LOADBALANCER

DEPLOYMENT GUIDE

Load Balancing Leostream

Version 1.2.0

Table of Contents

1. About this Guide . . . 4
2. Loadbalancer.org Appliances Supported . . . 4
3. Software Versions Supported . . . 4
   3.1. Loadbalancer.org Appliance . . . 4
   3.2. Leostream . . . 4
4. Leostream . . . 4
5. Leostream Platform Components . . . 4
6. Load Balancing Leostream . . . 5
   6.1. Load Balancing & HA Requirements . . . 5
   6.2. Persistence (aka Server Affinity) . . . 5
   6.3. Virtual Service (VIP) Requirements . . . 5
   6.4. Port Requirements . . . 5
7. Deployment Concept . . . 6
   7.1. Multiple Leostream Gateways Connecting to a Single Leostream Connection Broker . . . 6
   7.2. Multiple Leostream Gateways Connecting to a Cluster of Leostream Connection Brokers . . . 7
8. Load Balancer Deployment Methods . . . 8
   8.1. Layer 4 DR Mode . . . 8
   8.2. Layer 4 NAT Mode . . . 9
   8.3. Our Recommendation . . . 12
   8.4. Leostream Gateway Configuration . . . 12
   8.5. Leostream Connection Broker Configuration . . . 13
   8.6. Leostream Agent Configuration . . . 16
   8.7. Leostream Connect Client Configuration . . . 17
      8.7.1. Windows Clients . . . 17
      8.7.2. Java Clients . . . 18
9. Loadbalancer.org Appliance – the Basics . . . 18
   9.1. Virtual Appliance . . . 18
   9.2. Initial Network Configuration . . . 19
   9.3. Accessing the Appliance WebUI . . . 19
      9.3.1. Main Menu Options . . . 20
   9.4. Appliance Software Update . . . 21
      9.4.1. Online Update . . . 21
      9.4.2. Offline Update . . . 21
   9.5. Ports Used by the Appliance . . . 22
   9.6. HA Clustered Pair Configuration . . . 23
10. Appliance Configuration for Leostream - Using Layer 4 DR Mode . . . 23
   10.1. Configuring VIP 1 - Leostream Gateway Service . . . 23
      10.1.1. Configuring the Virtual Service (VIP) . . . 23
      10.1.2. Defining the Real Servers (RIPs) . . . 24
   10.2. Configuring VIP 2 - Leostream Connection Broker Service . . . 24
      10.2.1. Configuring the Virtual Service (VIP) . . . 25
      10.2.2. Defining the Real Servers (RIPs) . . . 26
11. Appliance & Server Configuration for Leostream - Using Layer 4 NAT Mode . . . 26
   11.1. Configure the Load Balancer’s Network Interfaces . . . 26
   11.2. Configuring VIP 1 - Leostream Gateway Service . . . 27
      11.2.1. Configuring the Virtual Service (VIP) . . . 27
      11.2.2. Defining the Real Servers (RIPs) . . . 28
   11.3. Configuring VIP 2 - Leostream Connection Broker Service . . . 29

11.3.1. Configuring the Virtual Service (VIP) 29
11.3.2. Defining the Real Servers (RIPs) 30
11.4. Create a Floating IP to Use for the Leostream Servers' Default Gateway 31
11.4.1. To Create a Floating IP Address on the Load Balancer 31
11.5. Leostream Server Configuration 31
11.5.1. Default Gateway 32
12. Testing & Verification 32
12.1. Testing the Load Balanced Gateway Service 32
12.2. Using System Overview 34
13. Technical Support 34
14. Further Documentation 34
15. Appendix 35
15.1. Configuring HA - Adding a Secondary Appliance 35
15.1.1. Non-Replicated Settings 35
15.1.2. Configuring the HA Clustered Pair 36
15.2. Solving the ARP Problem for Linux 37
15.2.1. Method 1: ARP Behavior and Loopback Interface Changes 37
15.2.2. Method 2: NAT "redirect" via iptables 40
15.2.3. Method 3: NAT "redirect" via nftables 41
15.2.4. Method 4: NAT "redirect" via firewall-cmd 42
16. Document Revision History 44

1. About this Guide

This guide details the steps required to configure a load balanced Leostream environment utilizing Loadbalancer.org appliances. It covers the configuration of the load balancers and also any Leostream 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 Leostream. 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. Leostream

- Leostream Connection Broker – 9.0 and later
- Leostream Gateway – 2.0 and later

4. Leostream

Leostream provides the critical remote desktop connection management technology required for organizations to build successful large-scale remote access solutions for physical, virtual, and cloud-hosted desktops. The Leostream Platform is the industry's most widely deployed vendor-independent remote desktop connection management solution, enabling enterprises to integrate the complex array of clients, hosting platforms, guest operating systems, and display protocols required for successful VDI, hosted desktop, and application deployments.

5. Leostream Platform Components

- **Connection Broker**: The main application that manages the hosted desktop environment. The Connection Broker is the central management layer for configuring your deployment, including inventorying and provisioning desktops, assigning and connecting users to these desktops, and defining the end-user experience. The Connection Broker also includes a web portal for users to access their hosted resources.

- **Leostream Gateway**: An optional application that provides HTML5-based clientless remote access for users connecting to their remote desktop. The Leostream Gateway also provides gateway functionality for protocols such as RDP, HP ZCentral Remote Boost, NICE DCV, and Mechdyne TGX, to connect users to desktops that are hosted in a network that is isolated from the user's client device.
- **Leostream Agent**: When installed on the remote desktop, the Leostream Agent provides the Connection Broker with insight into the connection status of remote users, including when they log out, disconnect, or are idle on their desktop. The Agent also manages enhancements such as USB device passthrough and network printer redirection. The Leostream Agent is available for Microsoft Windows, Linux, and macOS operating systems.
- **Leostream Connect**: A software client provided by Leostream that allows users to log into your Leostream environment and access their hosted resources from fat or thin clients. Using Leostream Connect, you can repurpose existing desktops and laptops as client devices, lowering the cost of VDI deployments. Some thin clients provide built-in Leostream Connect clients.

The Leostream Connection Broker and Gateway are deployed onto Linux hosts.

The Leostream Client and Agent can be deployed onto Windows, Linux, and Mac hosts.

6. Load Balancing Leostream

Note	It's highly recommended that you have a working Leostream environment first before implementing the load balancer.

6.1. Load Balancing & HA Requirements

For high availability and scalability, it is recommended that multiple Leostream Gateway servers **and** multiple Connection Broker servers are deployed in load balanced clusters.

6.2. Persistence (aka Server Affinity)

Source IP address based persistence is required to successfully load balance a Leostream deployment. This is true for load balancing Leostream gateway servers and for load balancing connection brokers.

6.3. Virtual Service (VIP) Requirements

To provide load balancing and HA for Leostream, the following VIP is required:

- Leostream Gateway Service

Optionally, an additional VIP may be required as follows:

- Leostream Connection Broker Service

6.4. Port Requirements

For the purposes of this guide, the focus will be on the RDP, PCoIP, and HP ZCentral remote protocols. Leostream is also compatible with a plethora of other different remote connection protocols, however. Refer to the official Leostream documentation for further details.

The following table shows the ports that are load balanced:

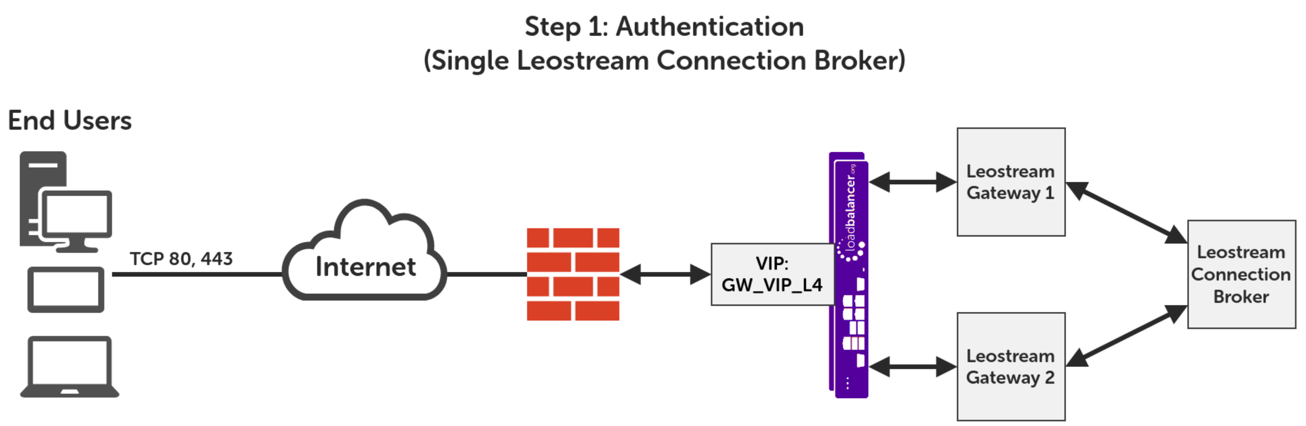
Port	Protocols	Use
80	TCP/HTTP	HTTP Logon to Leostream Service
443	TCP/HTTPS	HTTPS Logon to Leostream Service
3389	TCP/UDP/RDP	(Optional) Connection to RDP Hosts
42966	TCP/UDP/HP RGS	(Optional) ZCentral Remote Boost (Formerly HP Remote Graphics Software)
4172	TCP/UDP/PCoIP	(Optional) PC-over-IP Remote Display Protocol
50001	TCP/PCoIP	(Optional) PC-over-IP Remote Display Protocol
50002	TCP/PCoIP	(Optional) PC-over-IP Remote Display Protocol

> **Note** Optional protocols are dependent on the remote desktop protocol in use for client connections.

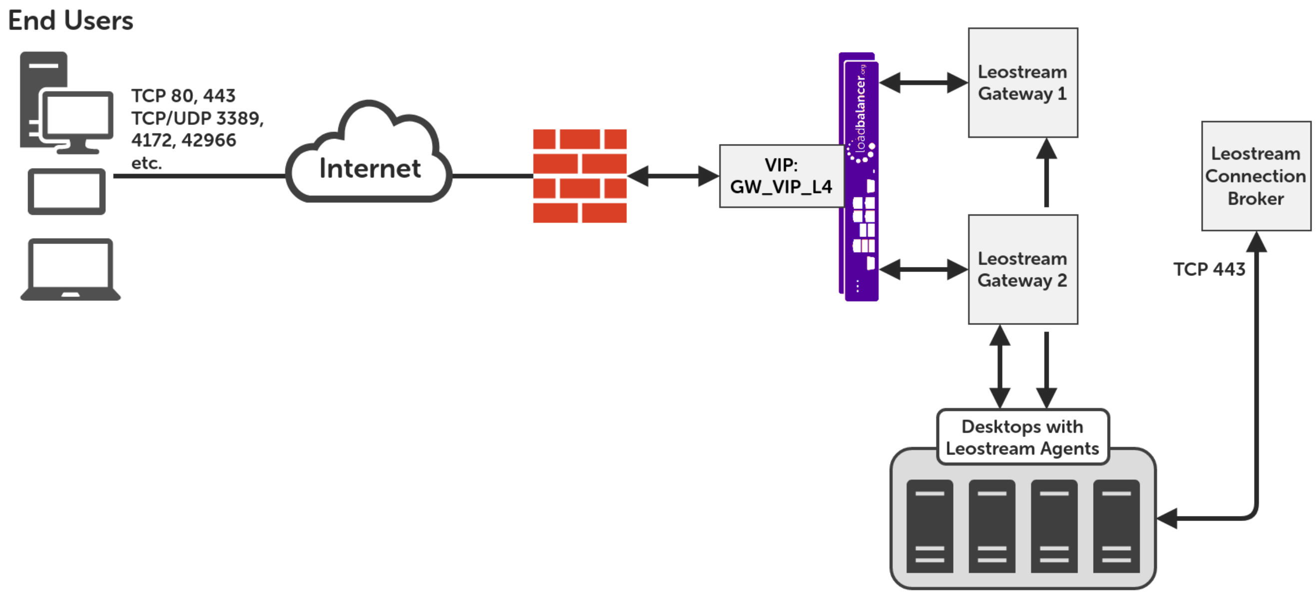
7. Deployment Concept

Leostream can be deployed in two different ways that can be load balanced.

