Load Balancing Meditech RESTful API

v1.0.1
## Contents

1. About this Guide .......................................................................................................................... 4
2. Loadbalancer.org Appliances Supported .................................................................................... 4
3. Loadbalancer.org Software Versions Supported ......................................................................... 4
4. Meditech RESTful API Software Versions Supported ............................................................... 4
5. Meditech RESTful API .................................................................................................................. 4
6. Load Balancing Meditech RESTful API ...................................................................................... 4

   Requirements ............................................................................................................................... 5
   API Servers(s) ............................................................................................................................... 5
   Cache Server ............................................................................................................................... 5
   Database Server ........................................................................................................................... 6
   HA Load balancers ....................................................................................................................... 6
   Persistence (aka Server Affinity) .................................................................................................. 6
   Virtual Service (VIP) Requirements ............................................................................................ 6
   Port Requirements ...................................................................................................................... 6
   TLS/SSL Termination ................................................................................................................... 7

7. Deployment Concept ..................................................................................................................... 7

   Scenario 1 – Recommended Deployment Using SSL Offloading ................................................ 7
   Scenario 2 – Recommended Deployment Using SSL Bridging .................................................... 8

8. Load Balancer Deployment Methods .......................................................................................... 8

   Layer 7 SNAT Mode ..................................................................................................................... 9
   Our Recommendation ................................................................................................................ 10

9. Configuring Meditech RESTful API for Load Balancing ............................................................ 11

   DNS Entries ............................................................................................................................... 11

10. Loadbalancer.org Appliance – the Basics .................................................................................. 12

   Virtual Appliance Download & Deployment ............................................................................... 12
   Initial Network Configuration .................................................................................................... 12
   Accessing the Web User Interface (WebUI) ............................................................................... 13
   HA Clustered Pair Configuration ............................................................................................... 14

11. Appliance Configuration for Meditech API – Using Layer 7 SNAT Mode (Scenario 1: Recommended Deployment Using SSL Offloading) ..................................................................... 15

   Configuring the API Virtual Service (VIP1) .............................................................................. 15
   Defining the Real Servers (RIPs) ............................................................................................... 16
   Configuring the Application Virtual Service (VIP2) ................................................................. 16
   Defining the Real Servers (RIPs) ............................................................................................... 17

12. Appliance Configuration for Meditech API – Using Layer 7 SNAT Mode (Scenario 2: Recommended Deployment Using SSL Bridging) ............................................................ 18

   Configuring the API Virtual Service (VIP1) .............................................................................. 18
   Defining the Real Servers (RIPs) ............................................................................................... 19
   Configuring the Application Virtual Service (VIP2) ................................................................. 19
   Defining the Real Servers (RIPs) ............................................................................................... 20
   Setting Up the TLS/SSL Termination ......................................................................................... 21
   Uploading a Certificate ............................................................................................................... 21
   Creating the TLS/SSL Termination (SSL Offloading) .............................................................. 22
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating the TLS/SSL Termination (SSL Bridging)</td>
<td>22</td>
</tr>
<tr>
<td>Finalizing the Configuration</td>
<td>22</td>
</tr>
<tr>
<td>13. Testing &amp; Verification</td>
<td>23</td>
</tr>
<tr>
<td>Using System Overview</td>
<td>23</td>
</tr>
<tr>
<td>14. Technical Support</td>
<td>24</td>
</tr>
<tr>
<td>15. Further Documentation</td>
<td>24</td>
</tr>
<tr>
<td>16. Conclusion</td>
<td>24</td>
</tr>
<tr>
<td>17. Appendix</td>
<td>25</td>
</tr>
<tr>
<td>1 - Clustered Pair Configuration – Adding a Slave Unit</td>
<td>25</td>
</tr>
<tr>
<td>18. Document Revision History</td>
<td>28</td>
</tr>
</tbody>
</table>
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 relevant Administration Manual:

- v7 Administration Manual
- v8 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

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:

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocols</th>
<th>Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>TCP/HTTP</td>
<td>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.</td>
</tr>
<tr>
<td>8081</td>
<td>TCP/HTTPS</td>
<td>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.</td>
</tr>
</tbody>
</table>
**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.

3. **SSL Pass-through** – this 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 = Virtual IP Addresses
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 4 DR
mode is not recommended and cannot be used due to the requirement to terminate TLS/SSL traffic on the load balancer. Since Pound & stunnel both act as a proxy, the real servers see requests with a source IP address of the VIP. However, since the Real Servers believe that they own the VIP (due to the loopback adapter configured to handle to ARP problem) they are unable to reply to Pound.

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 request to the chosen Real Server. As a result, Layer 7 is a slower technique than DR or NAT mode at Layer 4. 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.

This mode can be deployed in a one-arm or two-arm configuration and does not require any changes to the Real Servers. However, since the load balancer is acting as a full proxy it doesn't have the same raw throughput as the layer 4 methods.

The load balancer proxies the application traffic to the servers so that the source of all traffic becomes the load balancer.

- SNAT mode is a full proxy and therefore load balanced Real Servers do not need to be changed in any way
- Because SNAT mode is a full proxy any server in the cluster can be on any accessible subnet including across the Internet or WAN
- 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 balancers 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 clients IP address can be passed through either by enabling TProxy on the load balancer, or for HTTP, using X-forwarded-For headers. Please refer to chapter 6 in the administration manual for more details
- SNAT mode can be deployed using either a 1-arm or 2-arm configuration
Our Recommendation

For simplicity we recommend using layer 7 SNAT mode. This mode requires no changes to the Meditech RESTful API servers and enables the servers to be located on any route-able network.
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.
Virtual Appliance Download & Deployment
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 and XEN and has been optimized for each Hypervisor. By default, the VA is allocated 1 CPU, 2GB of RAM and has an 8GB 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 the Administration Manual and the ReadMe.txt text file included in the VA download for more detailed information on deploying the VA using various Hypervisors.

