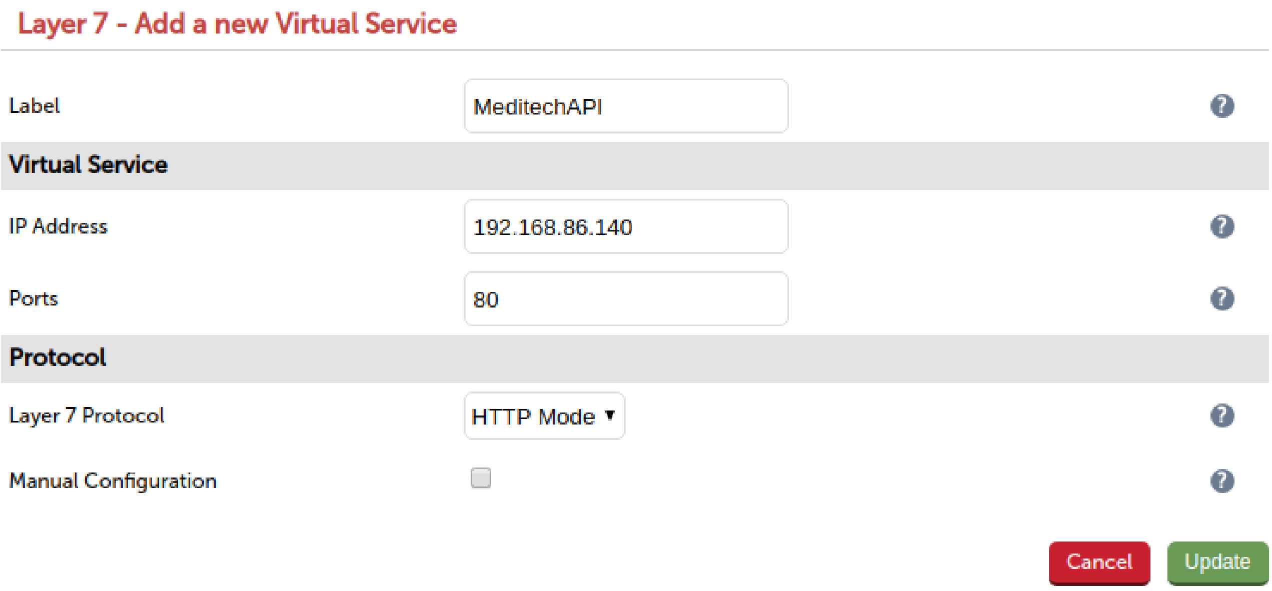
HA Clustered Pair Configuration

Loadbalancer.org recommend that load balancer appliances are deployed in pairs for high availability. In this guide a single unit is deployed first, adding a secondary unit is covered in Configuring HA - Adding a Secondary Appliance.

11. Appliance Configuration for Meditech API – Using Layer 7 SNAT Mode (Scenario 1: Recommended Deployment Using SSL Offloading)

Configuring the API Virtual Service (VIP1)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click on **Add a new Virtual Service**.
2. Define the *Label* for the virtual service as required, e.g. **MeditechAPI**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.86.140**.
4. Set the *Ports* field to **80**.
5. Leave the *Protocol* set to **HTTP mode**.



Layer 7 - Add a new Virtual Service

Label: MeditechAPI

Virtual Service

IP Address: 192.168.86.140

Ports: 80

Protocol

Layer 7 Protocol: HTTP Mode

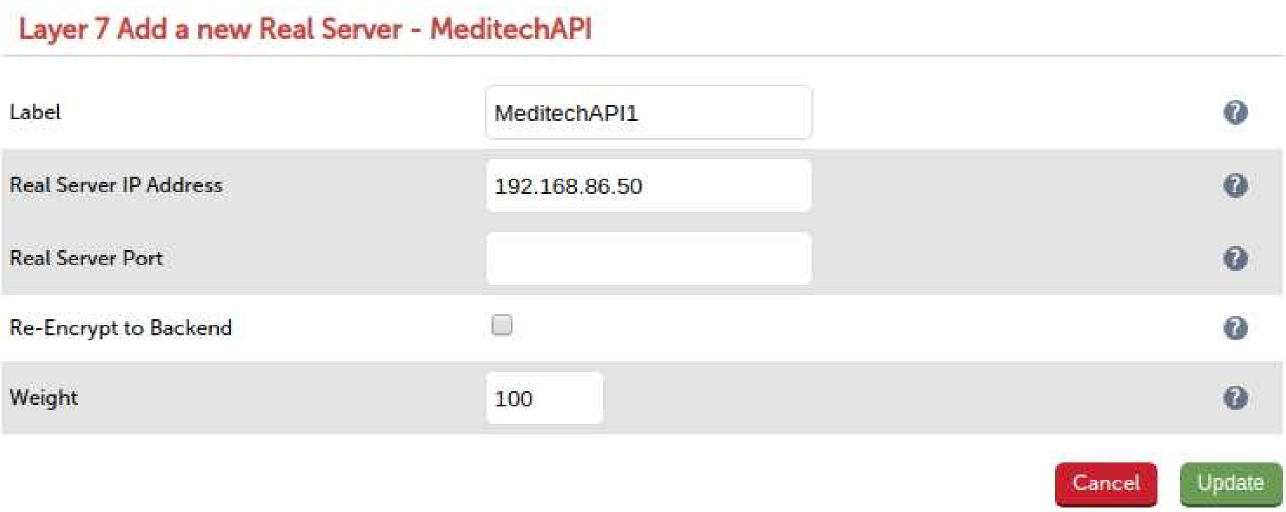
Manual Configuration: □

6. Click **Update** to create the virtual service.
7. Now click **Modify** next to the newly created VIP.
8. Set *Balance mode* to **Weighted Round Robin**.
9. Scroll down to the Persistence section and set *Persistence Mode* to **None**.
10. In the Health Checks section set_Health Checks_ to **Negotiate HTTP (HEAD)**.
11. Leave *Response expected* blank, which will configure the load balancer to look for a '**200 OK**' response.
12. Scroll down to the Other section and click [**Advanced**].
13. Enable (check) the Timeout checkbox and set both *Client Timeout* & *Real Server Timeout* to **5m**.

14. Ensure that **Set X-forwarded-For Header** is enabled (checked).
15. Set *Force to HTTPS* to **Yes**.
16. Click **Update**.