7.1. Multiple Leostream Gateways Connecting to a Single Leostream Connection Broker



Step 2: Remote Connection
(Single Leostream Connection Broker)

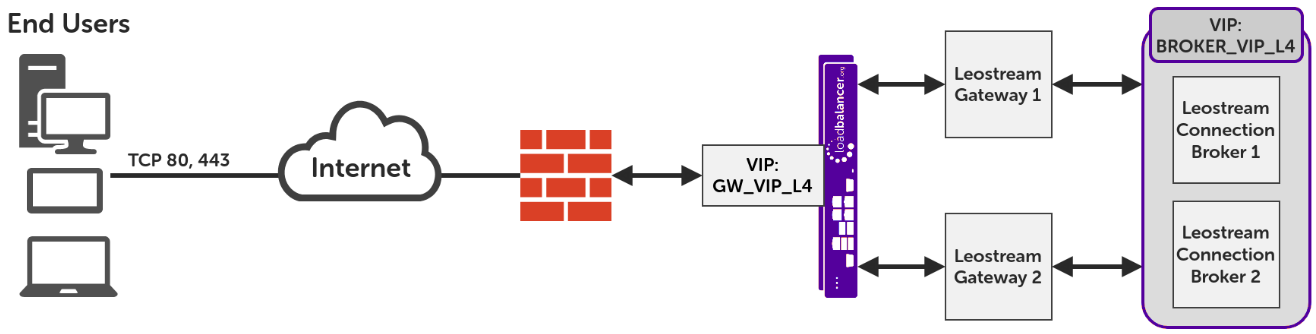


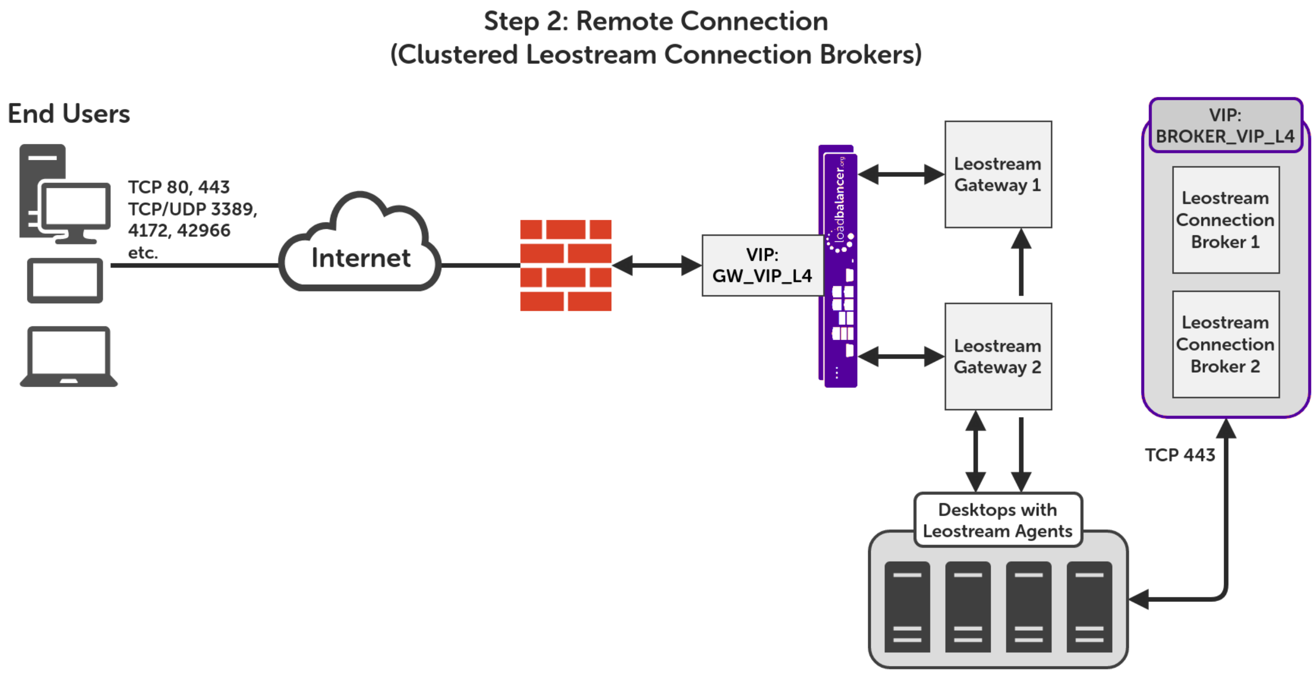
VIP = **V**irtual **IP** Address

> 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.

7.2. Multiple Leostream Gateways Connecting to a Cluster of Leostream Connection Brokers

Step 1: Authentication
(Clustered Leostream Connection Brokers)





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.

8. Load Balancer Deployment Methods

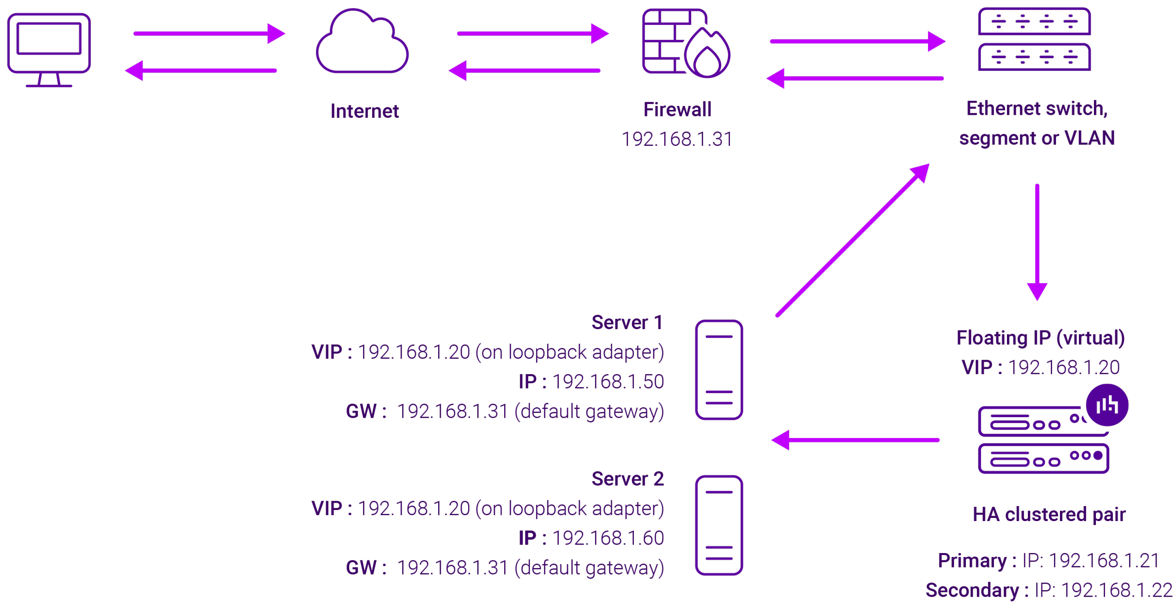
The load balancer can be deployed in 4 fundamental ways: ***Layer 4 DR mode***, ***Layer 4 NAT mode***, ***Layer 4 SNAT mode***, and ***Layer 7 SNAT mode***.

For Leostream, using layer 4 DR mode is recommended. It it also possible to use layer 4 NAT mode, however the performance of this set up is not as great as layer 4 DR 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 Leostream - Using Layer 4 DR Mode", and for configuring using layer 4 NAT mode refer to Section 11, "Appliance & Server Configuration for Leostream - Using Layer 4 NAT Mode".

8.1. 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.

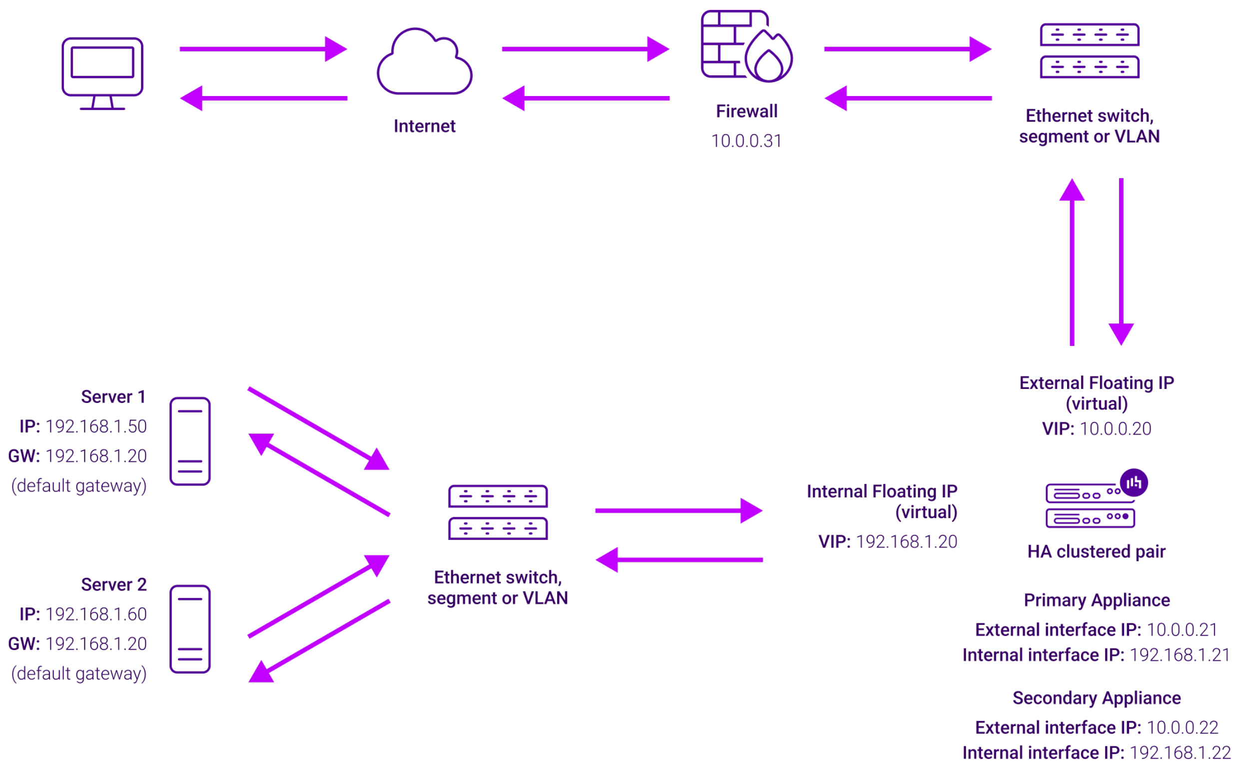
> **Note** Kemp, Brocade, Barracuda & A10 Networks call this ***Direct Server Return*** and F5 call it ***nPath***.



