LOADBALANCER

DEPLOYMENT BRIEF

Load Balancing Wowza Streaming Engine

Version 1.1.0

Table of Contents

1. About this Brief

This brief outlines the steps required to configure a load balanced Wowza Streaming Engine environment utilizing Loadbalancer.org appliances. It covers the configuration of the load balancers and also any Wowza Streaming Engine 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 Wowza Streaming Engine. For full specifications of available models please refer to https://www.loadbalancer.org/products/enterprise.

Some features may not be available or fully supported in all cloud platforms due to platform specific limitations. For more details, please refer to the "Main Differences to our Standard (Non-Cloud) Product" section in the appropriate cloud platform Quick Start Guide or check with Loadbalancer.org support.

3. Software Versions Supported

3.1. Loadbalancer.org Appliance

- V8.9.1 and later

> **Note** The screenshots used throughout this document aim to track the latest Loadbalancer.org software version. If you're using an older version, or the very latest, the screenshots presented here may not match your WebUI exactly.

3.2. Wowza Streaming Engine

- All versions

4. Wowza Streaming Engine

Wowza Streaming Engine, formerly known as ***Wowza Media Server***, is a unified streaming media server software from Wowza. The media server is used to stream both live and on-demand audio and video over IP networks to desktop, laptop, and tablet computers, mobile devices, IPTV set-top boxes, internet-connected TV sets, game consoles, and other network-connected devices. The media server itself is a Java application which makes it flexible: it can be deployed on most operating systems.

5. Load Balancing Wowza Streaming Engine

> **Note** It's highly recommended that you have a working Wowza Streaming Engine environment first before implementing the load balancer.

5.1. Persistence (aka Server Affinity)

Source IP address-based session affinity is required when load balancing Wowza Streaming Engine servers.

5.2. Virtual Service (VIP) Requirements

To provide load balancing and HA for Wowza Streaming Engine, the following VIPs are required:

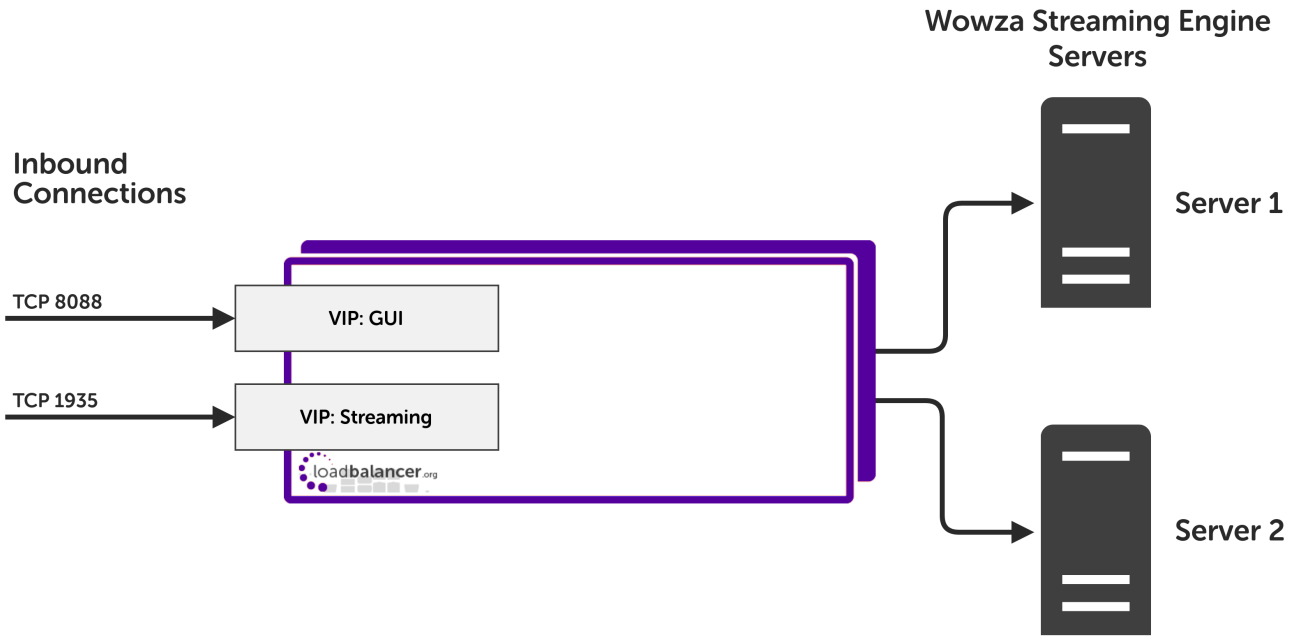
- GUI
- Streaming

5.3. Port Requirements

The following table shows the ports that are load balanced:

Port	Protocols	Use
1935	TCP/RTMP	Wowza Streaming Access
8088	TCP/HTTP	Wowza GUI Access

6. Deployment Concept



VIP = **V**irtual **IP** Address

> **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 the section Configuring HA - Adding a Secondary Appliance in the appendix for more details on configuring a clustered pair.

7. 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 Wowza Streaming Engine, using layer 4 DR mode is recommended. It is also possible to use layer 7 SNAT mode. These modes are described below and are used for the configurations presented in this guide.

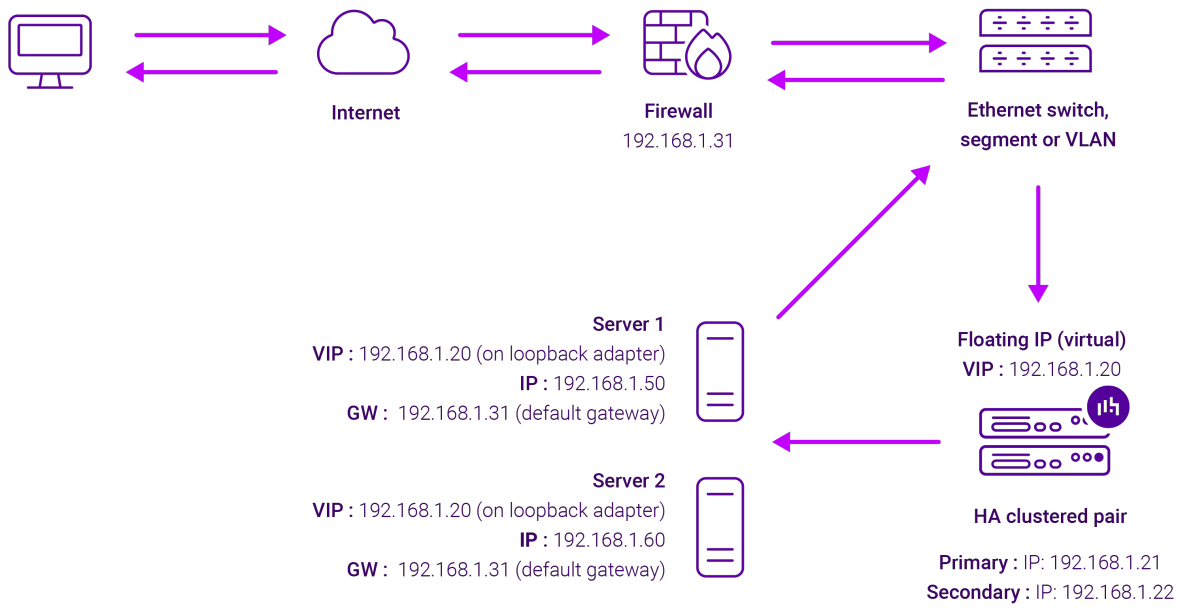
For configuring using DR mode please refer to Section 10, "Appliance Configuration for Wowza Streaming Engine – Using Layer 4 DR Mode".

For configuring using layer 7 SNAT mode please refer to Section 11, "Appliance Configuration for Wowza Streaming Engine – Using Layer 7 SNAT Mode".

