1. About this Guide
This guide details the steps required to configure a load balanced Evertz Mediator-X environment utilizing Loadbalancer.org appliances. It covers the configuration of the load balancers and also any Evertz Mediator-X 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 for load balancing Evertz Mediator-X. The complete list of models is shown below:

<table>
<thead>
<tr>
<th>Discontinued Models</th>
<th>Current Models *</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise R16</td>
<td>Enterprise R20</td>
</tr>
<tr>
<td>Enterprise VA R16</td>
<td>Enterprise MAX</td>
</tr>
<tr>
<td>Enterprise VA</td>
<td>Enterprise 10G</td>
</tr>
<tr>
<td>Enterprise R320</td>
<td>Enterprise 40G</td>
</tr>
<tr>
<td></td>
<td>Enterprise Ultra</td>
</tr>
<tr>
<td></td>
<td>Enterprise VA R20</td>
</tr>
<tr>
<td></td>
<td>Enterprise VA MAX</td>
</tr>
<tr>
<td></td>
<td>Enterprise AWS **</td>
</tr>
<tr>
<td></td>
<td>Enterprise AZURE **</td>
</tr>
</tbody>
</table>

* For full specifications of these models please refer to: [http://www.loadbalancer.org/products/hardware](http://www.loadbalancer.org/products/hardware)

** Some features may not be supported, please check with Loadbalancer.org support

3. Loadbalancer.org Software Versions Supported

• V8.3.6 and later

4. Evertz Mediator Software Versions Supported

• Evertz Mediator-X
5. Evertz Mediator-X

Evertz Mediator-X unifies content acquisition, content processing, media management, production, playout, and delivery into a single, integrated environment. The unification of these services on a single platform delivers optimized media workflows and increased operational efficiency.

Built on over fifteen years of Mediator product development and deployment expertise, Mediator-X has a modern, scalable, infrastructure-agnostic architecture which can be deployed in public cloud, private cloud, or hybrid environments, enabling users to be flexible with their deployment strategies and to grow the platform wherever the business case dictates.

Evertz recommends Loadbalancer.org appliances to provide high availability and load balancing of the Mediator-X platform.

6. Load Balancing Evertz Mediator-X

Note: It's highly recommended that you have a working Evertz Mediator-X environment first before implementing the load balancer.

Sizing, Capacity, and Performance for a Virtual Load Balancer Deployment

The Loadbalancer.org appliances can be deployed as virtual appliances.

For small deployments handling up to 300 concurrent connections/users, your virtual host should be allocated a minimum of 4 vCPUs, 4 GB of RAM, and 8 GB of disk storage.

For large deployments handling over 300 concurrent connections/users, your virtual host should be allocated a minimum of 8 vCPUs, 8 GB of RAM, and 8 GB of disk storage.

For significantly larger deployments, your Evertz representative will give you custom sizing and resource guidelines based on the expected load on your load balancers and your predicted usage profile.

Persistence (aka Server Affinity)

For the layer 4 DR mode scenario, each virtual service uses source IP address based persistence.

For the layer 7 load balancing scenario (the configuration that adds TLS based encryption), the persistence mode X-Forwarded-For and Source IP is used. This uses X-Forwarded-For HTTP headers as the primary persistence method, with source IP addresses used as a backup persistence method.

Virtual Service (VIP) Requirements

To provide load balancing and HA for Evertz Mediator-X, the following VIP is required:

- Mediator-X Global Access

The “Global” virtual service handles Mediator-X user interface traffic and API endpoint traffic. “Global” access to both services is provided using a single virtual service on the load balancer.

Additionally, a TLS/SSL termination service is required for the scenario that adds TLS based encryption.
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>Mediator-X user interface access, Mediator-X API endpoint access</td>
</tr>
<tr>
<td>443</td>
<td>TCP/HTTPS</td>
<td>Mediator-X user interface access over TLS (optional)</td>
</tr>
</tbody>
</table>

TLS/SSL Termination
It is possible to configure a TLS/SSL termination service in front of the plaintext, port 80, HTTP based Mediator-Global service. This enables inbound client connections to be secured using TLS. Connections from the load balancer to the Mediator-X servers remain as plaintext HTTP connections (not encrypted) on port 80. In this way, inbound client connections can be secured using encryption without needing to make any changes to the back end Mediator-X servers.

7. Deployment Concept
Evertz Mediator-X can be load balanced in two different ways:

- Simple deployment: Uses a single virtual service to load balance all of the port 80 traffic used by Mediator-X (the user interface traffic as well as the API endpoint traffic)

- Deployment using TLS based encryption: An alternative deployment type that should only be used when there is the requirement to secure client connections using TLS based encryption. Using this deployment type, clients can connect to the Mediator-X User Interface using HTTPS on port 443

Scenario 1 – Simple Deployment

VIPs = Virtual 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 section 1 in the appendix on page 21 for more details on configuring a clustered pair.

In this deployment, a single virtual service is used. The virtual service uses layer 4 DR mode, offering the greatest possible network speed and scalability.