Initial Network Configuration
The IP address, subnet mask, default gateway and DNS settings can be configured in several ways as detailed below:

Method 1 - Using the Network Setup Wizard at the console
After boot up, follow the instructions on the console to configure the IP address, subnet mask, default gateway and DNS settings.

Method 2 - Using the WebUI
Using a browser, connect to the WebUI on the default IP address/port: https://192.168.2.21:9443
To set the IP address & subnet mask, use: Local Configuration > Network Interface Configuration
To set the default gateway, use: Local Configuration > Routing
To configure DNS settings, use: Local Configuration > Hostname & DNS
Accessing the Web User Interface (WebUI)
The WebUI can be accessed via HTTPS at the following URL: https://192.168.2.21:9443/lbadmin

* Note the port number → 9443

(replace 192.168.2.21 with the IP address of your load balancer if it’s been changed from the default)

Login using the following credentials:

**Username:** loadbalancer

**Password:** loadbalancer

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

Once logged in, the WebUI will be displayed as shown on the following page:
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 slave unit is covered in section 1 of the appendix on page 25.
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
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
5. Click Update
6. Repeat the above steps to add additional MeditechAPI Server(s)
7. Click Reload HAPerxy to commit the configuration

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
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
4. Click Update
5. Repeat the above steps to add your other Meditech_APP Server(s)
6. Click Reload HAProxy to commit the configuration
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
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 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. Click **Update**
5. Repeat the above steps to add your other Meditech_API Server(s)
6. Click **Reload HAProxy** to commit the configuration

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

### Setting Up the 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
6. Click **Upload certificate**
Further information on creating PEM files and converting between certificate formats can be found in our Administration Manual.

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. Instructions on creating a CSR can be found in our Administration Manual.

Our Administration Manual can be found at: http://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf

Creating The TLS/SSL Termination (SSL Offloading)
1. Using the web user interface, navigate to Cluster Configuration > SSL Termination and click on Add a new Virtual Service
2. From the Associated Virtual Service drop-down list, select the MeditechAPI service which was created previously
3. Set the Virtual Service Port field to 443
4. From the SSL Certificate drop-down list, select the appropriate certificate
5. Click Update to create the TLS/SSL termination service
6. Click Reload HAProxy and Reload Stunnel to commit the configuration
7. Repeat the above steps to configure TLS/SSL termination for your other Meditech_API virtual service

Creating The TLS/SSL Termination (SSL Bridging)
1. To configure SSL bridging (SSL with re-encryption to the backend) follow the steps as per section 12 of this guide and enable re-encrypt to backend on the VIP
2. Ensure that the real servers have port 443 defined with an SSL certificate installed
3. Next upload an SSL certificate to the load balancer as per the step above
4. Then create an SSL termination as per the steps above
5. The system overview page will display real servers with re-encrypt to backend padlock icon

Finalizing the Configuration
To apply the new settings, HAProxy and stunnel must both be reloaded as follows:
1. Using the WebUI, navigate to: Maintenance > Restart Services and click **Reload STunnel**
2. Using the WebUI, navigate to: Maintenance > Restart Services and click **Reload HAProxy**

13. Testing & Verification

**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:
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

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

1 - Clustered Pair Configuration – Adding a Slave Unit
If you initially configured just the master unit and now need to add a slave - our recommended procedure, please refer to the relevant section below for more details:

Note: A number of settings are not replicated as part of the master/slave pairing process and therefore must be manually configured on the slave appliance. These are listed below:

- Hostname & DNS settings
- Network settings including IP addresses, bonding configuration and VLANs
- Routing configuration including default gateways and static routes
- Date & time settings
- Physical – Advanced Configuration settings including Internet Proxy IP address & port, Firewall table size, SMTP relay and Syslog server
- SNMP settings
- Graphing settings
- Firewall Script & Firewall Lockdown Script settings
- Software updates

Version 7:

Please refer to Chapter 8 – Appliance Clustering for HA in the v7 Administration Manual.

Version 8:

To add a slave node – i.e. create a highly available clustered pair:

- Deploy a second appliance that will be the slave and configure initial network settings
- Using the WebUI, navigate to: Cluster Configuration > High-Availability Configuration
• Specify the IP address and the `loadbalancer` users password (the default is 'loadbalancer') for the slave (peer) appliance as shown above.

• Click `Add new node`.

• The pairing process now commences as shown below:

• Once complete, the following will be displayed:

• 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 master appliance will also automatically restart heartbeat on the slave appliance.

Note: Please refer to chapter 9 – Appliance Clustering for HA in the Administration Manual for more detailed information on configuring HA with 2 appliances.
18. Document Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change</th>
<th>Reason for Change</th>
<th>Changed By</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>1 October 2019</td>
<td>Initial version</td>
<td></td>
<td>IG</td>
</tr>
<tr>
<td>1.0.1</td>
<td>11 February 2021</td>
<td>New title page</td>
<td>Branding update</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Canadian contact details</td>
<td>Change to Canadian contact details</td>
<td></td>
</tr>
</tbody>
</table>
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

© Copyright Loadbalancer.org • www.loadbalancer.org