Defining the Real Servers (RIPs)

1. Using the WebUI, navigate to *Cluster Configuration > Layer 7 – Real Servers* and click **Add a new Real Server** next to the newly created VIP.
2. Enter an appropriate label for the RIP, e.g. **MeditechAPI1**.
3. Change the *Real Server IP Address* field to the required IP address, e.g. **192.168.86.50**.
4. Leave the *Real Server Port* field empty.



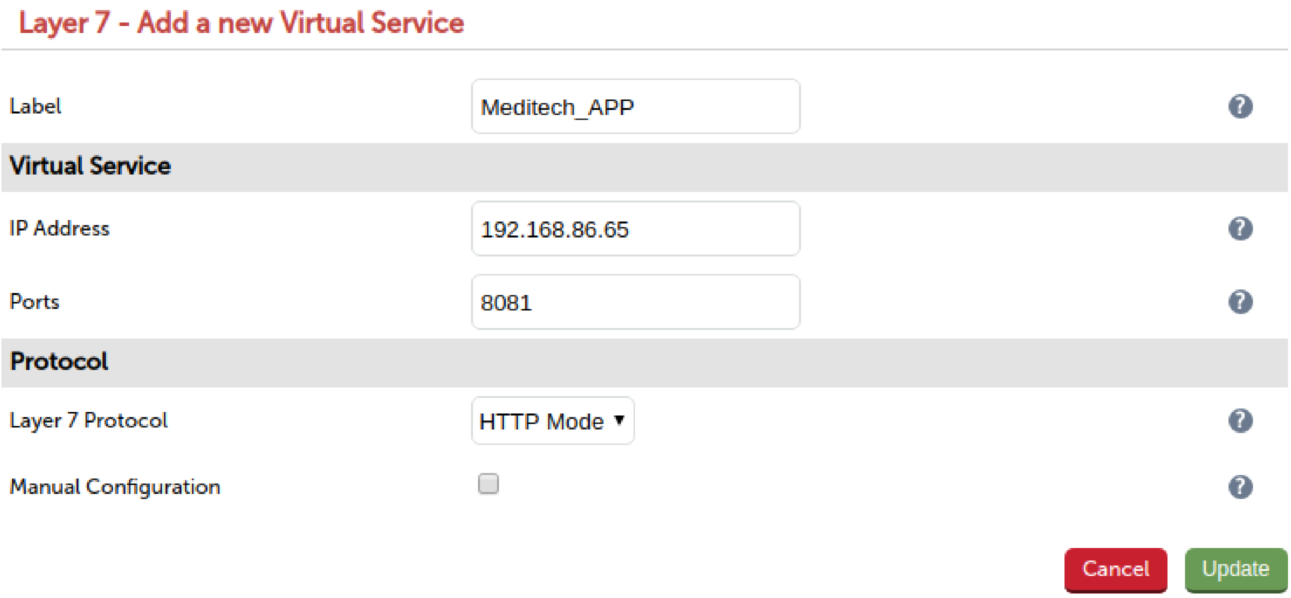
Layer 7 Add a new Real Server - MeditechAPI

Label	MeditechAPI1
Real Server IP Address	192.168.86.50
Real Server Port	
Re-Encrypt to Backend	☐
Weight	100

5. Click **Update**.
6. Repeat the above steps to add additional MeditechAPI Server(s).

Configuring the Application Virtual Service (VIP2)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click on **Add a new Virtual Service**.
2. Define the *Label* for the virtual service as required, e.g. **Meditech_APP**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.86.65**.
4. Set the *Ports* field to **8081**.
5. Leave the *Protocol* set to **HTTP mode**.



Layer 7 - Add a new Virtual Service

Label: Meditech_APP

Virtual Service

IP Address: 192.168.86.65

Ports: 8081

Protocol

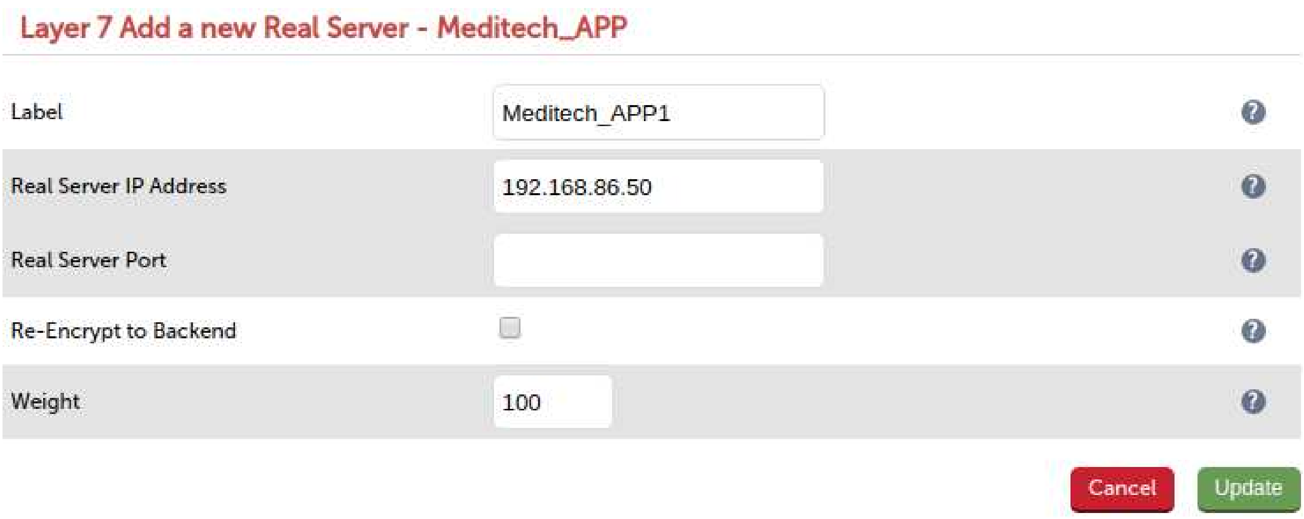
Layer 7 Protocol: HTTP Mode ▾

Manual Configuration: □

6. Click **Update** to create the virtual service.
7. Now click **Modify** next to the newly created VIP.
8. Set *Balance mode* to **Weighted Round Robin**.
9. Scroll down to the Persistence section and set_Persistence Mode_ to **None**.
10. In the Health Checks section set *Health Checks* to **Negotiate HTTP (HEAD)**.
11. Leave *Response expected* blank, which will configure the load balancer to look for a '200 OK' response.
12. Scroll down to the Other section and click [**Advanced**].
13. Enable (check) the Timeout checkbox and set both *Client Timeout _&_Real Server Timeout* to **5m**.
14. Ensure that **Set X-forwarded-For Header** is enabled (checked).
15. Set *Force to HTTPS* to **Yes**.
16. Click **Update**.