7.1. Layer 4 DR Mode

Layer 4 DR (Direct Routing) mode is a very high performance solution that requires little change to your existing infrastructure. The image below shows an example network diagram for this mode.

> **Note** Kemp, Brocade, Barracuda & A10 Networks call this ***Direct Server Return*** and F5 call it ***nPath***.

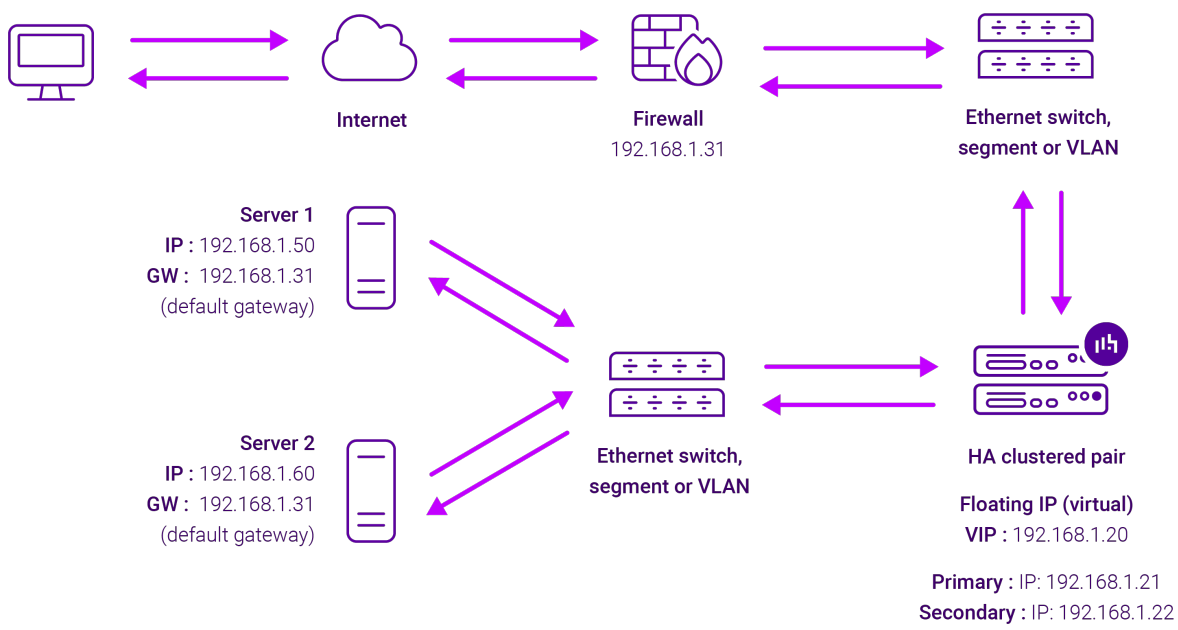


- DR mode works by changing the destination MAC address of the incoming packet to match the selected Real Server on the fly which is very fast.
- When the packet reaches the Real Server it expects the Real Server to own the Virtual Services IP address (VIP). This means that each Real Server (and the load balanced application) must respond to both the Real Server's 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 Server in this way is referred to as "Solving the ARP Problem". For more information please refer to DR Mode Considerations.
- On average, DR mode is 8 times quicker than NAT mode for HTTP and much faster for other applications such as Remote Desktop Services, streaming media and FTP.

- The load balancer must have an interface in the same subnet as the Real Servers to ensure layer 2 connectivity which is required for DR mode to operate.
- 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 with DR mode, e.g. VIP:80 → RIP:8080 is not supported.
- DR mode is transparent, i.e. the Real Server will see the source IP address of the client.

7.2. 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. The image below shows an example network diagram for this mode.



- Because layer 7 SNAT mode is a full proxy, Real Servers in the cluster can be on any accessible network 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, **eth1** is typically used for client side connections and **eth0** is used for Real Server connections, although this is not mandatory since any interface can be used for any purpose.

- Requires no mode-specific 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.

8. Configuring Wowza Streaming Engine for Load Balancing

The configuration steps required on the Wowza Streaming Engine servers depend on the load balancing mode used, as described below.

8.1. Layer 4 DR Mode

The "ARP problem" must be solved on each Wowza Streaming Engine server for DR mode to work. For detailed steps on solving the ARP problem, refer to the appendix Solving the ARP Problem for Linux.

8.2. Layer 7 SNAT Mode

When using SNAT mode, no mode-specific configuration changes to the Wowza Streaming Engine servers are required.

9. Loadbalancer.org Appliance – the Basics

9.1. 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.

9.2. Initial Network Configuration

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

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

9.3. Accessing the Appliance WebUI

The WebUI is accessed using a web browser. By default, users are authenticated using Apache authentication. Users can also be authenticated against LDAP, LDAPS, Active Directory or Radius - for more information, please refer to External Authentication.

> **Note** There are certain differences when accessing the WebUI for the cloud appliances. For details, please refer to the relevant Quick Start / Configuration Guide.

1. Using a browser, navigate to the following URL:

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

   > **Note** You'll receive a warning about the WebUI's SSL certificate. This is due to the default self signed certificate that is used. If preferred, you can upload your own certificate - for more information, please refer to Appliance Security Features.

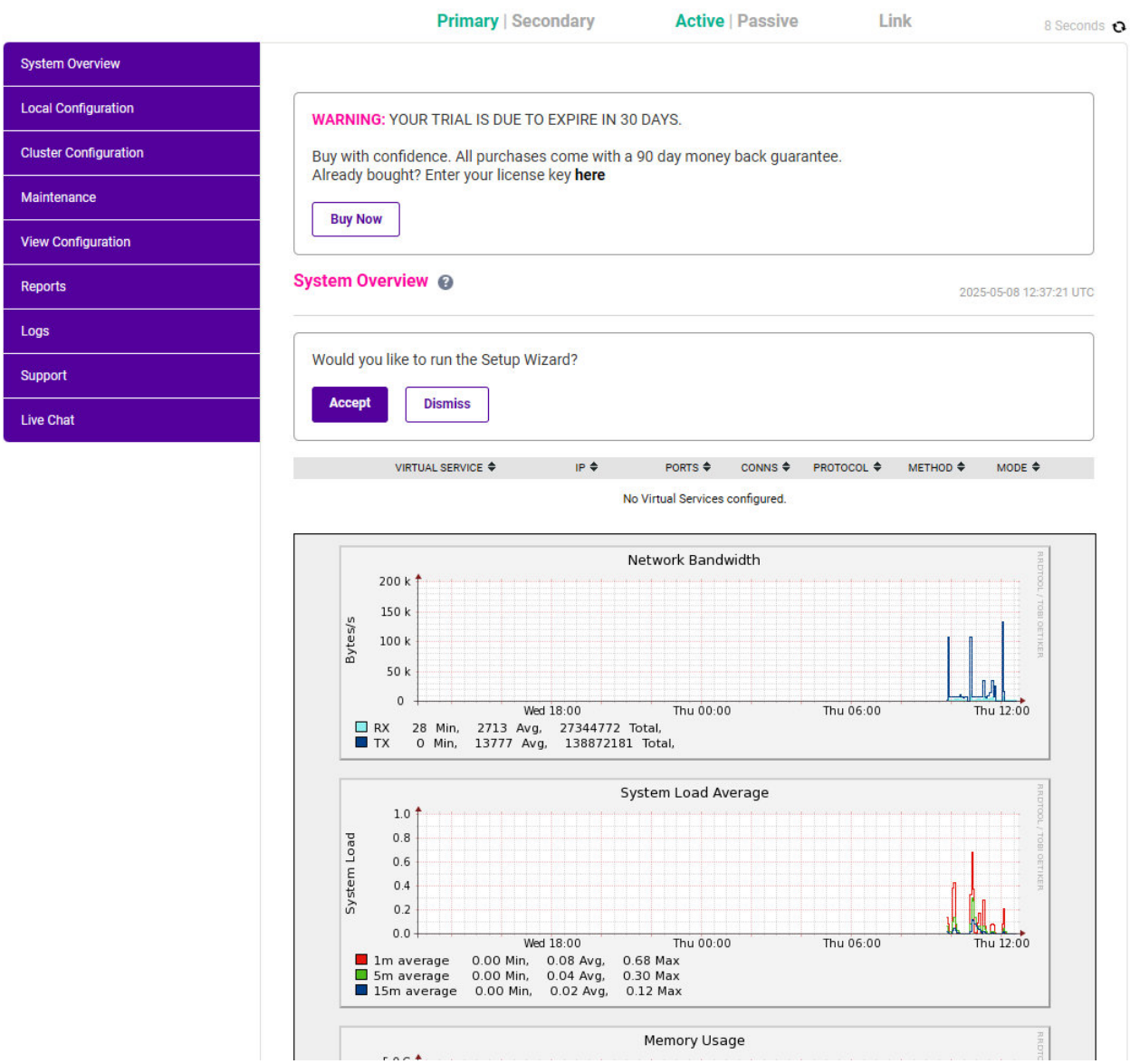
   > **Note** If you need to change the port, IP address or protocol that the WebUI listens on, please refer to Service Socket Addresses.