- 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.

8.2. Layer 4 NAT Mode

Layer 4 NAT mode is a high performance solution, although not as fast as layer 4 DR mode. This is because real server responses must flow back to the client via the load balancer rather than directly as with DR mode. The image below shows an example network diagram for this mode.



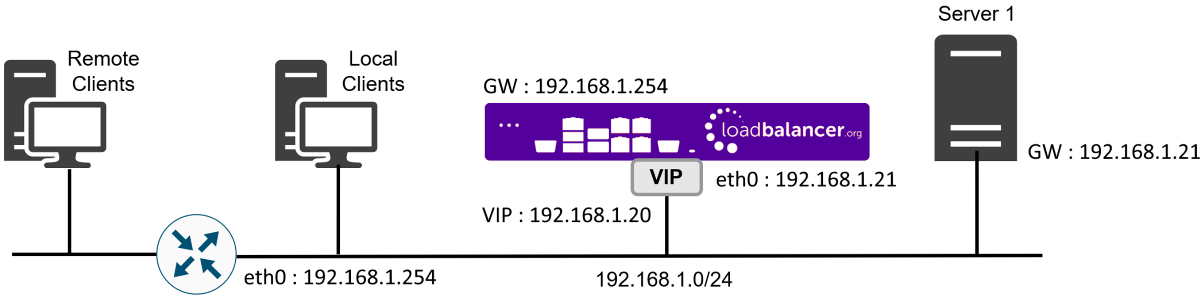
- The load balancer translates all requests from the Virtual Service to the Real Servers.
- NAT mode can be deployed in the following ways:
  - **Two-arm (using 2 Interfaces)** (as shown above) - Here, 2 subnets are used. The VIP is located in one subnet and the load balanced Real Servers are located in the other. The load balancer requires 2 interfaces, one in each subnet.

    > **Note** This can be achieved by using two network adapters, or by creating VLANs on a single adapter.

    - Normally **eth0** is used for the internal network and **eth1** is used for the external network, although this is not mandatory since any interface can be used for any purpose.
    - If the Real Servers require Internet access, *Auto-NAT* should be enabled using the WebUI menu option: *Cluster Configuration > Layer 4 - Advanced Configuration*, the external interface should be selected.
    - The default gateway on the Real Servers must be set to be an IP address on the load balancer.

    > **Note** For an HA clustered pair, a floating IP should be added to the load balancer and used as the Real Server's default gateway. This ensures that the IP address can "float" (move) between Primary and Secondary appliances.

    - Clients can be located in the same subnet as the VIP or any remote subnet provided they can route to the VIP.
  - **One-arm (using 1 Interface)** - Here, the VIP is brought up in the same subnet as the Real Servers.



- To support remote clients, the default gateway on the Real Servers must be an IP address on the load balancer and routing on the load balancer must be configured so that return traffic is routed back via the router.

Note	For an HA clustered pair, a floating IP should be added to the load balancer and used as the Real Server's default gateway. This ensures that the IP address can "float" (move) between Primary and Secondary appliances.

- To support local clients, return traffic would normally be sent directly to the client bypassing the load balancer which would break NAT mode. To address this, the routing table on the Real Servers must be modified to force return traffic to go via the load balancer. For more information please refer to One-Arm (Single Subnet) NAT Mode.

- If you want Real Servers to be accessible on their own IP address for non-load balanced services, e.g. RDP, you will need to setup individual SNAT and DNAT firewall script rules for each Real Server or add additional VIPs for this.
- Port translation is possible with Layer 4 NAT mode, e.g. VIP:80 → RIP:8080 is supported.
- NAT mode is transparent, i.e. the Real Server will see the source IP address of the client.

NAT Mode Packet re-Writing

In NAT mode, the inbound destination IP address is changed by the load balancer from the Virtual Service IP address (VIP) to the Real Server. For outbound replies the load balancer changes the source IP address of the Real Server to be the Virtual Services IP address.

The following table shows an example NAT mode setup:

Protocol	VIP	Port	RIP	Port
TCP	10.0.0.20	80	192.168.1.50	80

In this simple example all traffic destined for IP address 10.0.0.20 on port 80 is load-balanced to the real IP address 192.168.1.50 on port 80.

Packet rewriting works as follows:

1) The incoming packet for the web server has source and destination addresses as:

Source	x.x.x.x:34567	**Destination**	10.0.0.20:80

2) The packet is rewritten and forwarded to the backend server as:

Source	x.x.x.x:34567	**Destination**	192.168.1.50:80

3) Replies return to the load balancer as:

Source	192.168.1.50:80	**Destination**	x.x.x.x:34567

4) The packet is written back to the VIP address and returned to the client as:

Source	10.0.0.20:80	**Destination**	x.x.x.x:34567

8.3. 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 NAT mode is recommended.

8.4. Leostream Gateway Configuration

Carry out the following instructions on each gateway server:

1. **If deploying using DR mode:** Change the ARP behaviour of the server by following the instructions in the section Solving the ARP Problem for Linux of the appendix.
2. Open an SSH connection to the Leostream Gateway host.
3. Run the command leostream-gateway --broker <BROKER_VIP_L4>
   - ***Non-clustered connection broker deployment:*** Use the IP address / FQDN of the connection broker server.
   - ***Clustered connection broker deployment:*** Use the VIP address of the connection broker virtual service.

```
[root@localhost ~]# leostream-gateway --broker 192.168.98.237
Connection Broker forwarding is enabled
```

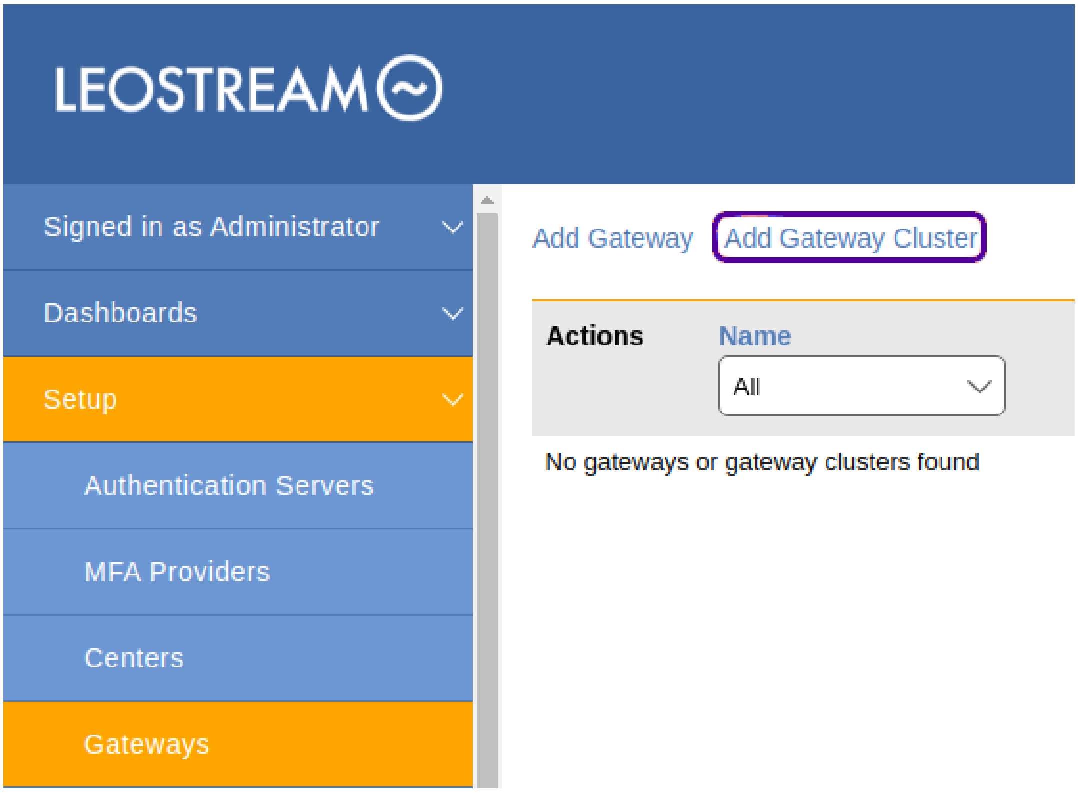
4. Run the command leostream-gateway --info to confirm that the connection broker has been added to the configuration.

```
[root@localhost ~]# leostream-gateway --info
OS is CentOS 7
Port range is 20001-23000
Gateway version is 2.0.0.18
The VPN is OFF
Connection Broker forwarding is ON to 192.168.98.237
Azure Broker forwarding is OFF
Guacamole is ENABLED
This Gateway is attached to a Connection Broker
```

8.5. Leostream Connection Broker Configuration

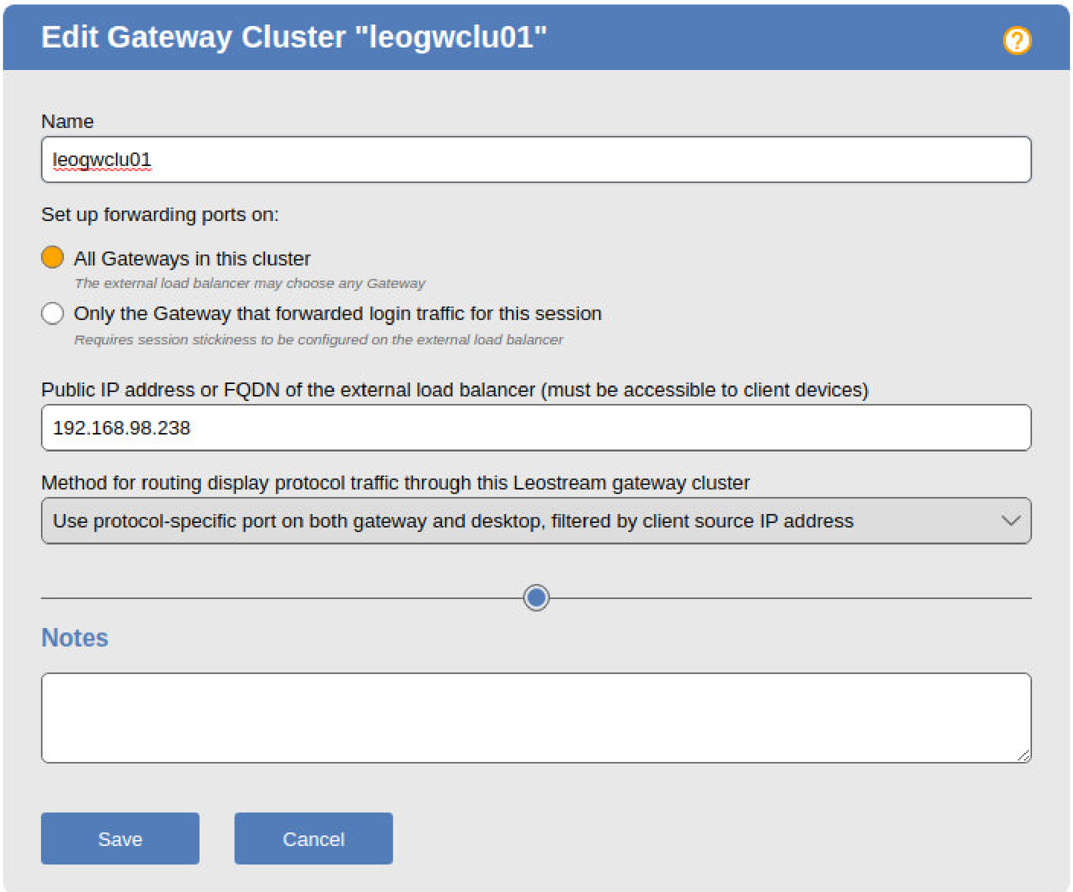
If load balancing multiple connection brokers (this is optional), carry out the following instructions on each connection broker server:

1. **If deploying using DR mode:** Change the ARP behaviour of the server by following the instructions in the section Solving the ARP Problem for Linux of the appendix.
2. Connect to the connection broker server via browser and login as an admin user.
3. From the left hand menu, expand ***Setup***, navigate to ***Gateways***, and click on **Add Gateway Cluster** as the top of the main window.