Defining the Real Servers (RIPs)

1. Enter an appropriate label for the RIP, e.g. **Meditech_APP1**.
2. Change the *Real Server IP Address* field to the required IP address, e.g. **192.168.86.50**.
3. Leave the *Real Server Port* field empty.



Layer 7 Add a new Real Server - Meditech_APP

Label	Meditech_APP1
Real Server IP Address	192.168.86.50
Real Server Port	
Re-Encrypt to Backend	□
Weight	100

4. Click **Update**.

5. Repeat the above steps to add your other Meditech_APP Server(s).

Setting Up TLS/SSL Termination

Uploading a Certificate

An appropriate certificate must be present on the load balancer for TLS/SSL termination to work. Typically, a valid certificate is uploaded to the load balancer for use. The process for doing this is as follows:

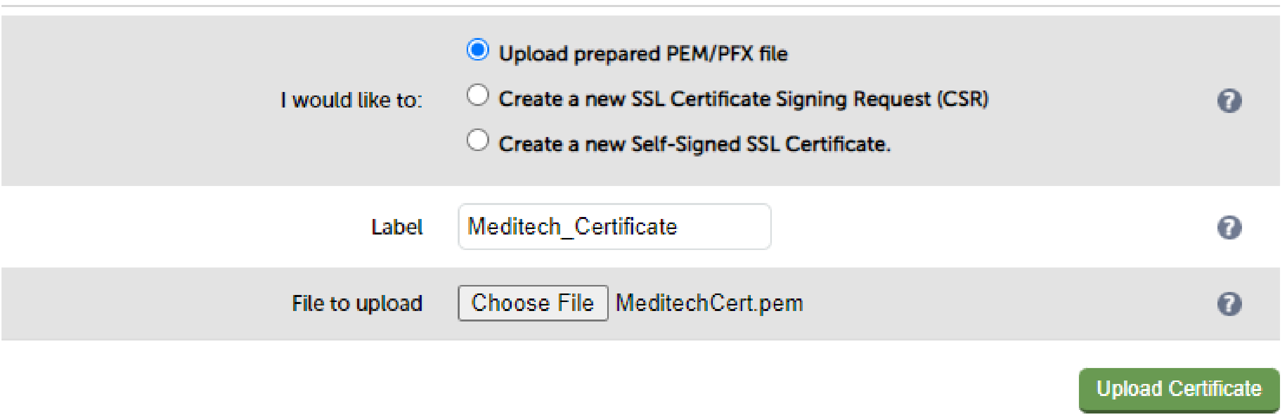
1. Using the web user interface, navigate to *Cluster Configuration > SSL Certificate* and click on **Add a new SSL Certificate**.

2. Press the **Upload prepared PEM/PFX file** radio button.

3. Define the *Label* for the certificate as required, e.g. **Meditech_Certificate**.

4. Click on **Browse** and select the appropriate PEM or PFX style certificate.

5. If uploading a PFX certificate, enter the certificate's password in the *PFX File Password* field.



I would like to:	● Upload prepared PEM/PFX file ○ Create a new SSL Certificate Signing Request (CSR) ○ Create a new Self-Signed SSL Certificate.
Label	Meditech_Certificate
File to upload	Choose File MeditechCert.pem

6. Click **Upload certificate**.

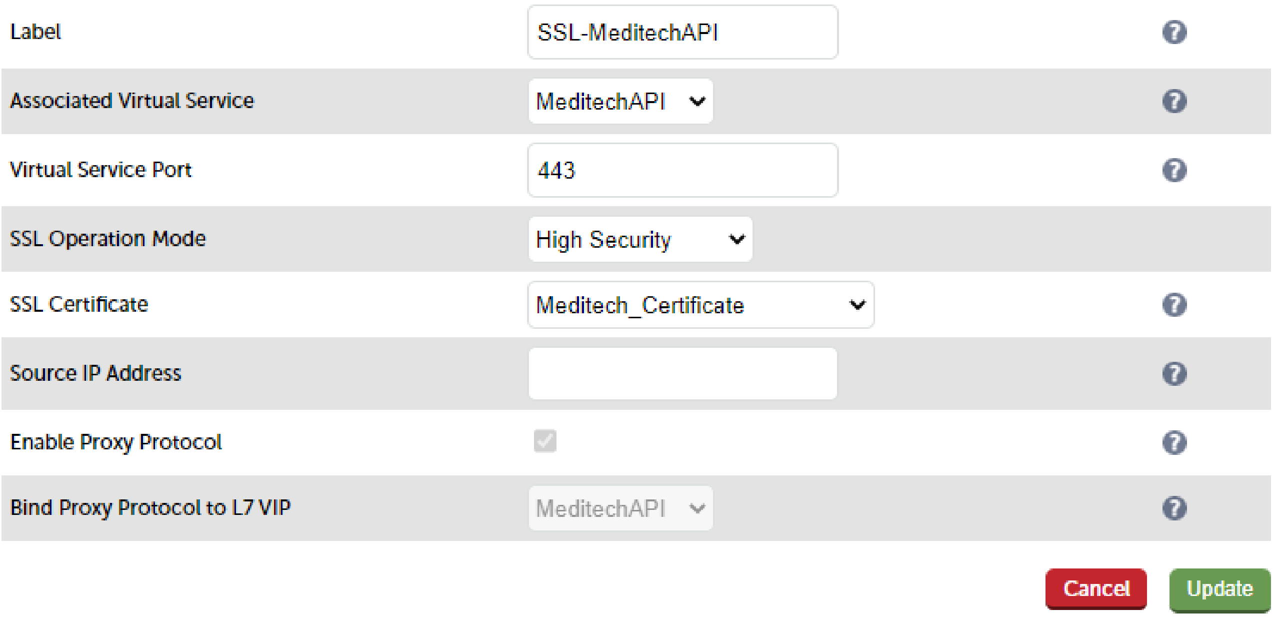
In the absence of a valid certificate, it is also possible to create a Certificate Signing Request (CSR) on the load balancer. A CSR can be submitted to a certificate authority for the issuance of a certificate. For more information please refer to Generating a CSR on the Load Balancer

Creating the TLS/SSL Terminations

1. Using the web user interface, navigate to *Cluster Configuration > SSL Termination* and click on **Add a new Virtual Service**.
2. Using the *Associated Virtual Service* drop-down list, select the **MeditechAPI** service which was created previously.