2. Log in to the WebUI using the following credentials:

   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:



3. You'll be asked if you want to run the Setup Wizard. Click **Dismiss** if you're following a guide or want to configure the appliance manually. Click **Accept** to start the Setup Wizard.

> **Note** The Setup Wizard can only be used to configure Layer 7 services.

9.3.1. 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 creating 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

9.4. Appliance Software Update

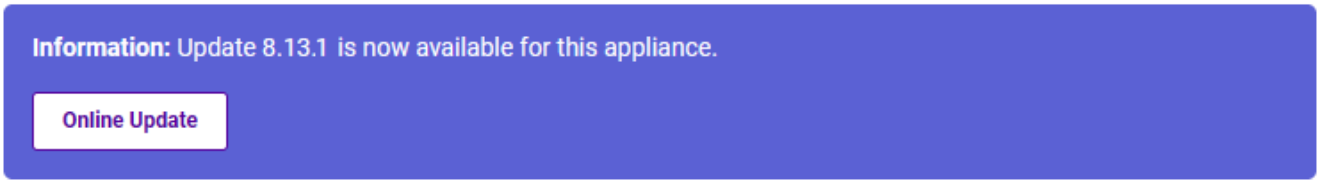
We recommend that the appliance is kept up to date to ensure that you benefit from the latest bug fixes, security updates and feature improvements. Both online and offline update are supported.

> **Note** For full details, please refer to Appliance Software Update in the Administration Manual.

> **Note** Services may need to be restarted/reloaded after the update process completes or in some cases a full appliance restart may be required. We therefore recommend performing the update during a maintenance window.

9.4.1. Online Update

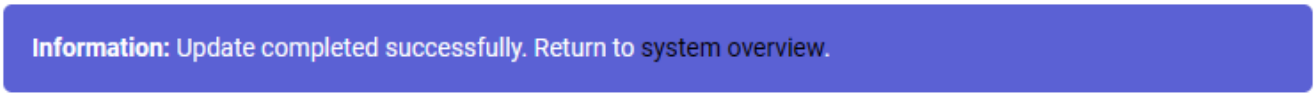
The appliance periodically contacts the Loadbalancer.org update server (**update.loadbalancer.org**) and checks for updates. This is the default behavior and can be disabled if preferred. If an update is found, a notification similar to the example below will be displayed at the top of the WebUI:



Click **Online Update**. A summary of all new features, improvements, bug fixes and security updates included in the update will be displayed. Click **Update** at the bottom of the page to start the update process.

> **Important** Do not navigate away whilst the update is ongoing, this may cause the update to fail.

The update can take several minutes depending on download speed and upgrade version. Once complete, the following message will be displayed:



If services need to be reloaded/restarted or the appliance needs a full restart, you'll be prompted accordingly.

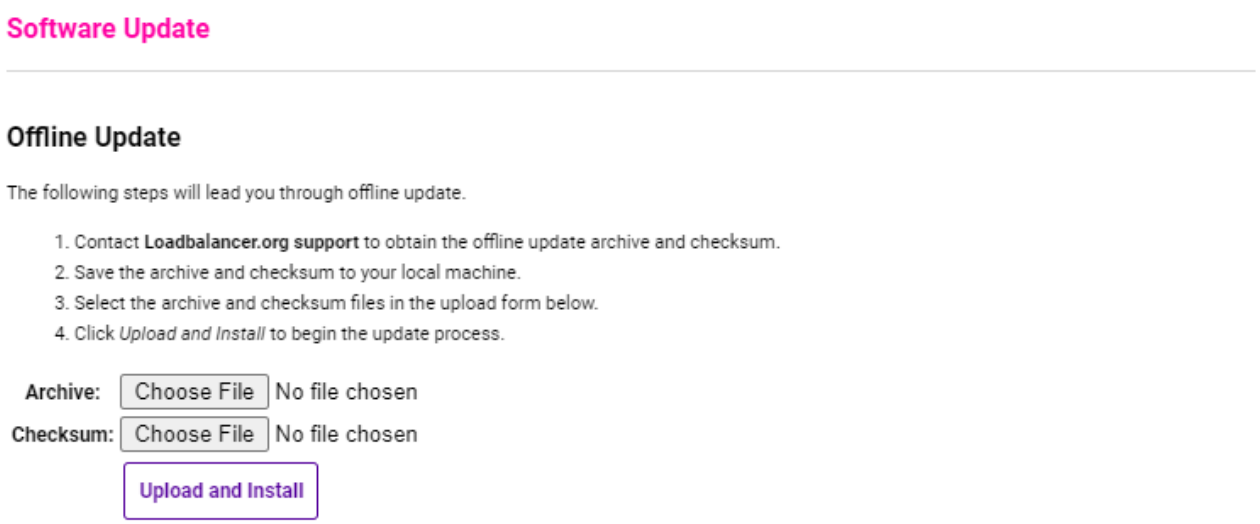
9.4.2. Offline Update

If the appliance does not have access to the Internet, offline update can be used.

To check for the latest version, please refer to our product roadmap page available here. To obtain the latest offline update files contact support@loadbalancer.org.

To perform an offline update:

1. Using the WebUI, navigate to: ***Maintenance > Software Update***.
2. Select **Offline Update**.
3. The following screen will be displayed:



4. Select the ***Archive*** and ***Checksum*** files.
5. Click **Upload and Install**.
6. If services need to be reloaded/restarted or the appliance needs a full restart, you'll be prompted accordingly.

9.5. Ports Used by the Appliance

By default, the appliance uses the following TCP & UDP ports:

Protocol	Port	Purpose
TCP	22 *	SSH
TCP & UDP	53 *	DNS / GSLB
TCP & UDP	123	NTP
TCP & UDP	161 *	SNMP
UDP	6694	Heartbeat between Primary & Secondary appliances in HA mode
TCP	7778	HAProxy persistence table replication
TCP	9000 *	Gateway service (Centralized/Portal Management)
TCP	9080 *	WebUI - HTTP (disabled by default)
TCP	9081 *	Nginx fallback page
TCP	9443 *	WebUI - HTTPS
TCP	25565 *	Shuttle service (Centralized/Portal Management)

> **Note** The ports used for SSH, GSLB, SNMP, the WebUI, the fallback page, the gateway service and the shuttle service can be changed if required. For more information, please refer to Service Socket

Addresses.