Layer 4 DR mode is the load balancing method that has traditionally been used with Evertz Mediator deployments.

**Scenario 2 – Deployment Using TLS Based Encryption**

In this deployment, one virtual service is used in addition to a TLS/SSL termination. The virtual service uses layer 7 SNAT mode.

This alternative deployment type allows for Mediator-X traffic to be secured using TLS, with clients sending encrypted traffic on port 443 instead of plaintext traffic on port 80.

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 Mediator-X, using layer 4 DR mode is recommended due to its raw throughput and huge scalability. It is also possible to use layer 7 SNAT mode, which allows adding TLS based encryption for client traffic, however the performance of this set
up is not as great as layer 4 DR mode. These load balancing modes are described below and are used for the configurations presented in this guide. For configuring using DR mode please refer to the section starting on page 14, and for configuring using layer 7 SNAT mode, which allows adding TLS based encryption, refer to the section starting on page 16.

Layer 4 DR Mode

One-arm direct routing (DR) mode is a very high performance solution that requires little change to your existing infrastructure.

Note: Kemp, Brocade, Barracuda & A10 Networks call this Direct Server Return and F5 call it N-Path.

- DR mode works by changing the destination MAC address of the incoming packet to match the selected Real Server on the fly which is very fast
- When the packet reaches the Real Server it expects the Real Server to own the Virtual Services IP address (VIP). This means that you need to ensure that the Real Server (and the load balanced application) respond to both the Real Servers own IP address and the VIP
- The Real Server should not respond to ARP requests for the VIP. Only the load balancer should do this. Configuring the Real Servers in this way is referred to as Solving the ARP Problem. This is handled automatically across all nodes in a Mediator-X deployment by following the instructions described in section Configuring Evertz Mediator-X for Load Balancing
- On average, DR mode is 8 times quicker than NAT for HTTP, 50 times quicker for Terminal Services and much, much faster for streaming media or FTP
- The load balancer must have an interface in the same subnet as the Real Servers to ensure layer 2 connectivity required for DR mode to work
- The VIP can be brought up on the same subnet as the Real Servers, or on a different subnet provided that the load balancer has an interface in that subnet
- Port translation is not possible in DR mode i.e. having a different RIP port than the VIP port
- DR mode is transparent, i.e. the Real Server will see the source IP address of the client
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 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 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

© Copyright Loadbalancer.org • www.loadbalancer.org • sales@loadbalancer.org
Our Recommendation

Where possible, we recommend that Layer 4 Direct Routing (DR) mode is used. This mode offers the best possible performance since replies go directly from the Real Servers to the client, not via the load balancer. It's also relatively simple to implement. Ultimately, the final choice does depend on your specific requirements and infrastructure.

If DR mode cannot be used, for example if the real servers are located in remote routed networks, then SNAT mode is recommended. SNAT is also recommended if TLS based encryption is required for the HTTP aspect of the Mediator-X inbound client traffic.

If the load balancer is deployed in AWS or Azure, layer 7 SNAT mode must be used as layer 4 direct routing is not currently possible on these platforms.

9. Configuring Evertz Mediator-X for Load Balancing

Some changes must be made to the Mediator-X real servers in order for them to be correctly load balanced. These changes need to be configured by an Evertz Deployment Team. Contact your Evertz representative for further information.

1. On the Mediator-X deployment, connect to Node0 and log in as the root user. This can be done by executing the command `sudo su` and then entering the system specific shell access password

2. Navigate to the directory `/srv/salt/pillar`. This can be done by executing the command `cd /srv/salt/pillar`

3. Edit the file `system.sls`, for example using a text editor such as nano or vim: `nano system.sls`

4. Find the `virtual_ips` parameter and add the virtual IP address that will be used for the load balanced deployment. This is the user facing IP address that all clients will connect to when accessing the load balanced Mediator-X services. The result should look like the following:

```
# Set the loadbalancer virtual ip or leave blank
virtual_ips : 10.0.1.50
```

5. Save and exit the `system.sls` file

6. Run the salt command and call the `state.highstate` function, which will automatically apply the changed configuration across all Mediator-X nodes. The full command to execute is:

```
sudo salt "*" state.highstate
```

7. Run the salt command and call a function to restart the nginx service across all Mediator-X nodes. The full command to execute is:

```
sudo salt -G 'is_mediatorx:True' service.restart nginx
```
10. Loadbalancer.org Appliance – the Basics

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 21.
11. Appliance Configuration for Evertz Mediator-X – Using Layer 4 DR Mode (Scenario 1: Simple Deployment)

Configuring the Virtual Service (VIP)

1. Using the web user interface, navigate to Cluster Configuration > Layer 4 - Virtual Services and click on Add a new Virtual Service.
2. Define the Label for the virtual service as required, e.g. Mediator-Global.
3. Set the Virtual Service IP Address field to the required IP address, e.g. 192.168.85.140.
4. Set the Ports field to 80.
5. Leave the Protocol set to TCP.
7. Click Update to create the virtual service.