Note	Once the VIP is selected, the Label field will be auto-populated with **SSL-MeditechAPI**. This can be changed if preferred.

3. Set the *Virtual Service Port* field to **443**.
4. From the *SSL Certificate* drop-down list, select the appropriate certificate.



Label	SSL-MeditechAPI
Associated Virtual Service	MeditechAPI
Virtual Service Port	443
SSL Operation Mode	High Security
SSL Certificate	Meditech_Certificate
Source IP Address	
Enable Proxy Protocol	☑
Bind Proxy Protocol to L7 VIP	MeditechAPI

5. Click **Update** to create the TLS/SSL termination service.
6. Repeat the above steps to configure TLS/SSL termination for your other **Meditech_APP** virtual service.

Finalizing the Configuration

To apply the new settings, HAProxy and STunnel must both be reloaded. This can be done using the buttons in the blue box at the top of the screen or by using the *Restart Services* menu option:

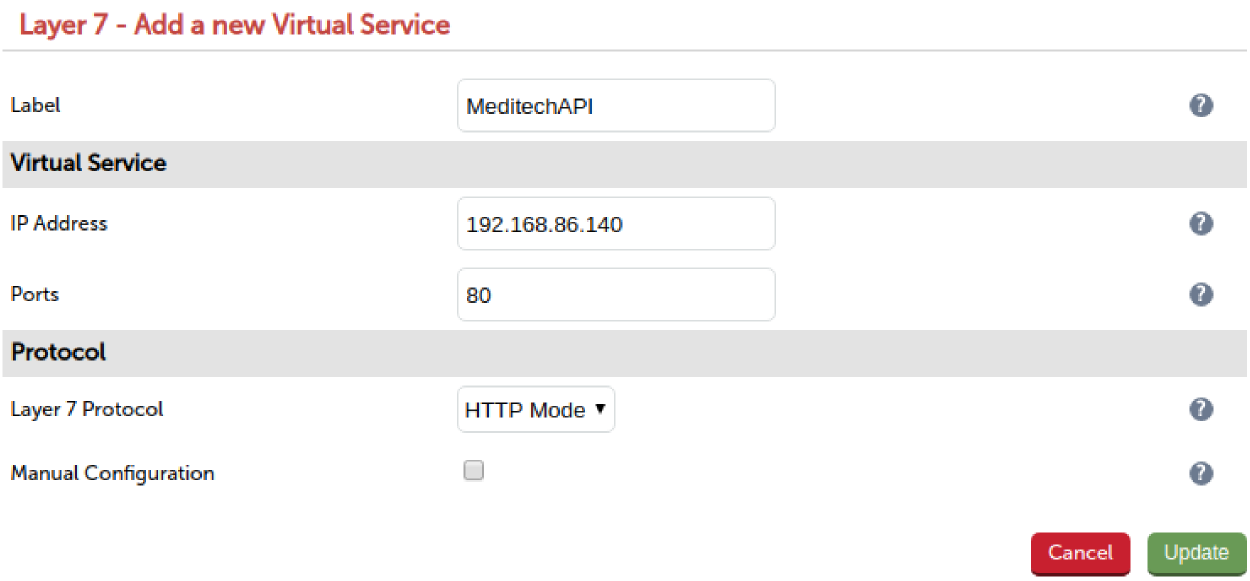
1. Using the WebUI, navigate to: *Maintenance > Restart Services*.
2. Click **Reload HAProxy**.
3. Click **Reload STunnel**.

12. Appliance Configuration for Meditech API – Using Layer 7 SNAT Mode (Scenario 2: Recommended Deployment Using SSL Bridging)

Configuring the API Virtual Service (VIP1)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click on **Add a new Virtual Service**.

2. Define the *Label* for the virtual service as required, e.g. **MeditechAPI**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.86.140**.
4. Set the *Ports* field to **80**.
5. Leave the *Protocol* set to **HTTP mode**.



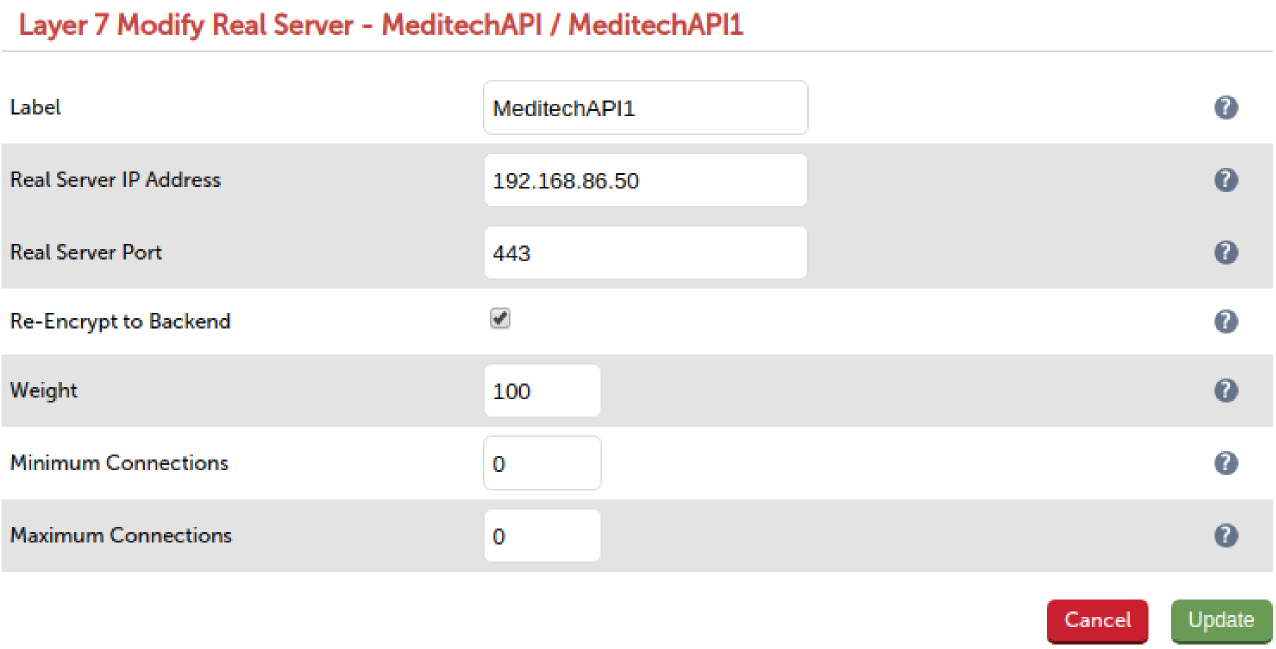
Layer 7 - Add a new Virtual Service

Label	MeditechAPI
Virtual Service	
IP Address	192.168.86.140
Ports	80
Protocol	
Layer 7 Protocol	HTTP Mode ▾
Manual Configuration	□