9.6. HA Clustered Pair Configuration

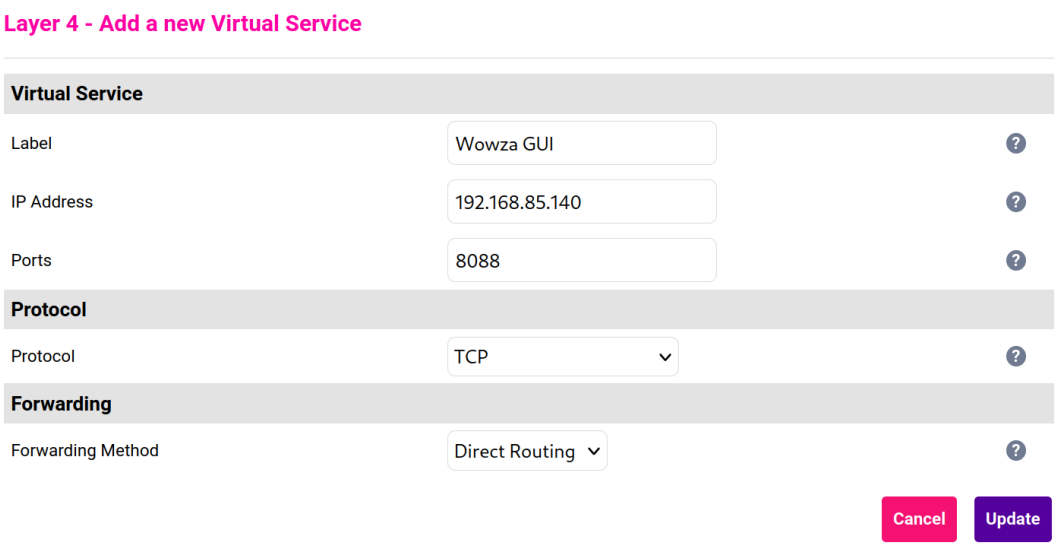
Loadbalancer.org recommend that load balancer appliances are deployed in pairs for high availability. In this guide a single unit is deployed first, adding a secondary unit is covered in the section Configuring HA - Adding a Secondary Appliance of the appendix.

10. Appliance Configuration for Wowza Streaming Engine – Using Layer 4 DR Mode

10.1. Configuring VIP 1 - Wowza GUI

10.1.1. Configuring the Virtual Service (VIP)

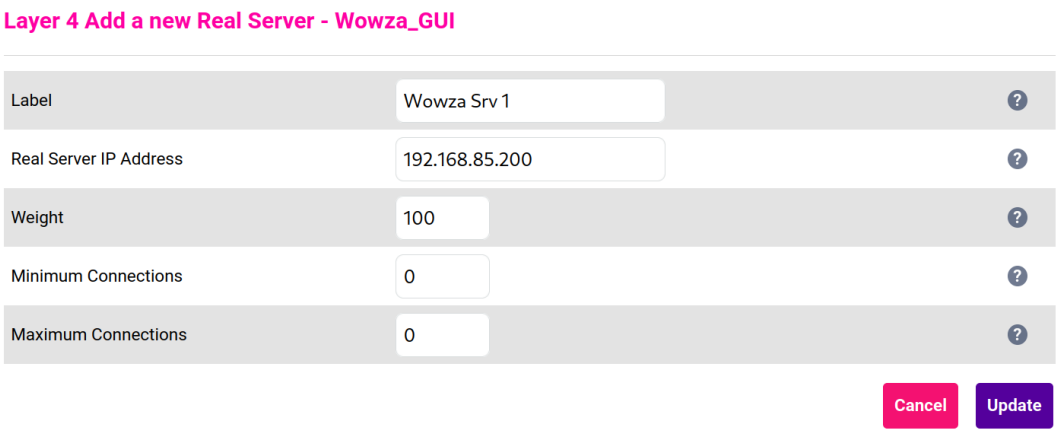
1. Using the web user interface, navigate to *Cluster Configuration > Layer 4 – Virtual Services* and click on **Add a new Virtual Service**.
2. Define the *Label* for the virtual service as required, e.g. **Wowza GUI**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.85.140**.
4. Set the *Ports* field to **8088**.
5. Set the *Protocol* to **TCP**.
6. Leave the *Forwarding Method* set to **Direct Routing**.
7. Click **Update** to create the virtual service.



10.1.2. 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. **Wowza Srv 1**.

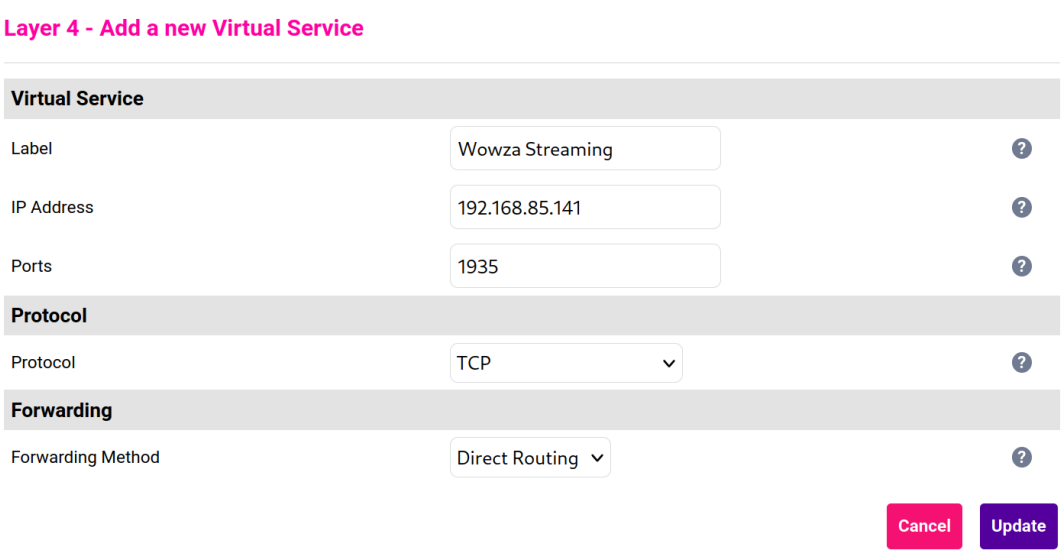
3. Set the *Real Server IP Address* field to the required IP address, e.g. **192.168.85.200**.
4. Click **Update**.
5. Repeat these steps to add additional Wowza Streaming Engine servers as required.



10.2. Configuring VIP 2 - Wowza Streaming

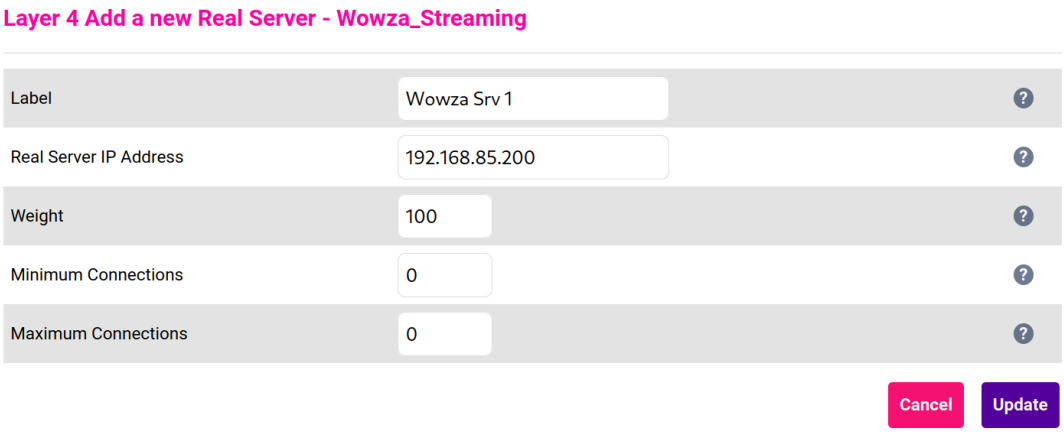
10.2.1. Configuring the Virtual Service (VIP)

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. **Wowza Streaming**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.85.141**.
4. Set the *Ports* field to **1935**.
5. Set the *Protocol* to **TCP**.
6. Leave the *Forwarding Method* set to **Direct Routing**.
7. Click **Update** to create the virtual service.