8. Click Modify next to the newly created VIP.
9. Set the Balance Mode to Weighted Round Robin.
10. Ensure that the Persistence Enable checkbox is checked and that the Timeout is set to 300 (this should already be configured by default).
11. Set the Health Checks Check Type to Negotiate.
12. Set the Check Port to 80.
13. Set the Protocol to HTTP.
14. Set the Request to send to /mediator/main/loadBalancing/isExternallyAccessibleAPI.
15. Ensure that the Response expected field is blank.
16. Click Update.
Defining the Real Servers (RIPs)

1. Using the web user interface, navigate to Cluster Configuration > Layer 4 – Real Servers and click on Add a new Real Server next to the newly created VIP.
2. Define the Label for the real server as required, e.g. node-04.
3. Set the Real Server IP Address field to the required IP address, e.g. 192.168.85.24.
4. Click Update.
5. Repeat these steps to add additional Mediator-X servers as required.
Enabling Multithreaded Load Balancing
For the layer 7 load balancing scenario, the Loadbalancer.org appliance should be configured to actively use multiple CPU cores for the load balancing process. This is required to achieve the high level of performance and throughput required when load balancing a Mediator-X deployment at layer 7.

Note: A virtual host should be allocated a minimum of 4 vCPUs.

To enable multithreaded mode from the WebUI:

1. Navigate to Cluster Configuration > Layer 7 - Advanced Configuration
2. Check the Enable Multithreading checkbox
3. Set Number of Threads as appropriate, e.g. 3

Note: We recommend using a number of threads equal to ‘the number of CPU cores - 1’. For example: on a 4 core CPU system, 3 threads are recommended; on an 8 core CPU system, 7 threads are recommended.

4. Click Update to apply the changes

Configuring the Virtual Service (VIP)

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. Mediator-Global
3. Set the Virtual Service IP Address field to the required IP address, e.g. 192.168.85.140
4. Set the Virtual Service Ports field to 80
5. Set the Layer 7 Protocol to HTTP Mode
6. Click Update to create the virtual service
7. Click **Modify** next to the newly created VIP.
8. Set the **Balance Mode** to **Weighted Round Robin**.
9. Set **Persistence Mode** to **X-Forwarded-For and Source IP**.
10. Click the **Persistence Advanced** button to expand the menu.
11. Set **Persistence Timeout** to **5**.
12. Set **Health Checks** to **Negotiate HTTP (HEAD)**.
13. Set the **Request to send** to `/mediator/main/loadBalancing/isExternallyAccessibleAPI`.

In the **Layer 7 - Add a new Virtual Service** section, set the **Label** to **Mediator-Global**, **IP Address** to **192.168.85.140**, **Ports** to **80**, **Layer 7 Protocol** to **HTTP Mode**, and the **Manual Configuration** to off.

For **Connection Distribution Method**, set the **Balance Mode** to **Weighted Round Robin**.

For **Protocol**, set the **Layer 7 Protocol** to **HTTP Mode**.

For **Persistence**, set the **Persistence Mode** to **X-Forwarded-For and Source IP**.

For **Persistence**, set the **Timeout** to **5**, **Table size** to **10240**, **XFF IP Position** to **-1**, and **Clear Stick on Drain** to off.

For **Health Checks**, set **Negotiate HTTP (HEAD)**.

For **Request to send**, set `/mediator/main/loadBalancing/isExternallyAccessibleAPI`.
14. Click **Update**

### Defining the Real Servers (RIPs)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Real Servers* and click on **Add a new Real Server** next to the newly created VIP
2. Enter an appropriate name for the server in the *Label* field, e.g. node-04
3. Change the *Real Server IP Address* field to the required IP address, e.g. 192.168.85.24
4. Set the *Real Server Port* field to 80
5. Click **Update**
6. Repeat these steps to add additional servers as required

![Layer 7 Add a new Real Server - Mediator-Global](image)

### 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. Mediator-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](http://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf)

**Creating The TLS/SSL Termination**

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 **Mediator-Global** 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

**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 Mediator-X 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 four Mediator-X nodes are healthy and available to accept connections:

![System Overview](image)

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 Evertz Mediator-X 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>30 May 2019</td>
<td>Initial version</td>
<td></td>
<td>AH</td>
</tr>
<tr>
<td>1.0.1</td>
<td>13 June 2019</td>
<td>Added section on enabling HAProxy multi-threaded mode</td>
<td>Required updates</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added a comment that the changes to the Mediator servers should be carried out by Evertz engineers</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changed some terminology at the request of Evertz</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changed the diagrams to reflect the new simplified configurations</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changed the instructions and screenshots to reflect the new single virtual service configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.0</td>
<td>24 July 2019</td>
<td>Styling and layout</td>
<td>General styling updates</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changed the health checks at the request of Evertz</td>
<td>Required updates</td>
<td></td>
</tr>
<tr>
<td>1.1.1</td>
<td>1 August 2019</td>
<td>Made changes to section “Configuring Evertz Mediator-X for Load Balancing” at the request of Evertz</td>
<td>Required updates</td>
<td>AH</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.