6. Click **Update** to create the virtual service.
7. Now click **Modify** next to the newly created VIP.
8. Set *Balance mode* to **Weighted Round Robin**.
9. Scroll down to the Persistence section and set_Persistence Mode_ to **None**.
10. In the Health Checks section set *Health Checks* to **Negotiate HTTP (HEAD)**.
11. Leave *Response expected* blank, which will configure the load balancer to look for a '200 OK' response.
12. Scroll down to the SSL section and check the **Enable Backend Encryption** box.
13. Scroll down to the Other section and click [**Advanced**].
14. Enable (check) the Timeout checkbox and set both *Client Timeout* & *Real Server Timeout* to **5m**.
15. Ensure that **Set X-forwarded-For Header** is enabled (checked).
16. Set *Force to HTTPS* to **Yes**.
17. Click **Update**.

Defining the Real Servers (RIPs)

1. Enter an appropriate label for the RIP, e.g. **Meditech_API1**.
2. Change the *Real Server IP Address* field to the required IP address, e.g. **192.168.86.50**.
3. Enter **443** in the *Real Server Port* field.
4. Enable the *Re-Encrypt to Backend* check-box.



Layer 7 Modify Real Server - MeditechAPI / MeditechAPI1

Label	MeditechAPI1
Real Server IP Address	192.168.86.50
Real Server Port	443
Re-Encrypt to Backend	☑
Weight	100
Minimum Connections	0
Maximum Connections	0

5. Click **Update**.

6. Repeat the above steps to add your other Meditech_API Server(s).

Configuring the Application Virtual Service (VIP2)

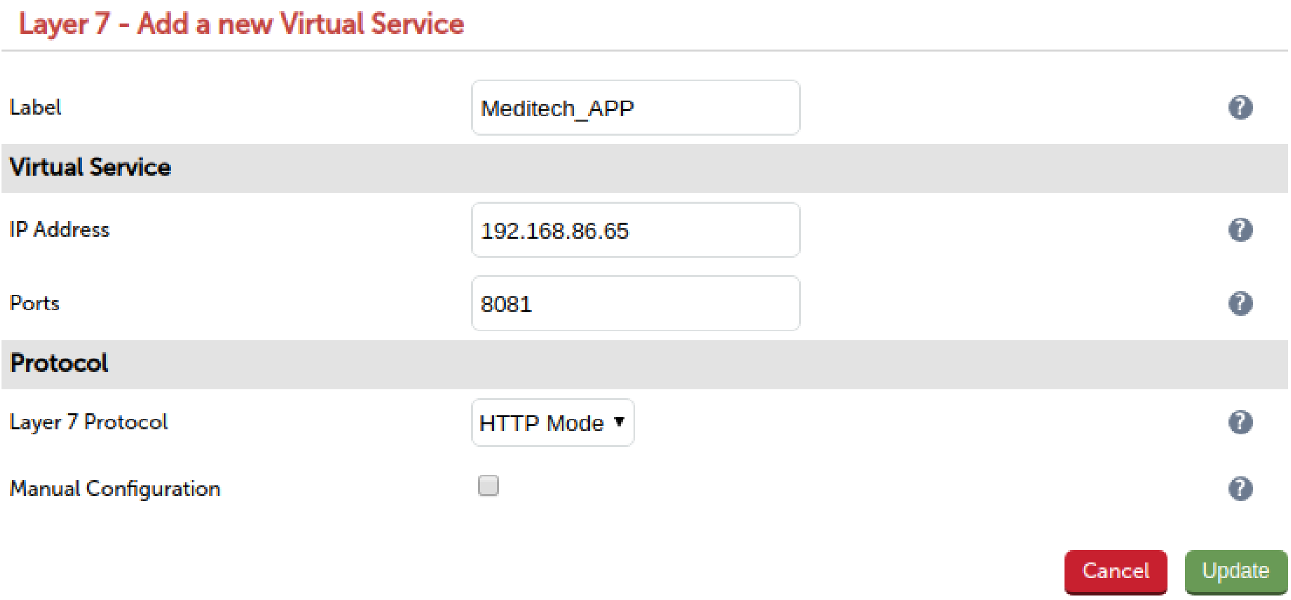
1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click on **Add a new Virtual Service**.

2. Define the *Label* for the virtual service as required, e.g. **Meditech_APP**.

3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.86.65**.