10.2.2. 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. **Wowza Srv 1**.
3. Set the *Real Server IP Address* field to the required IP address, e.g. **192.168.85.200**.
4. Click **Update**.
5. Repeat these steps to add additional Wowza Streaming Engine servers as required.



10.3. Wowza Streaming Engine Server Configuration Steps

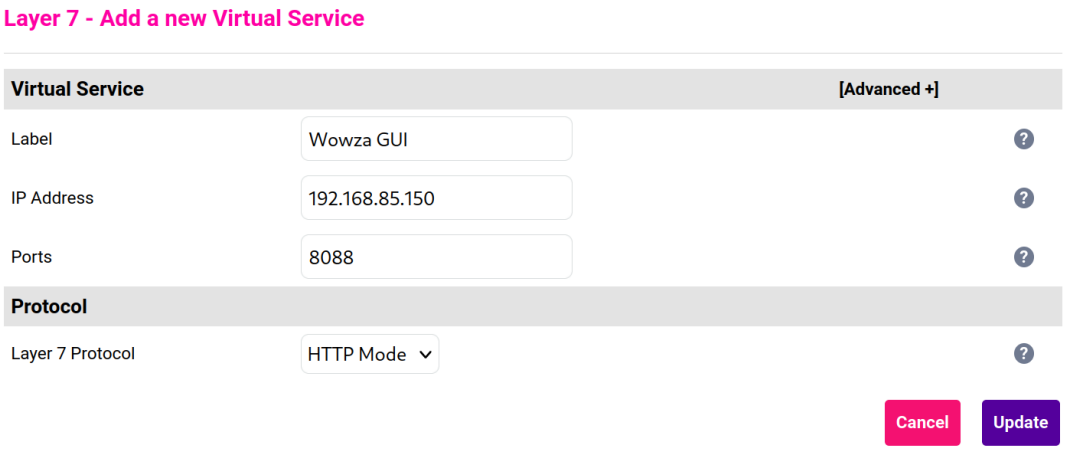
When using layer 4 DR mode, as mentioned earlier in Layer 4 DR Mode, the "ARP problem" must be solved on each Wowza Streaming Engine server. For full details of the steps required to do this, refer to Solving the ARP Problem for Linux.

11. Appliance Configuration for Wowza Streaming Engine – Using Layer 7 SNAT Mode

11.1. Configuring VIP 1 - Wowza GUI

11.1.1. Configuring the Virtual Service (VIP)

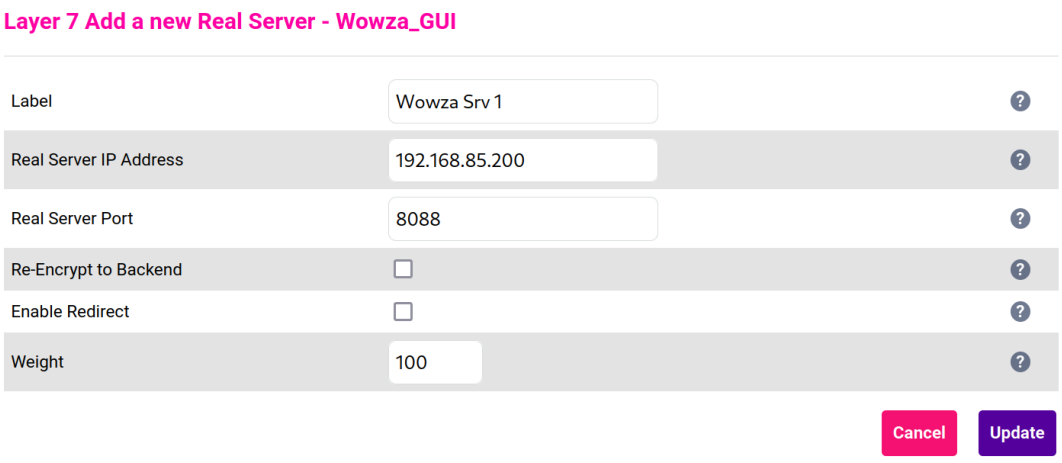
1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click on **Add a new Virtual Service**.
2. Define the *Label* for the virtual service as required, e.g. **Wowza GUI**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.85.150**.
4. Set the *Ports* field to **8088**.
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 *Persistence Mode* to **Source IP**.
9. Click **Update**.

11.1.2. Defining the Real Servers (RIPs)

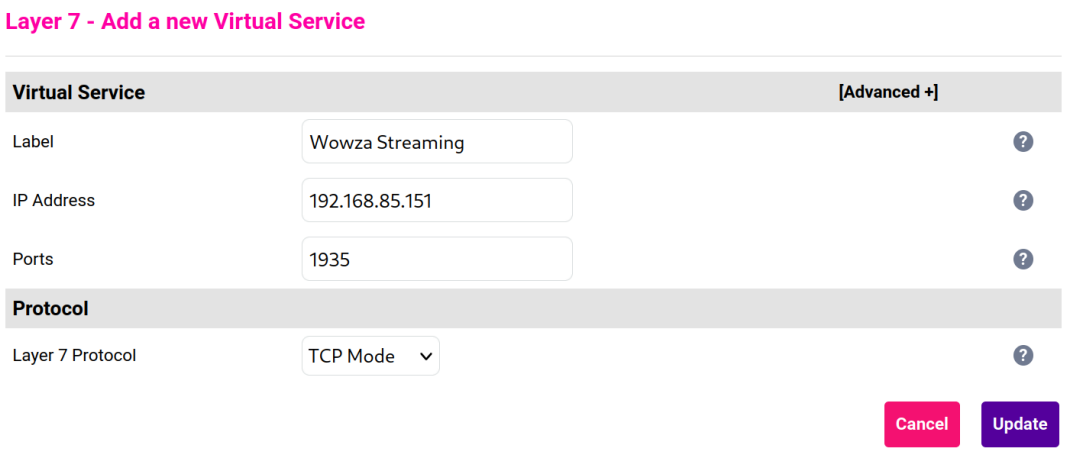
1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Real Servers* and click on **Add a new Real Server** next to the newly created VIP.
2. Define the *Label* for the real server as required, e.g. **Wowza Srv 1**.
3. Set the *Real Server IP Address* field to the required IP address, e.g. **192.168.85.200**.
4. Set the *Real Server Port* field to **8088**.
5. Click **Update**.
6. Repeat these steps to add additional Wowza Streaming Engine servers as required.



11.2. Configuring VIP 2 - Wowza Streaming

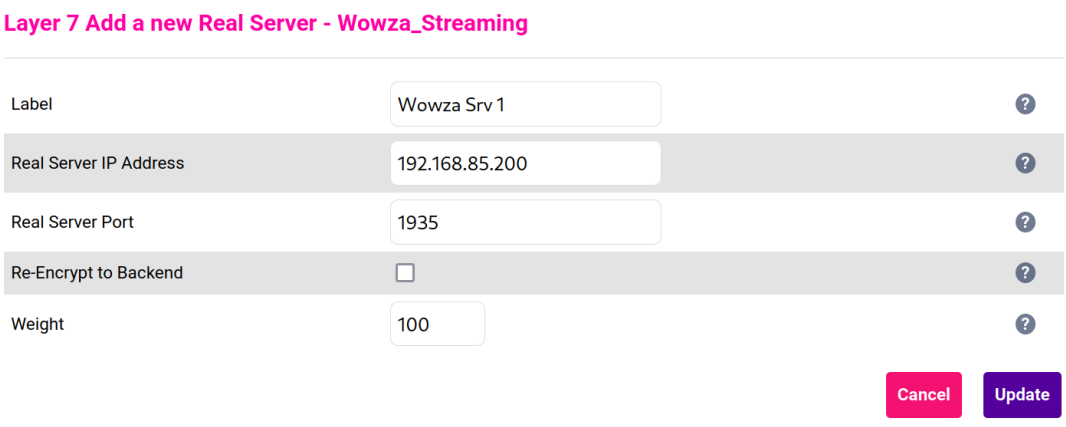
11.2.1. Configuring the Virtual Service (VIP)

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. **Wowza Streaming**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.85.151**.
4. Set the *Ports* field to **1935**.
5. Set the *Layer 7 Protocol* to **TCP Mode**.
6. Click **Update** to create the virtual service.