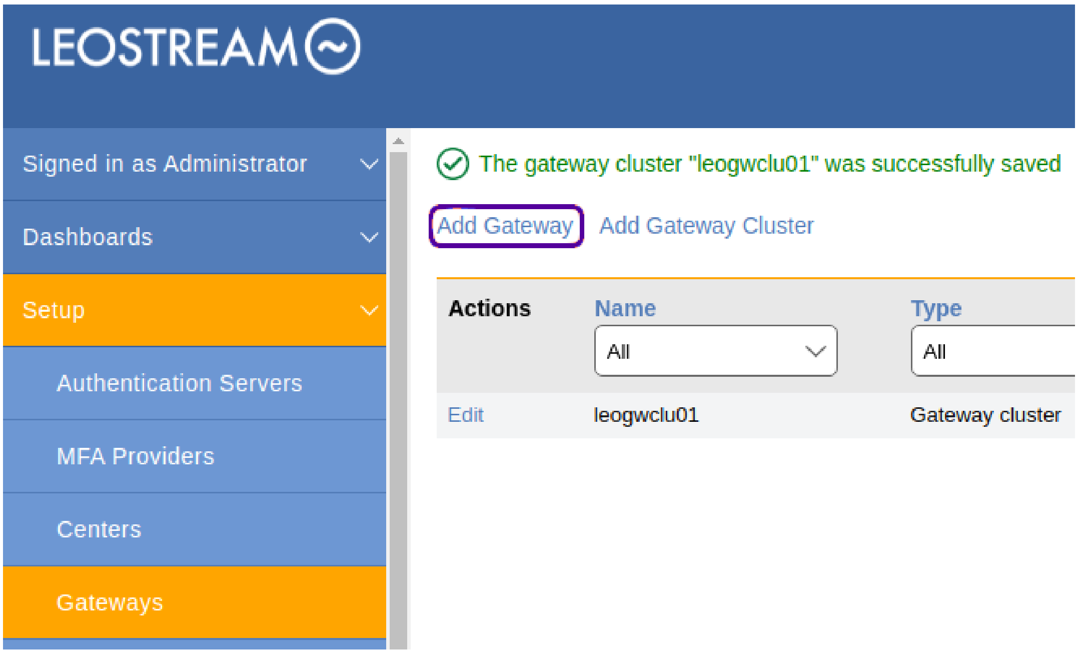


4. Set the name of the cluster.
5. Choose the option **All Gateways in this cluster**.
6. In the text box ***Public IP address or FQDN of the external load balancer***, put in the VIP address of the connection broker virtual service.

7. Set *Method for routing display protocol traffic through this Leostream gateway* to **From random gateway port to protocol-specific desktop port**.
8. Click **Save** to commit the changes.

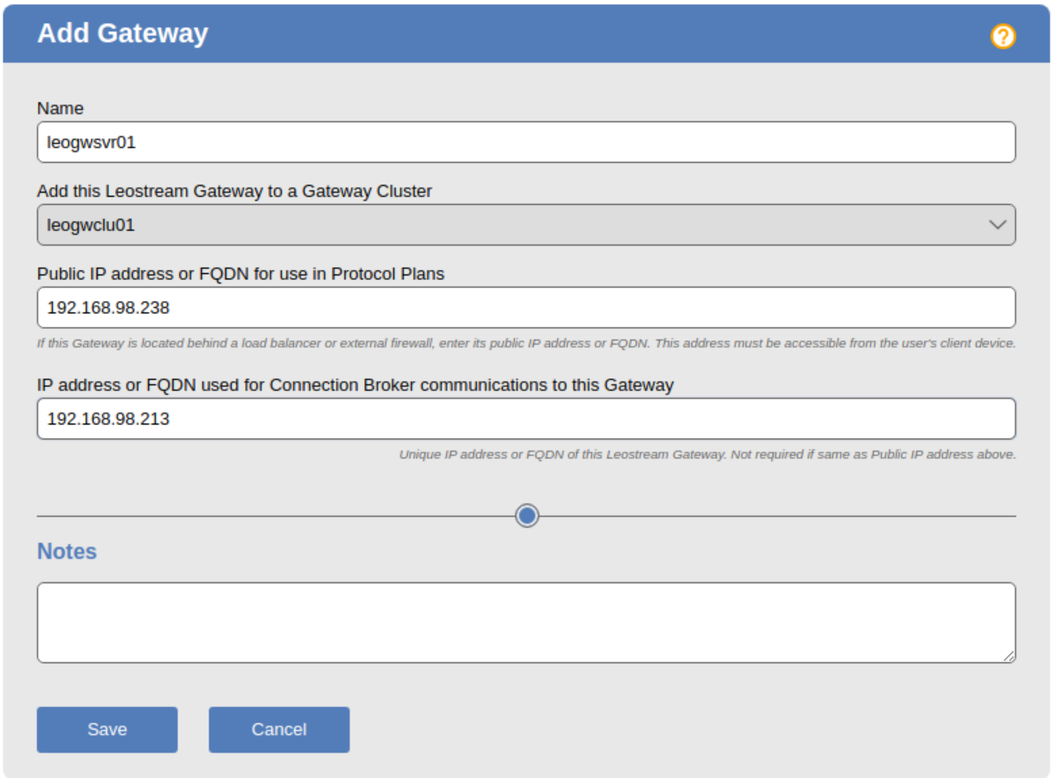


9. On the *Gateways* page, click **Add Gateway**.

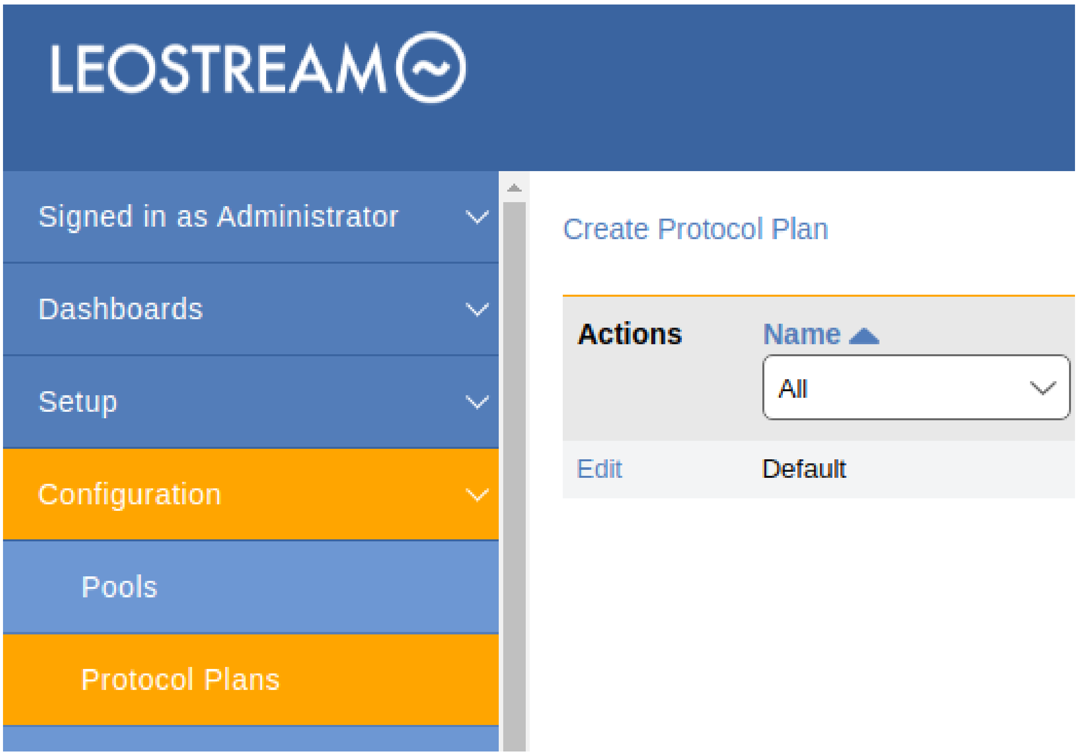


10. Select the gateway cluster created in the previous step from the drop-down list.

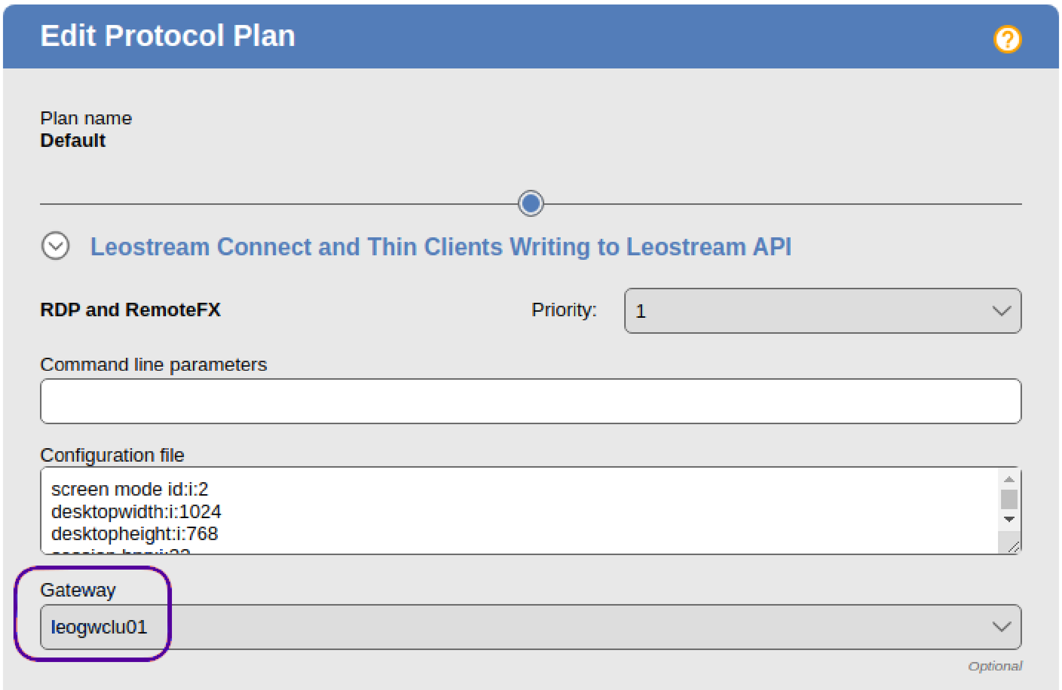
11. Set *Public IP address or FQDN for use in Protocol Plans* as the VIP address of the gateway virtual service.
12. Set *IP address or FQDN used for Connection Broker communications to this Gateway* as real server's own IP address / FQDN.
13. Click **Save** to commit the changes.