4. Set the *Ports* field to **8081**.

5. Leave the *Protocol* set to **HTTP mode**.



Layer 7 - Add a new Virtual Service

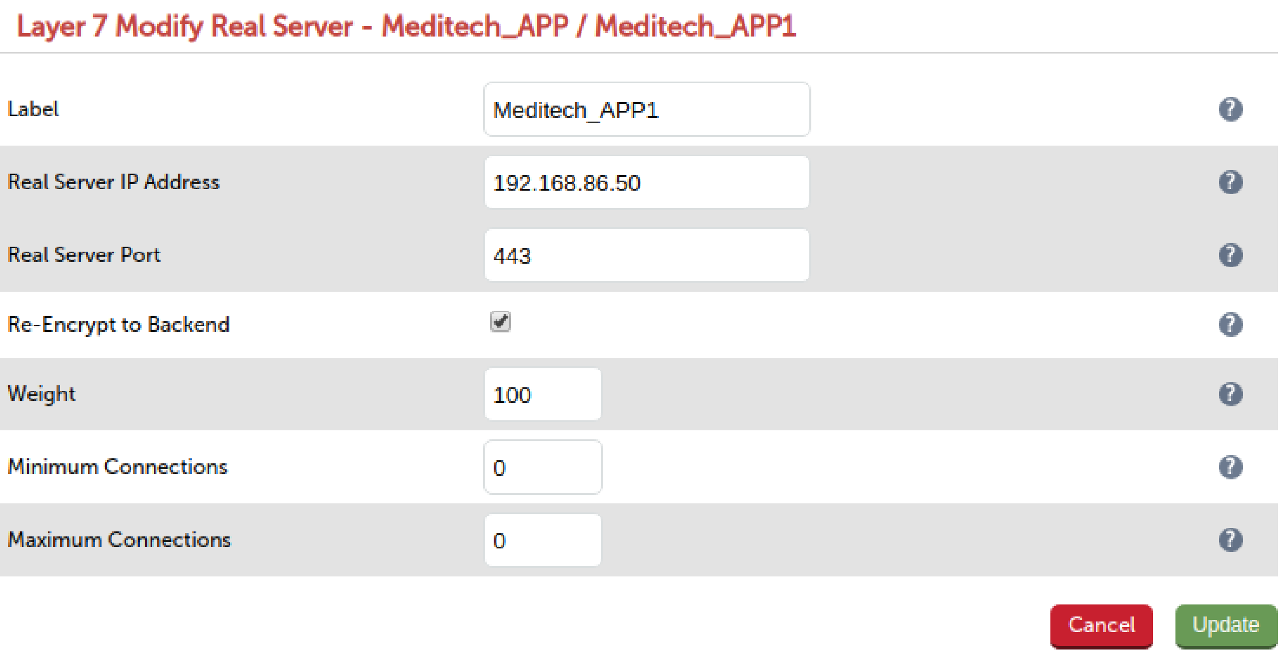
Label	Meditech_APP
Virtual Service	
IP Address	192.168.86.65
Ports	8081
Protocol	
Layer 7 Protocol	HTTP Mode ▾
Manual Configuration	□

6. Click **Update** to create the virtual service.

7. Now click **Modify** next to the newly created VIP.
8. Set *Balance mode* to **Weighted Round Robin**.
9. Scroll down to the Persistence section and set_Persistence Mode_ to **None**.
10. In the Health Checks section set *Health Checks* to **Negotiate HTTP (HEAD)**.
11. Leave *Response expected* blank, which will configure the load balancer to look for a '200 OK' response.
12. Scroll down to the SSL section and check the **Enable Backend Encryption** box.
13. Scroll down to the Other section and click [**Advanced**].
14. Enable (check) the Timeout checkbox and set both *Client Timeout* & *Real Server Timeout* to **5m**.
15. Ensure that **Set X-forwarded-For Header** is enabled (checked).
16. Set *Force to HTTPS* to **Yes**.
17. Click **Update**.

Defining the Real Servers (RIPs)

1. Enter an appropriate label for the RIP, e.g. **Meditech_APP1**.
2. Change the *Real Server IP Address* field to the required IP address, e.g. **192.168.86.50**.
3. Enter **443** in the *Real Server Port* field.
4. Enable the *Re-Encrypt to Backend* check-box.



Layer 7 Modify Real Server - Meditech_APP / Meditech_APP1

Label	Meditech_APP1
Real Server IP Address	192.168.86.50
Real Server Port	443
Re-Encrypt to Backend	☑
Weight	100
Minimum Connections	0
Maximum Connections	0

5. Click **Update**.
6. Repeat the above steps to add your other Meditech_APP Server(s).

Uploading a Certificate

1. Follow the steps in Uploading a Certificate.

Creating the TLS/SSL Terminations

1. Follow the steps in Creating the TLS/SSL Terminations.

Finalizing the Configuration

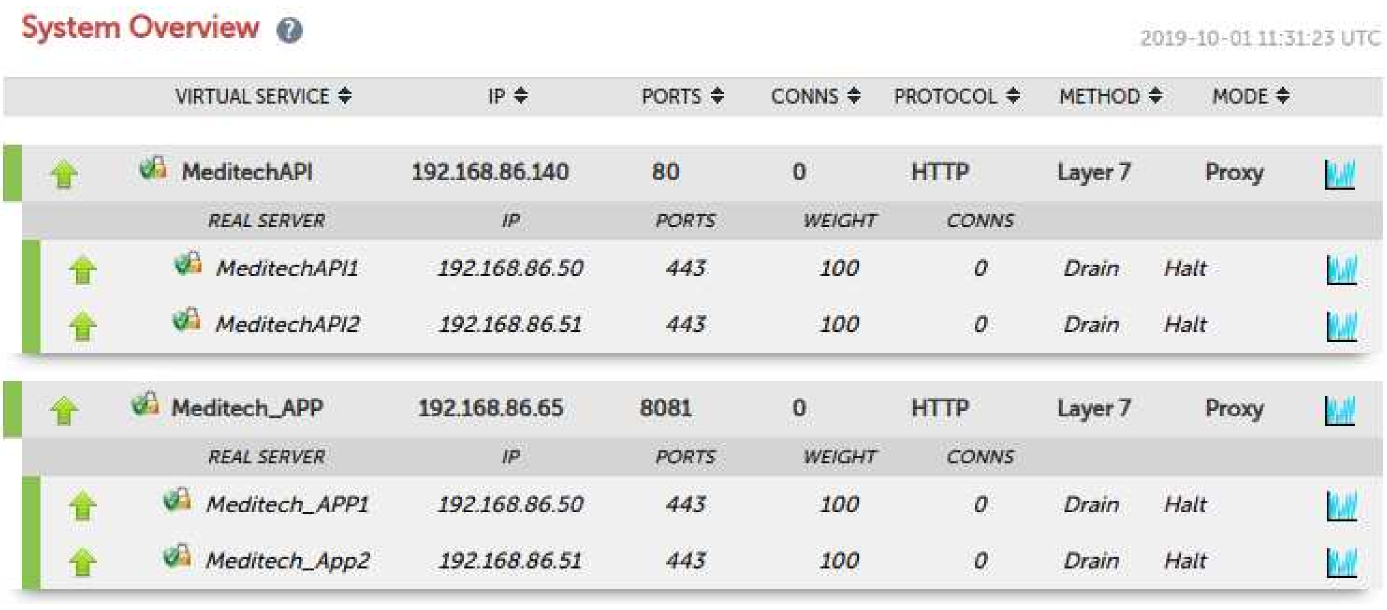
1. Follow the steps in Finalizing the Configuration.