11.2.2. 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. Define the *Label* for the real server as required, e.g. **Wowza Srv 1**.
3. Set the *Real Server IP Address* field to the required IP address, e.g. **192.168.85.200**.
4. Set the *Real Server Port* field to **1935**.
5. Click **Update**.
6. Repeat these steps to add additional Wowza Streaming Engine servers as required.



11.3. Finalizing the Configuration

To apply the new settings, HAProxy must be reloaded. This can be done using the button in the "Commit

changes" 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**.

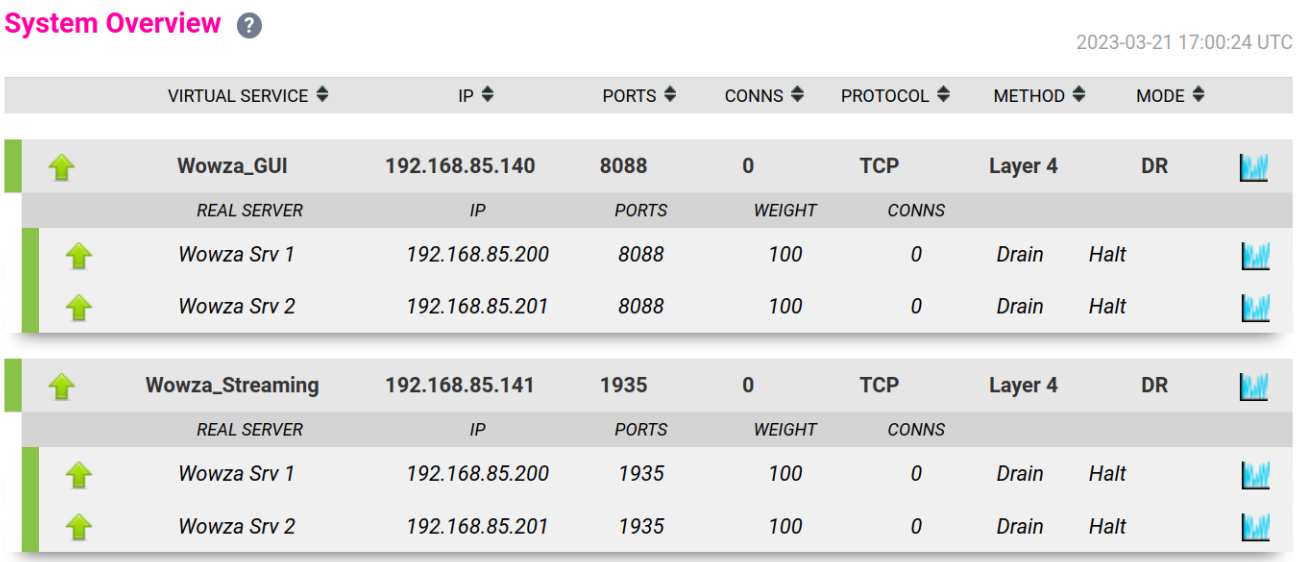
12. Testing & Verification

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

12.1. Using System Overview

The System Overview can be viewed in the WebUI. It shows a graphical view of all VIPs & RIPs (i.e. the Wowza servers) and shows the state/health of each server as well as the state of the cluster as a whole.

The example below shows a **layer 4 DR mode** configuration load balancing a pair of Wowza Streaming Engine servers, where both servers are healthy and available to accept connections:



13. 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.

14. Further Documentation

For additional information, please refer to the Administration Manual.

15. Appendix

15.1. 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 is fully configured first, then the Secondary appliance can be added to create an HA pair. Once the HA pair is configured, load balanced services must be configured and modified on the Primary appliance. The Secondary appliance will be automatically kept in sync.

> **Note** For Enterprise Azure, the HA pair should be configured first. For more information, please refer to the Azure Quick Start/Configuration Guide available in the documentation library

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.

15.1.1. 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:

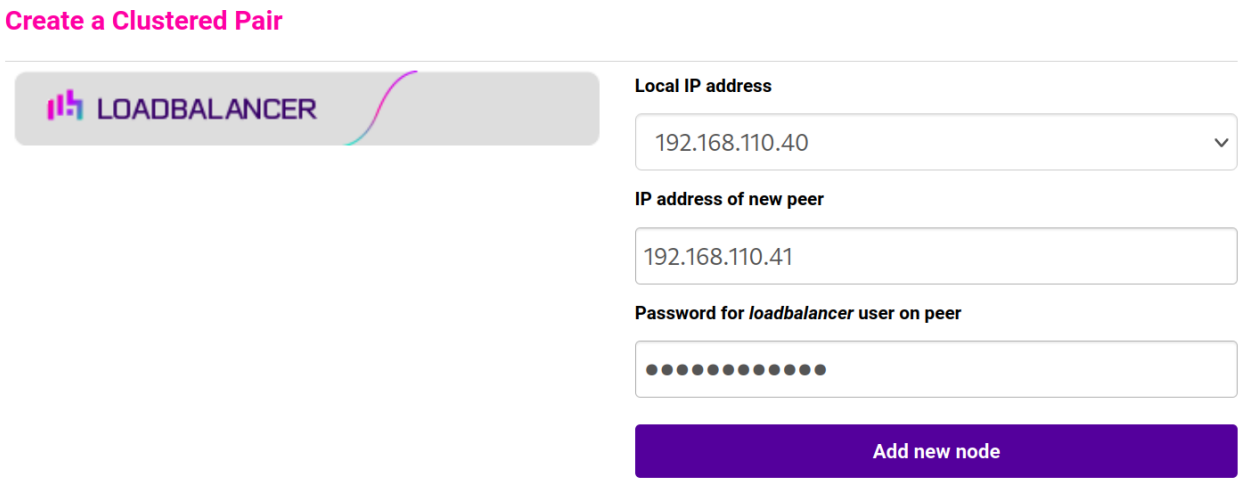
WebUI Main Menu Option	Sub Menu Option	Description
Local Configuration	Hostname & DNS	Hostname and DNS settings
Local Configuration	Network Interface Configuration	Interface IP addresses, bonding configuration and VLANs
Local Configuration	Routing	Default gateways and static routes
Local Configuration	System Date & time	Time and date related settings
Local Configuration	Physical – Advanced Configuration	Various appliance settings
Local Configuration	Portal Management	Portal management settings
Local Configuration	Security	Security settings
Local Configuration	SNMP Configuration	SNMP settings
Local Configuration	Graphing	Graphing settings
Local Configuration	License Key	Appliance licensing
Maintenance	Backup & Restore	Local XML backups
Maintenance	Software Updates	Appliance software updates
Maintenance	Fallback Page	Fallback page configuration
Maintenance	Firewall Script	Firewall (iptables) configuration
Maintenance	Firewall Lockdown Wizard	Appliance management lockdown settings

> **Important** Make sure that where any of the above have been configured on the Primary appliance, they're also configured on the Secondary.