14. Repeat the *Add Gateway* process to add additional Leostream Gateways as required.
15. From the left hand menu, navigate to *Configuration > Protocol Plans*.
16. Click on **Edit** next to the *Default* plan.



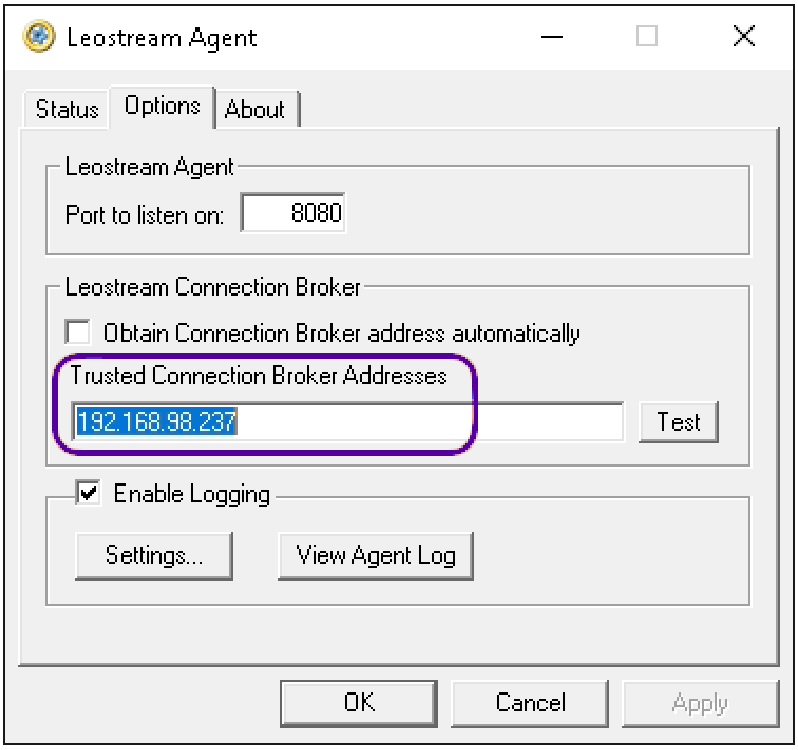
17. For each protocol in use, set the **Gateway** to the gateway cluster created previously.



18. Save the changes.

8.6. Leostream Agent Configuration

For each Leostream agent installed, the agent should be configured with either the connection broker VIP address **or** the (solo) connection broker's IP address / FQDN in a *non-clustered environment*. This should be set as the **Trusted Connection Broker Address**, like so:

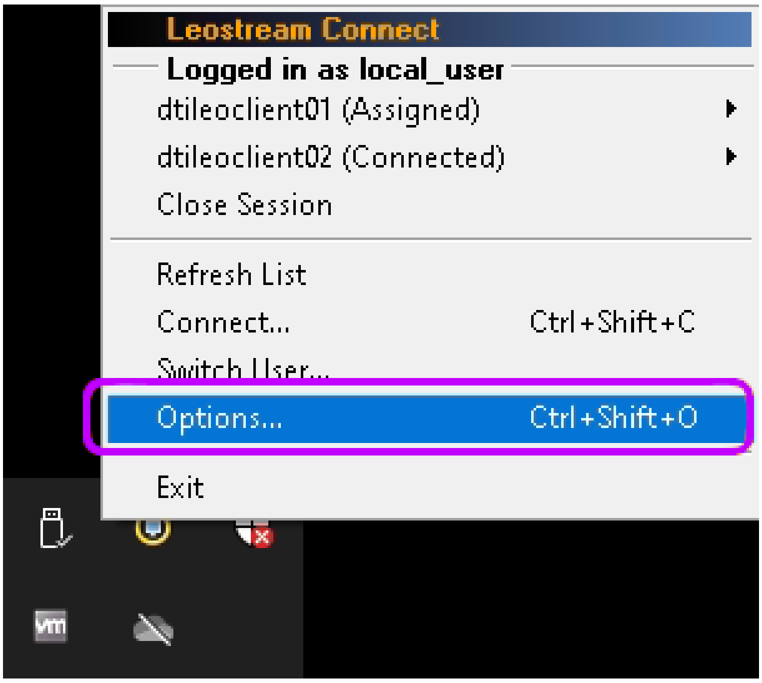


8.7. Leostream Connect Client Configuration

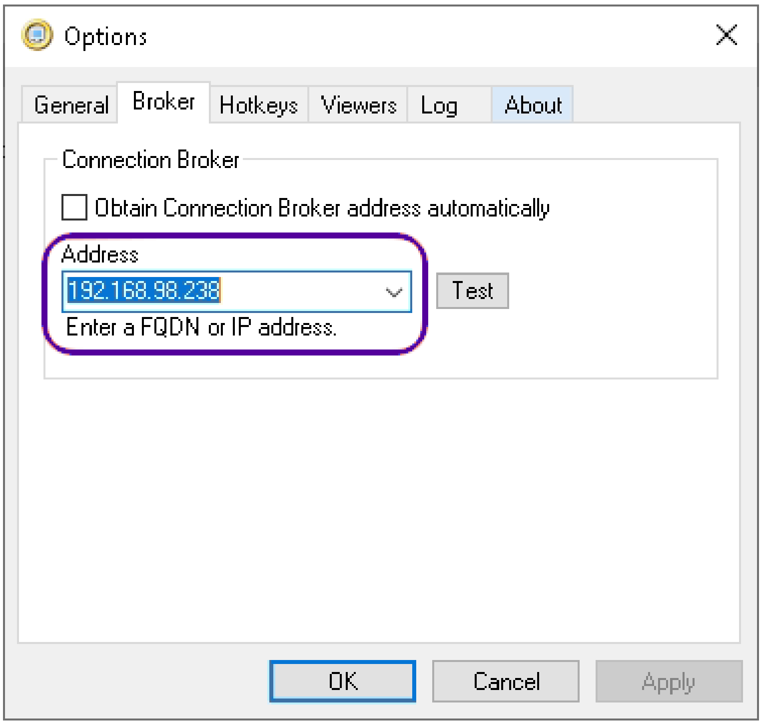
Leostream Connect clients must be configured as described below, depending on the specific platform in use.

8.7.1. Windows Clients

1. Open the Leostream Connect client.
2. Right-click on the Leostream icon in the Windows taskbar and click on **Options...**



3. Click on the *Broker* tab and set the **Address** to the VIP address of the gateway service.



8.7.2. Java Clients

1. Navigate to the location of (directory that contains) the LeostreamConnect.jar file.
2. Create or edit a file named lc.conf that contains the following minimum contents:

```
trace_level=ERROR,WARN,INFO,TRACE,DIAG
rdp_path=/usr/bin/remmina
connection_broker_auto_discovery=false
recent_brokers=192.168.98.231
connection_broker_address=<GW_VIP_L4>
```

where GW_VIP_L4 is the VIP address of the gateway service.

Note	Set the rdp_path variable to the location of the preferred RDP client.

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.

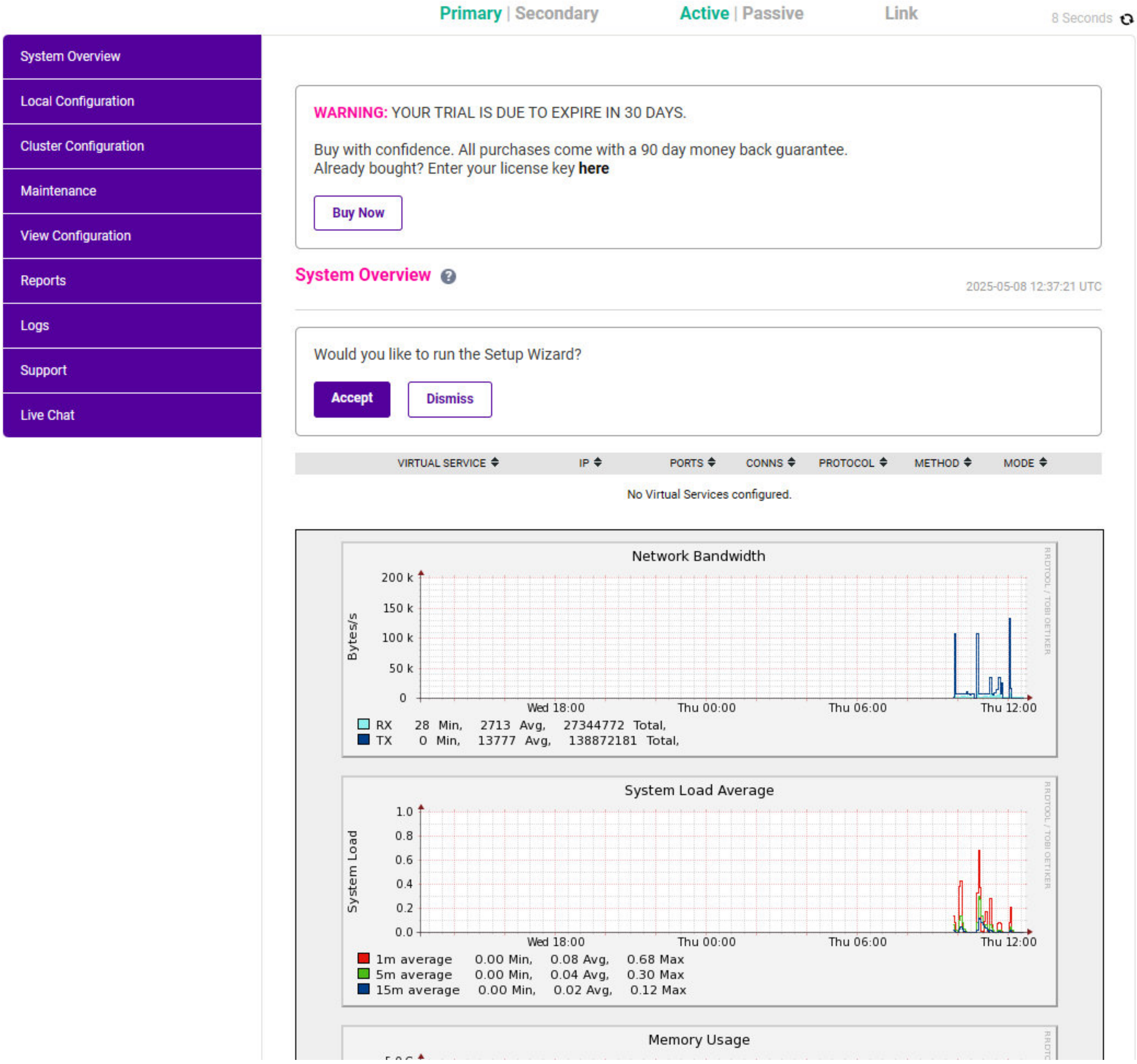
Note	If you need to change the port, IP address or protocol that the WebUI listens on, please refer to Service Socket Addresses.