13. Testing & Verification

Note | For additional guidance on diagnosing and resolving any issues you may have, please also refer to Diagnostics & Troubleshooting.

Using System Overview

The System Overview can be viewed in the WebUI. It shows a graphical view of all VIPs & RIPs (i.e. the Meditech nodes) and shows the state/health of each server as well as the state of the cluster as a whole. The example below shows that all Meditech nodes are healthy and available to accept connections:



System Overview

2019-10-01 11:31:23 UTC

VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE
MeditechAPI	192.168.86.140	80	0	HTTP	Layer 7	Proxy
REAL SERVER	IP	PORTS	WEIGHT	CONNS		
MeditechAPI1	192.168.86.50	443	100	0	Drain	Halt
MeditechAPI2	192.168.86.51	443	100	0	Drain	Halt
Meditech_APP	192.168.86.65	8081	0	HTTP	Layer 7	Proxy
REAL SERVER	IP	PORTS	WEIGHT	CONNS		
Meditech_APP1	192.168.86.50	443	100	0	Drain	Halt
Meditech_App2	192.168.86.51	443	100	0	Drain	Halt

Note | Shows scenario 2 with padlock icons on the Real Servers to indicate that backend re-encryption is enabled. The padlock icon next to the VIP indicate that a corresponding SSL termination has been configured.

14. Technical Support

For more details about configuring the appliance and assistance with designing your deployment please don't hesitate to contact the support team using the following email address: support@loadbalancer.org.

15. Further Documentation

The Administration Manual contains much more information about configuring and deploying the appliance. It's available here: https://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf.

16. Conclusion

Loadbalancer.org appliances provide a very cost effective solution for highly available load balanced Meditech RESTful API environments.

17. Appendix

Configuring HA - Adding a Secondary Appliance

Our recommended configuration is to use a clustered HA pair of load balancers to provide a highly available and resilient load balancing solution.

We recommend that the Primary appliance should be configured first, then the Secondary should be added. Once the Primary and Secondary are paired, all load balanced services configured on the Primary are automatically replicated to the Secondary over the network using SSH/SCP.

Note

For Enterprise Azure, the HA pair should be configured first. In Azure, when creating a VIP using an HA pair, 2 private IPs must be specified – one for the VIP when it's active on the Primary and one for the VIP when it's active on the Secondary. Configuring the HA pair first, enables both IPs to be specified when the VIP is created.

The clustered HA pair uses Heartbeat to determine the state of the other appliance. Should the active device (normally the Primary) suffer a failure, the passive device (normally the Secondary) will take over.

Non-Replicated Settings

A number of settings are not replicated as part of the Primary/Secondary pairing process and therefore must be manually configured on the Secondary appliance. These are listed by WebUI menu option in the table below:

WebUI Main Menu Option	Sub Menu Option	Description
Local Configuration	Hostname & DNS	Hostname and DNS settings
Local Configuration	Network Interface Configuration	All network settings including IP address(es), bonding configuration and VLANs
Local Configuration	Routing	Routing configuration including default gateways and static routes
Local Configuration	System Date & time	All time and date related settings
Local Configuration	Physical – Advanced Configuration	Various settings including Internet Proxy, Management Gateway, Firewall connection tracking table size, NIC offloading, SMTP relay, logging and Syslog Server
Local Configuration	Security	Appliance security settings
Local Configuration	SNMP Configuration	Appliance SNMP settings
Local Configuration	Graphing	Appliance graphing settings
Local Configuration	License Key	Appliance licensing
Maintenance	Software Updates	Appliance software update management
Maintenance	Firewall Script	Appliance firewall (iptables) configuration
Maintenance	Firewall Lockdown Wizard	Appliance management lockdown settings

Important

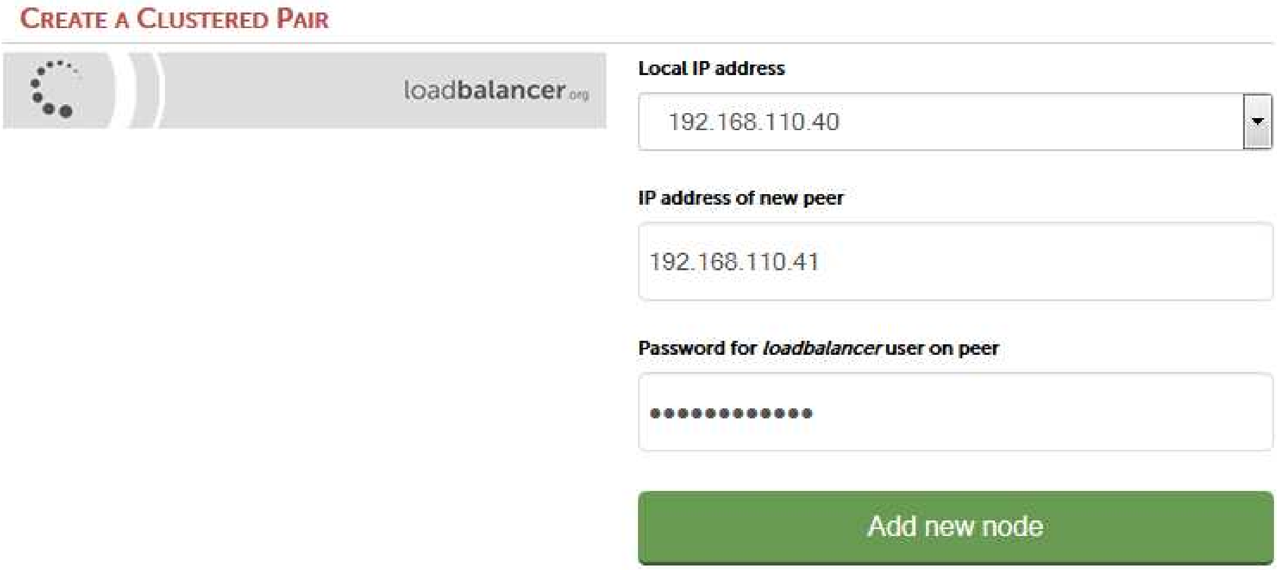
Make sure that if these settings/updates have been configured on the Primary appliance, they're also configured on the Secondary appliance.

To add a Secondary node - i.e. create a highly available clustered pair:

Note

If you have already run the firewall lockdown wizard on either appliance, you'll need to ensure that it is temporarily disabled on both appliances whilst performing the pairing process.