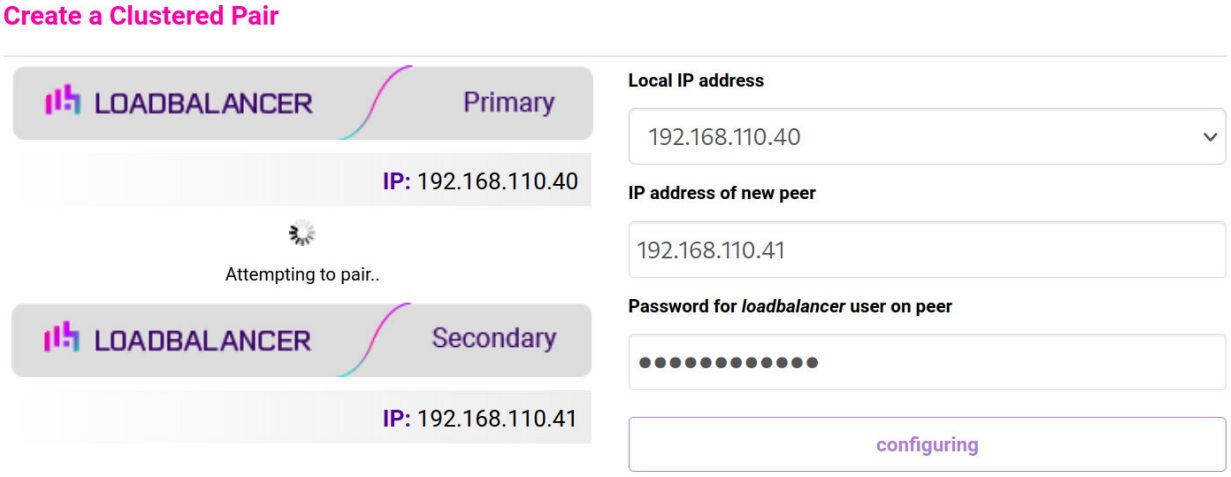
15.1.2. Configuring the HA Clustered Pair

> **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.

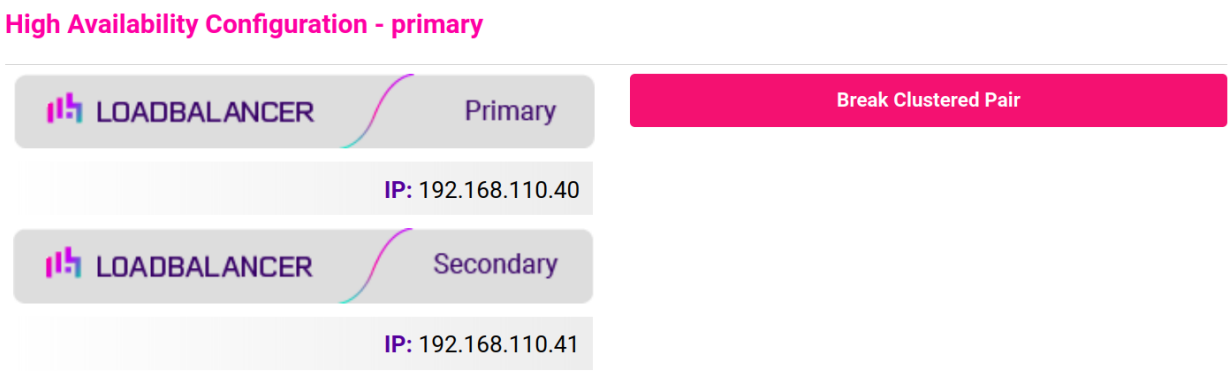
1. Deploy a second appliance that will be the Secondary and configure initial network settings.
2. Using the WebUI on the Primary appliance, navigate to: *Cluster Configuration > High-Availability Configuration*.



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



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



7. To finalize the configuration, restart heartbeat and any other services as prompted in the "Commit changes" 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.

15.2. Solving the ARP Problem for Linux

There are two different approaches on how to configure a Linux server for correct operation when DR mode load balancing is in use:

- Modifying the server's ARP behavior and adding the relevant VIP addresses to the loopback interface
- Using NAT to convince the server to accept and reply to packets addressed to the relevant VIP addresses

Four independent methods are described below along with instructions. Each method follows one of the two approaches above. The specific method chosen will depend on technical requirements, the Linux distribution in use, and personal preferences.

The first method involves setting kernel parameters to alter the server's ARP behavior and adding IP addresses to the loopback interface. This method should be universally applicable to any Linux server **making this the preferred method**.

If setting kernel parameters and adding IP addresses is not possible for some reason, the remaining three methods describe setting up a server for DR mode operation by using NAT via the **redirect** target/statement. The specific instructions depend on the packet filtering framework and tooling in use, which varies between Linux distributions. Methods are presented for iptables, nftables, and the `firewall-cmd` tool.

15.2.1. Method 1: ARP Behavior and Loopback Interface Changes

This is the preferred method as it should be applicable to any Linux server and doesn't require any additional

packet filtering or NAT considerations.

Each real server needs the loopback interface to be configured with the virtual IP addresses (VIPs) of the relevant load balanced services. This is often just a single VIP address, but the logic described below can be extended to cover multiple VIPs on a server. Having the VIPs on the loopback interface allows the server to accept inbound load balanced packets that are addressed to a VIP.

The server **must not** respond to ARP requests for the VIP addresses. The server also **must not** use ARP to announce the fact that it owns the VIP addresses. This is necessary to prevent IP address conflicts, as ***all*** of the real servers ***and*** the load balancer will own the VIP addresses. Only the load balancer should announce ownership of the VIPs.

To configure the behavior described above, follow all of the steps below on each real server.

Step 1 of 4: Re-configuring ARP behavior

This step is only applicable if IPv4-based virtual services are in use.

Add the following lines to the file `/etc/sysctl.conf` (create this file if it does not already exist):

```
net.ipv4.conf.all.arp_ignore=1
net.ipv4.conf.eth0.arp_ignore=1
net.ipv4.conf.eth1.arp_ignore=1
net.ipv4.conf.all.arp_announce=2
net.ipv4.conf.eth0.arp_announce=2
net.ipv4.conf.eth1.arp_announce=2
```

Adjust the commands shown above to suit the server's network configuration, e.g. a different number of network interfaces or a different interface naming convention.

Note

For reference, the effect of these kernel parameter changes on the server is as follows:

- `arp_ignore=1`: This configures the server to only reply to an ARP request if the request's target IP address is local to the incoming interface. This can never be true for VIP addresses on the loopback interface, as the loopback interface can never be an incoming interface for ARP requests from other devices. Hence, ARP requests for VIP addresses are always ignored.
- `arp_announce=2`: This prevents the server from sending an ARP request out of an interface ***A*** where the ARP request's sender/source address is stated to be an IP address that is local to some other interface ***B***. For example, this prevents the server from sending an ARP request ***from*** a VIP address (which is local to the loopback interface) out of `eth0`, which would announce that the server owns the VIP address.

Step 2 of 4: Re-configuring duplicate address detection (DAD) behavior

This step is only applicable if IPv6-based virtual services are in use.

Add the following lines to the file `/etc/sysctl.conf` (create this file if it does not already exist):

```
net.ipv6.conf.lo.dad_transmits=0
net.ipv6.conf.lo.accept_dad=0
```

Note

For reference, the effect of these kernel parameter changes on the server is as follows:

- `dad_transmits=0`: This prevents a given interface from sending out duplicate address detection probes in order to test the uniqueness of unicast IPv6 addresses. Any IPv6 VIP addresses will ***not*** be unique, so this mechanism is disabled.
- `accept_dad=0`: This prevents a given interface from accepting duplicate address detection messages. This prevents any IPv6 VIP addresses from being marked as duplicate addresses.

Step 3 of 4: Applying the new settings

To apply the new settings, either reboot the real server or execute the following command to immediately apply the changes:

```
/sbin/sysctl -p
```

Note

Steps 1, 2, and 3 can be replaced by instead modifying the necessary kernel variables by writing directly to their corresponding files under `/proc/sys/`. Note that changes made in this way ***will not persist across reboots***.