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

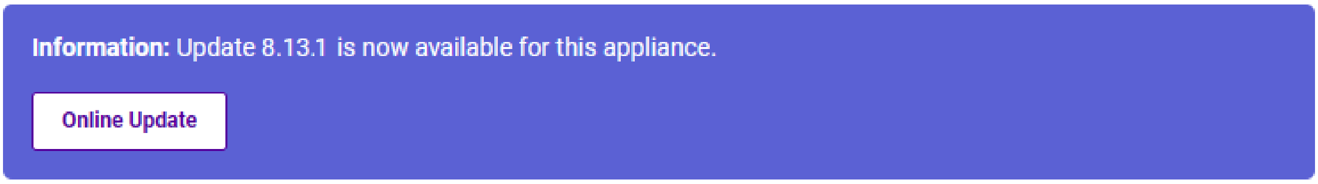
We recommend that the appliance is kept up to date to ensure that you benefit from the latest bug fixes, security updates and feature improvements. Both online and offline update are supported.

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

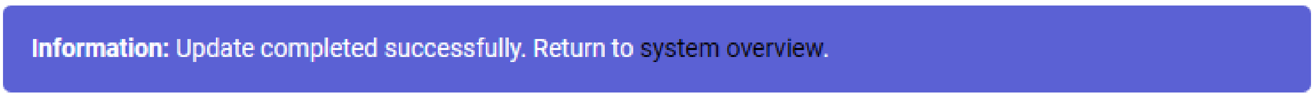
The appliance periodically contacts the Loadbalancer.org update server (**update.loadbalancer.org**) and checks for updates. This is the default behavior and can be disabled if preferred. If an update is found, a notification similar to the example below will be displayed at the top of the WebUI:



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.

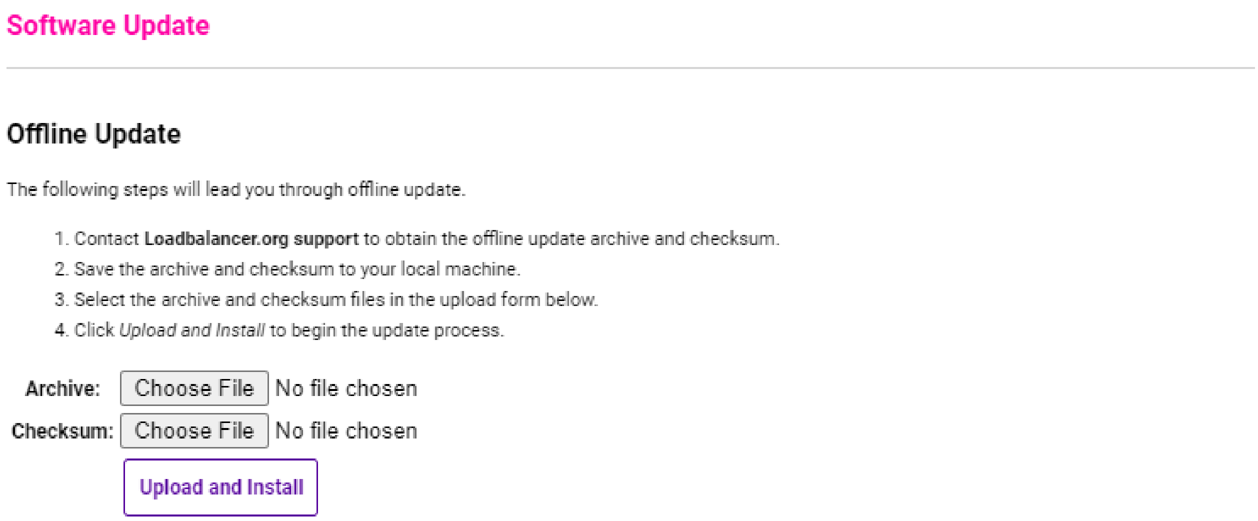
9.4.2. Offline Update

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

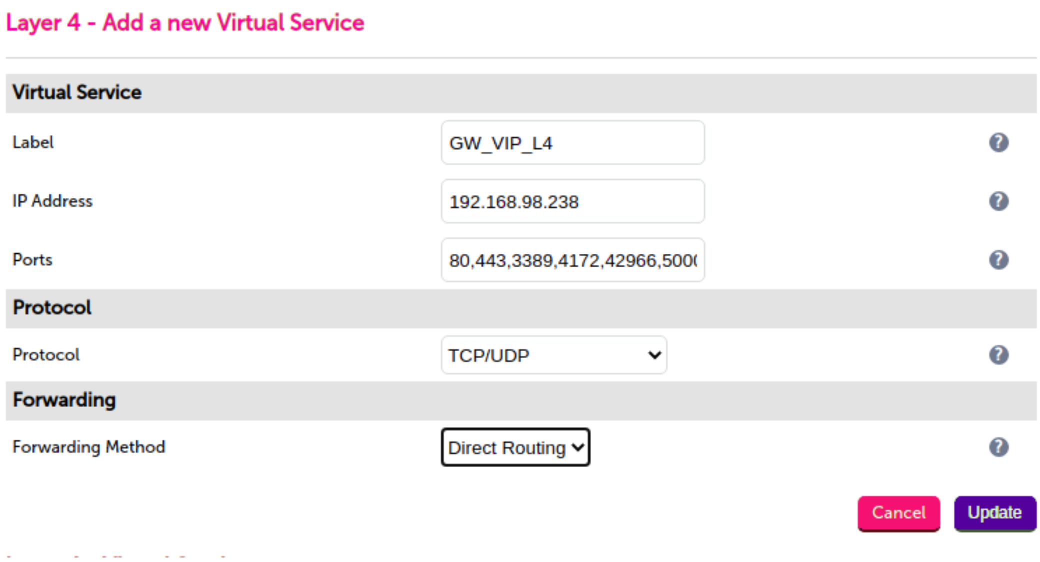
Loadbalancer.org recommend that load balancer appliances are deployed in pairs for high availability. In this guide a single unit is deployed first, adding a secondary unit is covered in the section Configuring HA - Adding a Secondary Appliance of the appendix.

10. Appliance Configuration for Leostream - Using Layer 4 DR Mode

10.1. Configuring VIP 1 - Leostream Gateway Service

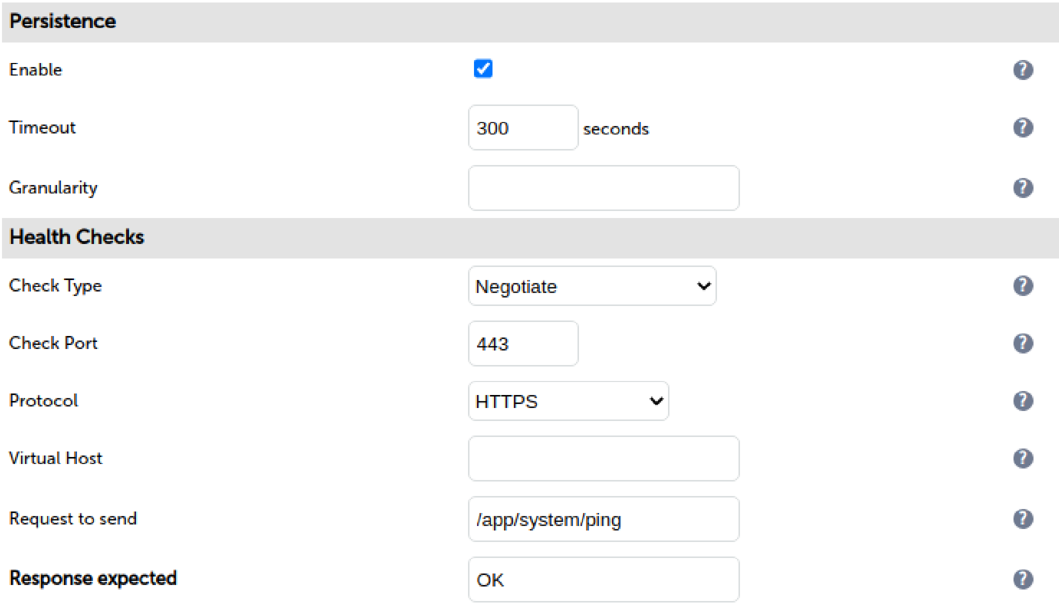
10.1.1. Configuring the Virtual Service (VIP)

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. **GW_VIP_L4**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.98.238**.
4. Set the *Ports* field to cover the remote desktop protocols in use, e.g. **80,443,3389,4172,42966,50001,50002**.
5. Set the *Protocol* to **TCP/UDP**.
6. Leave the *Forwarding Method* set to **Direct Routing**.
7. Click **Update** to create the virtual service.



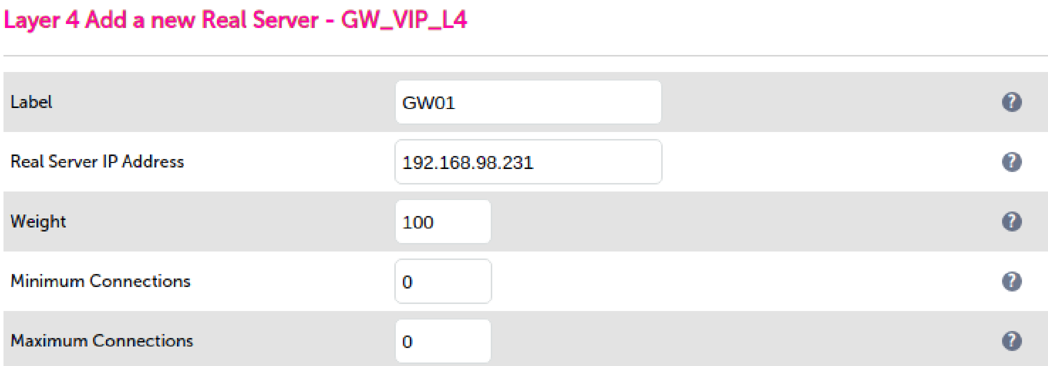
8. Click **Modify** next to the newly created VIP.
9. Ensure that the *Persistence Enable* checkbox is checked.
10. Set the *Health Checks Check Type* to **Negotiate**.
11. Set the *Check Port* to **443**.

12. Set the *Protocol* to **HTTPS**.
13. Set the *Request to send* to **/app/system/ping**
14. Set the *Response expected* to **OK**
15. Click **Update**.



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. **GW01**.
3. Set the *Real Server IP Address* field to the required IP address, e.g. **192.168.98.231**.
4. Click **Update**.
5. Repeat these steps to add additional Leostream Gateways as real servers as required.



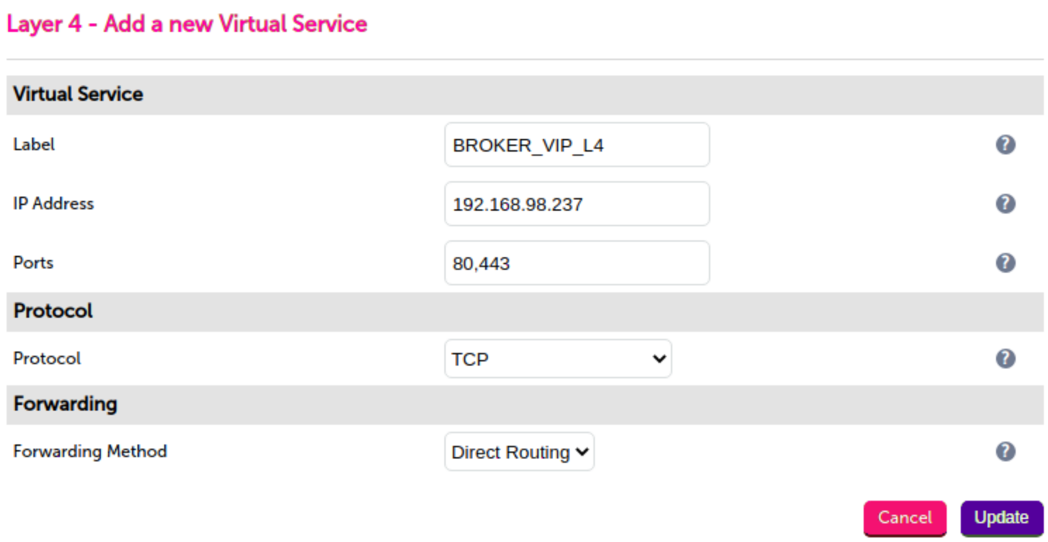
10.2. Configuring VIP 2 - Leostream Connection Broker Service

Important

This virtual service should **only** be configured in a deployment with multiple, clustered Leostream Connection Brokers. If operating with a **single** Leostream Connection Broker then skip setting up this service.