1. Deploy a second appliance that will be the Secondary and configure initial network settings.
2. Using the WebUI on the Primary appliance, navigate to: *Cluster Configuration > High-Availability Configuration*.



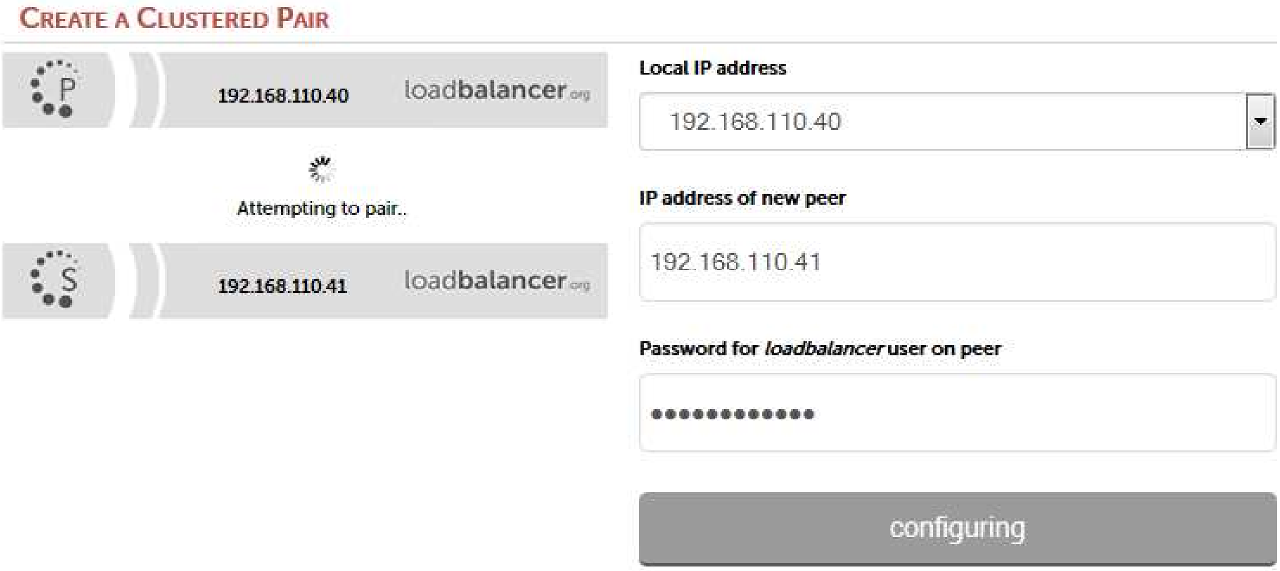
CREATE A CLUSTERED PAIR

Local IP address: 192.168.110.40

IP address of new peer: 192.168.110.41

Password for *loadbalancer* user on peer: ●●●●●●●●●●●●

3. Specify the IP address and the *loadbalancer* user's password for the Secondary (peer) appliance as shown above.
4. Click **Add new node**.
5. The pairing process now commences as shown below:



CREATE A CLUSTERED PAIR

P 192.168.110.40

Attempting to pair..

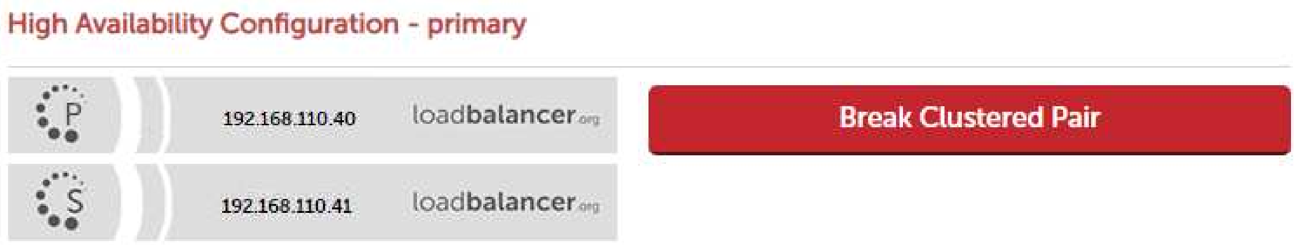
S 192.168.110.41

Local IP address: 192.168.110.40

IP address of new peer: 192.168.110.41

Password for *loadbalancer* user on peer: ●●●●●●●●●●●●

6. Once complete, the following will be displayed on the Primary appliance:


High Availability Configuration - primary

7. To finalize the configuration, restart heartbeat and any other services as prompted in the blue message box at the top of the screen.

Note | Clicking the **Restart Heartbeat** button on the Primary appliance will also automatically restart heartbeat on the Secondary appliance.

Note | For more details on configuring HA with 2 appliances, please refer to Appliance Clustering for HA.

Note | For details on testing and verifying HA, please refer to Clustered Pair Diagnostics.

18. Document Revision History

Version	Date	Change	Reason for Change	Changed By
1.0.0	1 October 2019	Initial version		IG
1.0.1	11 February 2021	New title page Updated Canadian contact details	Branding update Change to Canadian contact details	AH
1.1.0	1 December 2021	Converted the document to AsciiDoc	Move to new documentation system	AH, RJC, ZAC
1.1.1	22 April 2022	Updated SSL related content to reflect latest software version	New software release	RJC

About Loadbalancer.org

Loadbalancer.org's mission is to ensure that its clients' businesses are never interrupted. The load balancer experts ask the right questions to get to the heart of what matters, bringing a depth of understanding to each deployment. Experience enables Loadbalancer.org engineers to design less complex, unbreakable solutions - and to provide exceptional personalized support.

United Kingdom

Loadbalancer.org Ltd.
Compass House, North Harbour
Business Park, Portsmouth, PO6 4PS
UK:+44 (0) 330 380 1064
sales@loadbalancer.org
support@loadbalancer.org

Canada

Loadbalancer.org Appliances Ltd.
300-422 Richards Street, Vancouver,
BC, V6B 2Z4, Canada
TEL:+1 866 998 0508
sales@loadbalancer.org
support@loadbalancer.org

United States

Loadbalancer.org, Inc.
4550 Linden Hill Road, Suite 201
Wilmington, DE 19808, USA
TEL: +1 833.274.2566
sales@loadbalancer.org
support@loadbalancer.org

Germany

Loadbalancer.org GmbH
Tengstraße 2780798,
München, Germany
TEL: +49 (0)89 2000 2179
sales@loadbalancer.org
support@loadbalancer.org