Execute the following commands (as root) to implement these temporary changes (adapting the number of interfaces and interface names as needed):

```
echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
echo 1 > /proc/sys/net/ipv4/conf/eth0/arp_ignore
echo 1 > /proc/sys/net/ipv4/conf/eth1/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/eth0/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/eth1/arp_announce
echo 0 > /proc/sys/net/ipv6/conf/lo/dad_transmits
echo 0 > /proc/sys/net/ipv6/conf/lo/accept_dad
```

Step 4 of 4: Adding the virtual IP addresses (VIPs) to the loopback interface

Each of the VIP addresses must be permanently added to the loopback interface. VIPs must be added with a network prefix of /32 for IPv4 addresses or /128 for IPv6 addresses. The IP addresses can be added using the usual configuration files and tools for modifying network interfaces, which vary between different Linux distributions.

As an alternative, the `ip` command can be used as a universal way to add IP addresses to any Linux server. Note that addresses added in this way ***will not persist across reboots***. To make these addresses permanent, add the `ip` commands to an appropriate startup script such as `/etc/rc.local`.

Execute the following `ip` command for each IPv4 VIP:

```
ip addr add dev lo <IPv4-VIP>/32
```

Execute the following `ip` command for each IPv6 VIP:

```
ip addr add dev lo <IPv6-VIP>/128
```

To check that the VIPs have been successfully added, execute the command:

```
ip addr ls
```

To remove an IPv4 VIP from the loopback adapter, execute the command:

```
ip addr del dev lo <IPv4-VIP>/32
```

To remove an IPv6 VIP from the loopback adapter, execute the command:

```
ip addr del dev lo <IPv6-VIP>/128
```

15.2.2. Method 2: NAT "redirect" via iptables

iptables can be used on each real server to identify incoming packets that are addressed to a virtual IP address (VIP) and redirect those packets to the server itself. This is achieved using the **REDIRECT** target in iptables, which performs the necessary NAT to make this possible. This allows a real server to accept packets addressed to a VIP without the server owning the VIP.

Execute the following command to put the necessary iptables rule in place to redirect traffic for a single IPv4 VIP address. Note that iptables rules added in this way ***will not persist across reboots***. To make such a rule permanent, either add the rule to an iptables firewall script, if one is provided with the Linux distribution in question, or add the command to an appropriate startup script such as `/etc/rc.local` on each real server.

```
iptables -t nat -A PREROUTING -d <IPv4-VIP> -j REDIRECT
```

The VIP address should be changed to match the virtual service in question, for example:

```
iptables -t nat -A PREROUTING -d 10.0.0.21 -j REDIRECT
```

The example above will redirect any incoming packets destined for 10.0.0.21 (the virtual service) locally, i.e. to the primary address of the incoming interface on the real server.

If a real server is responsible for serving ***multiple*** VIPs then additional iptables rules should be added to cover each VIP.

For an IPv6 VIP address, a command like the following should be used:

```
ip6tables -t nat -A PREROUTING -d <IPv6-VIP> -j REDIRECT
```

The VIP address should be changed to match the virtual service in question, for example:

```
ip6tables -t nat -A PREROUTING -d 2001:db8::10 -j REDIRECT
```

> **Note** Method 2 may not be appropriate when using IP-based virtual hosting on a web server. This is because an iptables **REDIRECT** rule will redirect incoming packets to the ***primary address*** of the incoming interface on the web server rather than any of the virtual hosts that are configured. Where this is an issue, use method 1 instead.

15.2.3. Method 3: NAT "redirect" via nftables

nftables is the modern Linux kernel packet filtering framework. It is supported on all major Linux distributions and has replaced iptables as the default framework on most major distributions.

nftables can be used on each real server to identify incoming packets that are addressed to a virtual IP address (VIP) and redirect those packets to the server itself. This is achieved using the **redirect** statement in nftables, which performs the necessary NAT to make this possible. This allows a real server to accept packets addressed to a VIP without the server owning the VIP.

Use a script like the following to put the necessary nftables structures in place to redirect traffic for both IPv4 and IPv6 VIP addresses. To make such a configuration permanent, either add the `inet nat` table to an nftables firewall script, if one is provided with the Linux distribution in question, or configure a script like the following to execute as a startup script on each real server.

```
#!/usr/sbin/nft -f

table inet nat {
    chain prerouting {
        comment "Allow server to accept packets destined for VIP addresses";
        type nat hook prerouting priority -100; policy accept;
        ip daddr <IPv4-VIP> redirect comment "Description"
        ip6 daddr <IPv6-VIP> redirect comment "Description"
    }
}
```

The VIP addresses and comments should be changed to match the virtual services in question, for example:

```
#!/usr/sbin/nft -f

table inet nat {
    chain prerouting {
        comment "Allow server to accept packets destined for VIP addresses";
        type nat hook prerouting priority -100; policy accept;
        ip daddr 10.0.0.21 redirect comment "VIP 1: HTTP"
        ip6 daddr 2001:db8::10 redirect comment "VIP 2: HTTPS"
    }
```

```
}
```

The example above will redirect any incoming packets destined for 10.0.0.21 or 2001:db8::10 (the virtual services) locally, i.e. to the primary address of the incoming interface (for each IP version) on the real server.

Note that **Linux kernels prior to 5.2** may not support performing NAT (which is required for the **redirect** statement) in an inet family table. In this scenario, use either an ip or an ip6 family table instead, or both if a mixture of IPv4 and IPv6 VIPs are in use on the same server. Also note that older kernels may not support the use of comments in chains.

Note that **Linux kernels prior to 4.18** require explicitly registering both prerouting and postrouting chains in order for the implicit NAT of the **redirect** statement to be correctly performed in both the inbound and outbound directions.

A legacy-friendly setup may look like the following:

```
#!/usr/sbin/nft -f

table ip nat {
    chain prerouting {
        type nat hook prerouting priority -100; policy accept;
        ip daddr 10.0.0.21 counter redirect comment "VIP 1: HTTP"
    }

    chain postrouting {
        type nat hook postrouting priority 100; policy accept;
    }
}

table ip6 nat {
    chain prerouting {
        type nat hook prerouting priority -100; policy accept;
        ip6 daddr 2001:db8::10 counter redirect comment "VIP 2: HTTPS"
    }

    chain postrouting {
        type nat hook postrouting priority 100; policy accept;
    }
}
```

> **Note**
>
> Method 3 may not be appropriate when using IP-based virtual hosting on a web server. This is because an nftables **redirect** statement will redirect incoming packets to the *primary address* of the incoming interface on the web server rather than any of the virtual hosts that are configured. Where this is an issue, use method 1 instead.

15.2.4. Method 4: NAT "redirect" via firewall-cmd

Some recent versions of Linux distributions make use of firewalld as a high-level firewall configuration framework. In this case, while it may actually be iptables performing the work at a lower level, it may be preferred to implement the iptables NAT solution described in method 2 in firewalld, as opposed to directly manipulating iptables. This is achieved by using the `firewall-cmd` tool provided by firewalld and executing a command like

the following on each real server:

```
firewall-cmd --permanent --direct --add-rule ipv4 nat PREROUTING 0 -d <IPv4-VIP> -j REDIRECT
```

The VIP address should be changed to match the virtual service in question, for example:

```
firewall-cmd --permanent --direct --add-rule ipv4 nat PREROUTING 0 -d 10.0.0.50 -j REDIRECT
```

To apply the new configuration, reload the firewall rules like so:

```
firewall-cmd --reload
```

Configuration applied in this way will be permanent and will persist across reboots.

> **Note**
>
> Method 4 may not be appropriate when using IP-based virtual hosting on a web server. This is because an iptables **REDIRECT** rule will redirect incoming packets to the ***primary address*** of the incoming interface on the web server rather than any of the virtual hosts that are configured. Where this is an issue, use method 1 instead.

16. Document Revision History

Version	Date	Change	Reason for Change	Changed By
1.0.0	21 March 2023	Initial version		AH
1.1.0	24 March 2023	New document theme Modified diagram colours	Branding update	AH

LOADBALANCER

Visit us: www.loadbalancer.org

Phone us: +44 (0)330 380 1064

Phone us: +1 833 274 2566

Email us: info@loadbalancer.org

Follow us: @loadbalancer.org

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.