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. **BROKER_VIP_L4**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.98.237**.
4. Set the *Ports* field to **80,443**.
5. Set the *Protocol* to **TCP**.
6. Set the *Forwarding Method* to **Direct Routing**.
7. Click **Update** to create the virtual service.



8. Click **Modify** next to the newly created VIP.
9. Ensure that the *Persistence Enable* checkbox is checked.
10. Set the *Health Checks Check Type* to **Negotiate**.
11. Set the *Check Port* to **443**.
12. Set the *Protocol* to **HTTPS**.
13. Set the *Request to send* to **/index.pl?action=cb_status**
14. Set the *Response expected* to **CB_IS_OK**
15. Click **Update**.

Persistence		
Enable	☑	
Timeout	300 seconds	
Granularity		

Health Checks	
Check Type	Negotiate
Check Port	443
Protocol	HTTPS
Virtual Host	
Request to send	/index.pl?action=cb_status
Response expected	CB_IS_OK

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. **BRK01**.
3. Set the ***Real Server IP Address*** field to the required IP address, e.g. **192.168.98.230**.
4. Click **Update**.
5. Repeat these steps to add additional Leostream Connection Brokers as real servers as required.

Layer 4 Add a new Real Server - BROKER_VIP_L4

Label	BRK01
Real Server IP Address	192.168.98.230
Weight	100
Minimum Connections	0
Maximum Connections	0

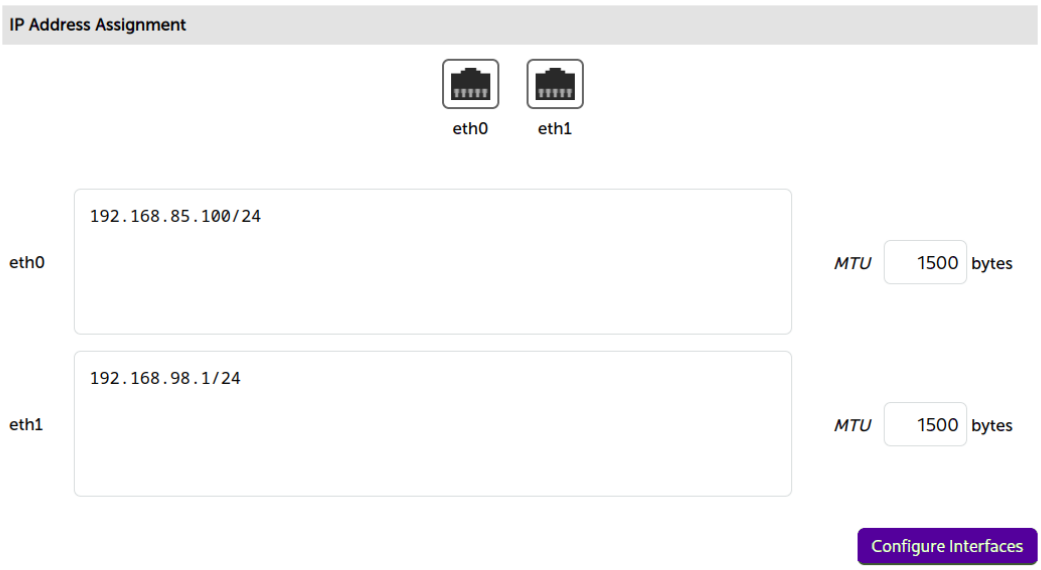
11. Appliance & Server Configuration for Leostream - Using Layer 4 NAT Mode

11.1. Configure the Load Balancer's Network Interfaces

Layer 4 NAT mode is typically used in a 2-arm configuration where the VIP is located in one subnet and the load balanced real servers are located in another. This can be achieved by using two network adapters, or by creating VLANs on a single adapter. Single arm configuration is also supported under certain conditions - for more information please refer to Layer 4 NAT Mode.

To configure an additional network interface for a 2-arm configuration:

1. Using the WebUI, navigate to *Local Configuration > Network Interface Configuration*.
2. Scroll to the *IP Address Assignment* section.



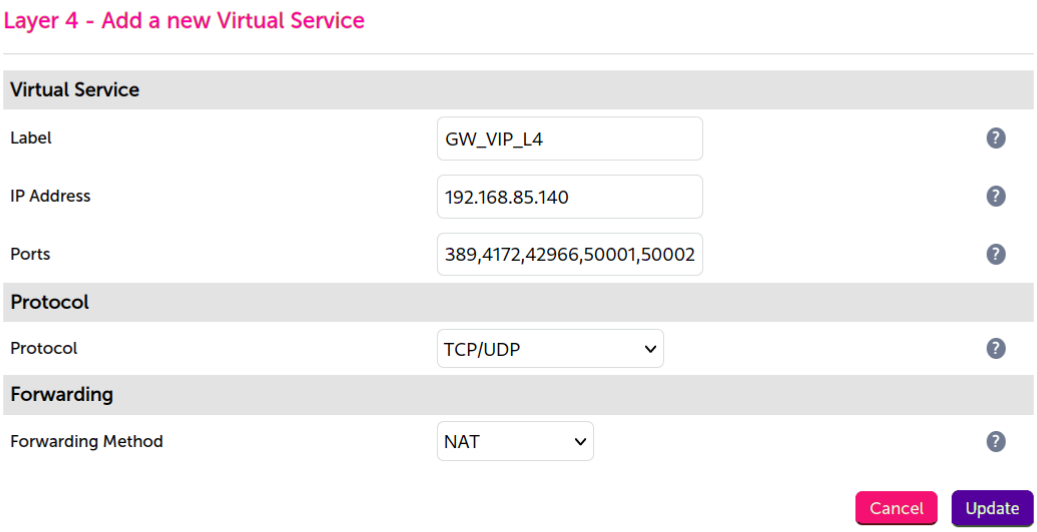
3. Specify an appropriate IP address for **eth1** in CIDR format as shown above.
4. Click **Configure Interfaces**.

Note	There are no restrictions on which interface is used for each requirement.

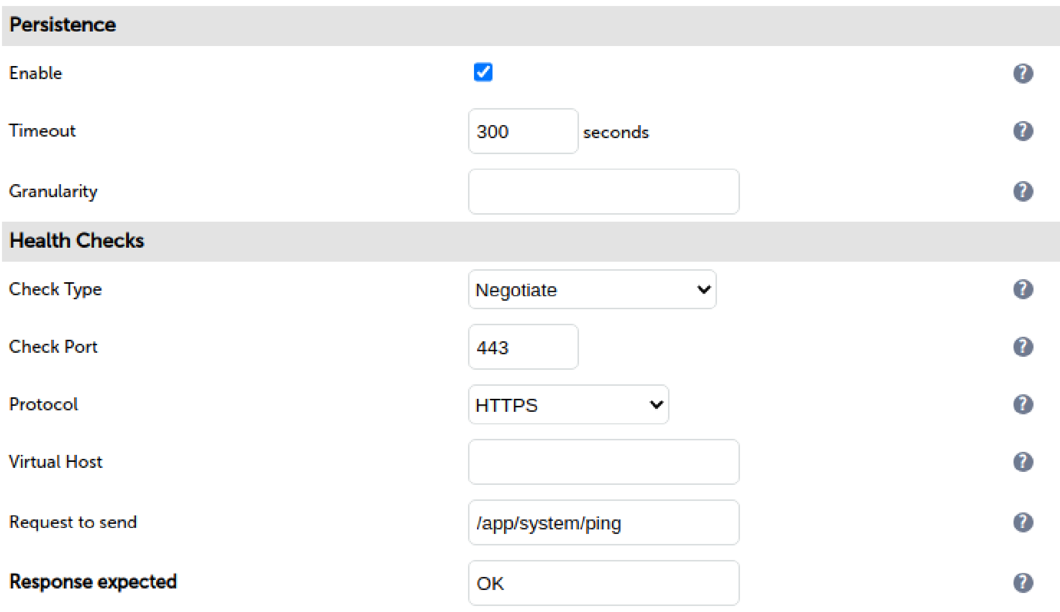
11.2. Configuring VIP 1 - Leostream Gateway Service

11.2.1. 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. **GW_VIP_L4**.
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 cover the remote desktop protocols in use, e.g. **80,443,3389,4172,42966,50001,50002**.
5. Set the *Protocol* to **TCP/UDP**.
6. Leave the *Forwarding Method* set to **NAT**.
7. Click **Update** to create the virtual service.



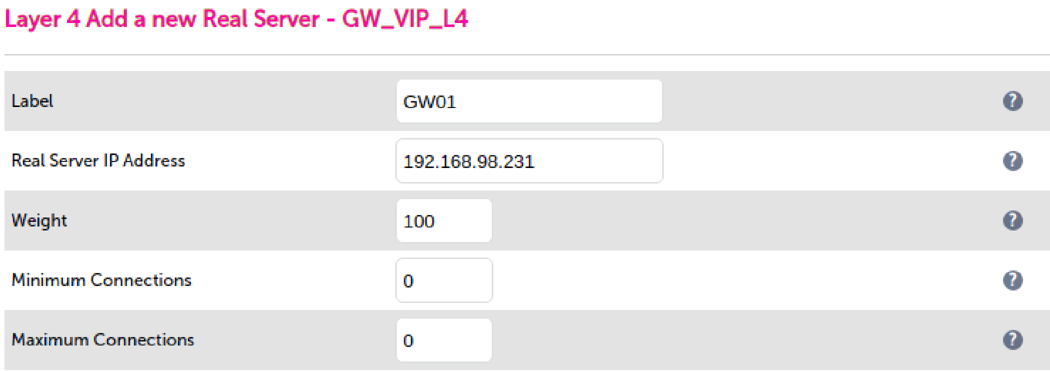
8. Click **Modify** next to the newly created VIP.
9. Ensure that the *Persistence Enable* checkbox is checked.
10. Set the *Health Checks Check Type* to **Negotiate**.
11. Set the *Check Port* to **443**.
12. Set the *Protocol* to **HTTPS**.
13. Set the *Request to send* to **/app/system/ping**
14. Set the *Response expected* to **OK**
15. Click **Update**.



11.2.2. 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. **GW01**.

3. Set the *Real Server IP Address* field to the required IP address, e.g. **192.168.98.231**.
4. Click **Update**.
5. Repeat these steps to add additional Leostream Gateways as real servers as required.

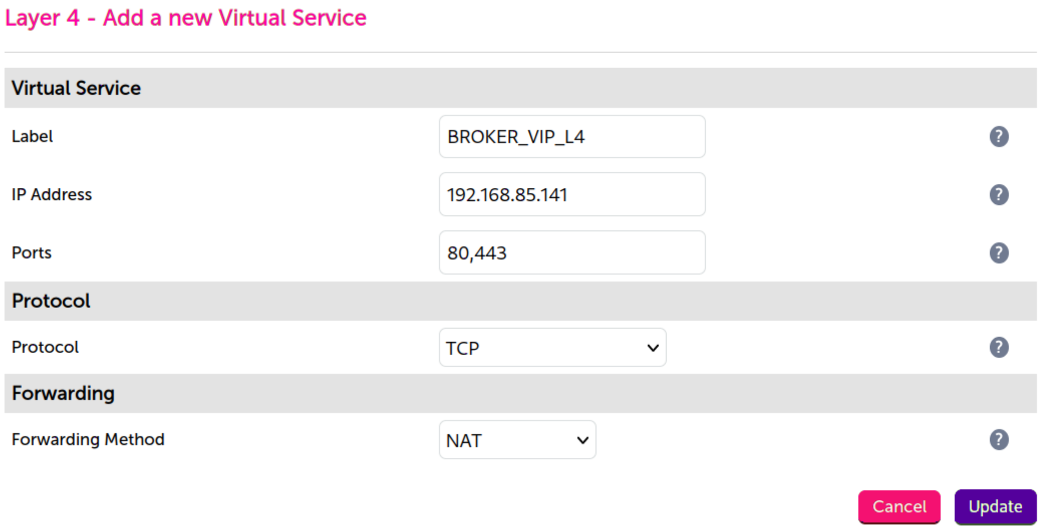


11.3. Configuring VIP 2 - Leostream Connection Broker Service

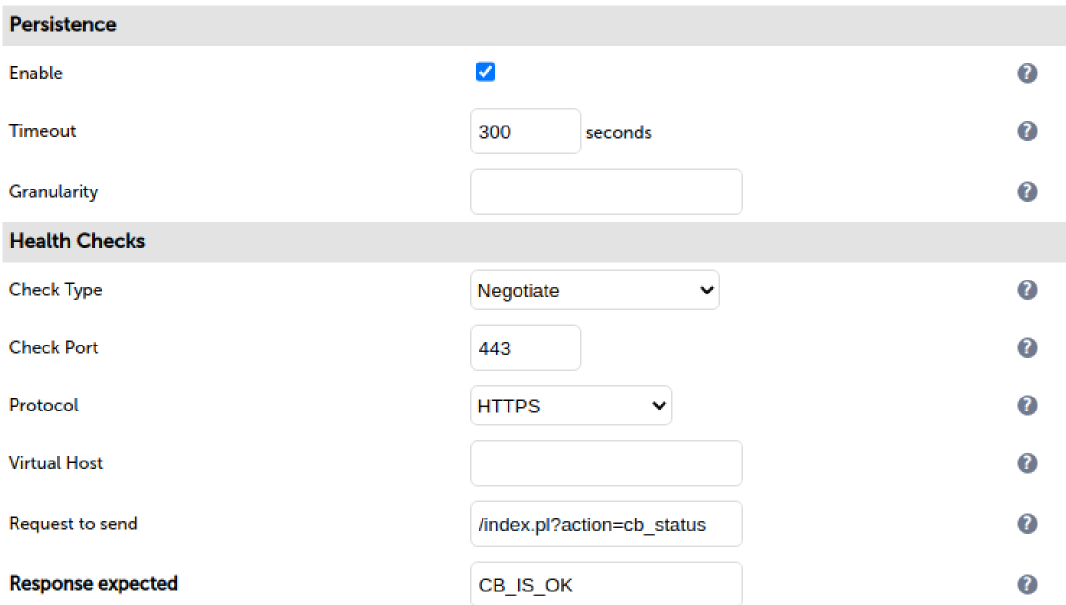
> Important
>
> This virtual service should **only** be configured in a deployment with multiple, clustered Leostream Connection Brokers. If operating with a **single** Leostream Connection Broker then skip setting up this service.

11.3.1. 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. **BROKER_VIP_L4**.
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 **80,443**.
5. Set the *Protocol* to **TCP**.
6. Set the *Forwarding Method* to **NAT**.
7. Click **Update** to create the virtual service.



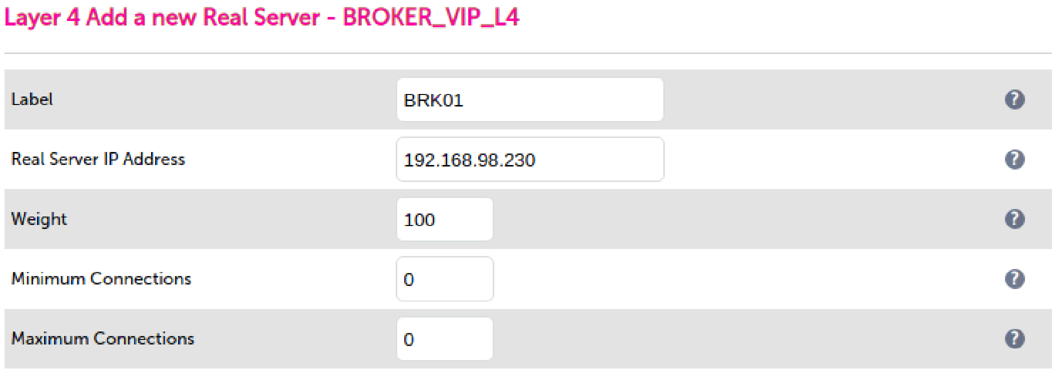
8. Click **Modify** next to the newly created VIP.
9. Ensure that the *Persistence Enable* checkbox is checked.
10. Set the *Health Checks Check Type* to **Negotiate**.
11. Set the *Check Port* to **443**.
12. Set the *Protocol* to **HTTPS**.
13. Set the *Request to send* to **/index.pl?action=cb_status**
14. Set the *Response expected* to **CB_IS_OK**
15. Click **Update**.



11.3.2. 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. **BRK01**.

3. Set the *Real Server IP Address* field to the required IP address, e.g. **192.168.98.230**.
4. Click **Update**.
5. Repeat these steps to add additional Leostream Connection Brokers as real servers as required.



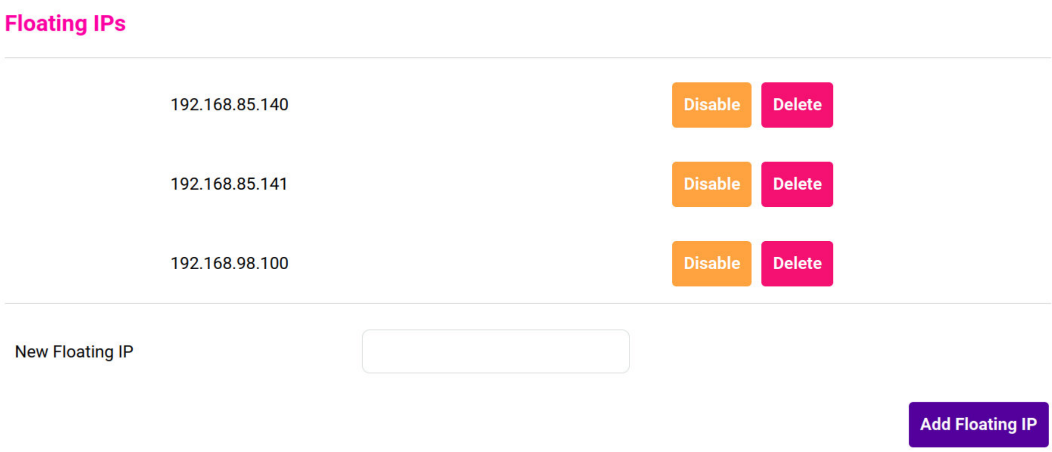
11.4. Create a Floating IP to Use for the Leostream Servers' Default Gateway

The default gateway on each Leostream server must be configured to be an IP address on the load balancer. It's possible to use the IP address assigned to the internal facing interface (**eth1** in this example) for the default gateway, although it's recommended that an additional floating IP is created for this purpose. This is required if two load balancers (our recommended configuration) are used. In this scenario if the primary unit fails, the floating IP will be brought up on the secondary.

11.4.1. To Create a Floating IP Address on the Load Balancer

1. Using the WebUI, navigate to: *Cluster Configuration > Floating IPs*.
2. Enter the required IP address to be used for the default gateway, e.g. **192.168.98.100**.
3. Click **Add Floating IP**.

   Once added, there will be multiple floating IPs: one for each virtual service (**192.168.85.140** and **192.168.85.141**, in the example presented here) and one for the default gateway (e.g. **192.168.98.100**) as shown below:



11.5. Leostream Server Configuration

11.5.1. Default Gateway

To ensure that return traffic passes back to the client via the load balancer, set the default gateway of each Leostream server (gateways and, if being load balanced, connection brokers) to be the floating IP address added in the previous step, in this example **192.168.98.100**.

⚠ Warning	The default gateway changes must be **permanent**, otherwise the changes will be lost on reboot and the virtual service(s) will cease to function.

ℹ Note	For more information about NAT mode, please refer to Layer 4 NAT Mode.

12. Testing & Verification

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

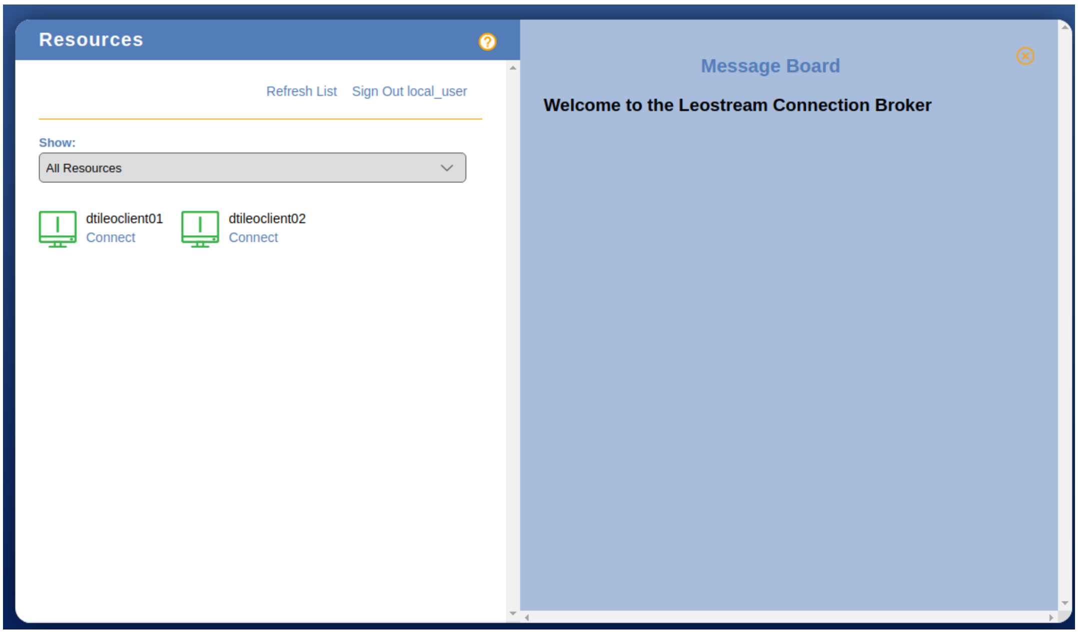
12.1. Testing the Load Balanced Gateway Service

The load balanced Leostream gateway service can be tested by using it.

1. Use SSH to connect to both Leostream gateway hosts as the root user.
2. Execute the command leostream --conn to view current connections.

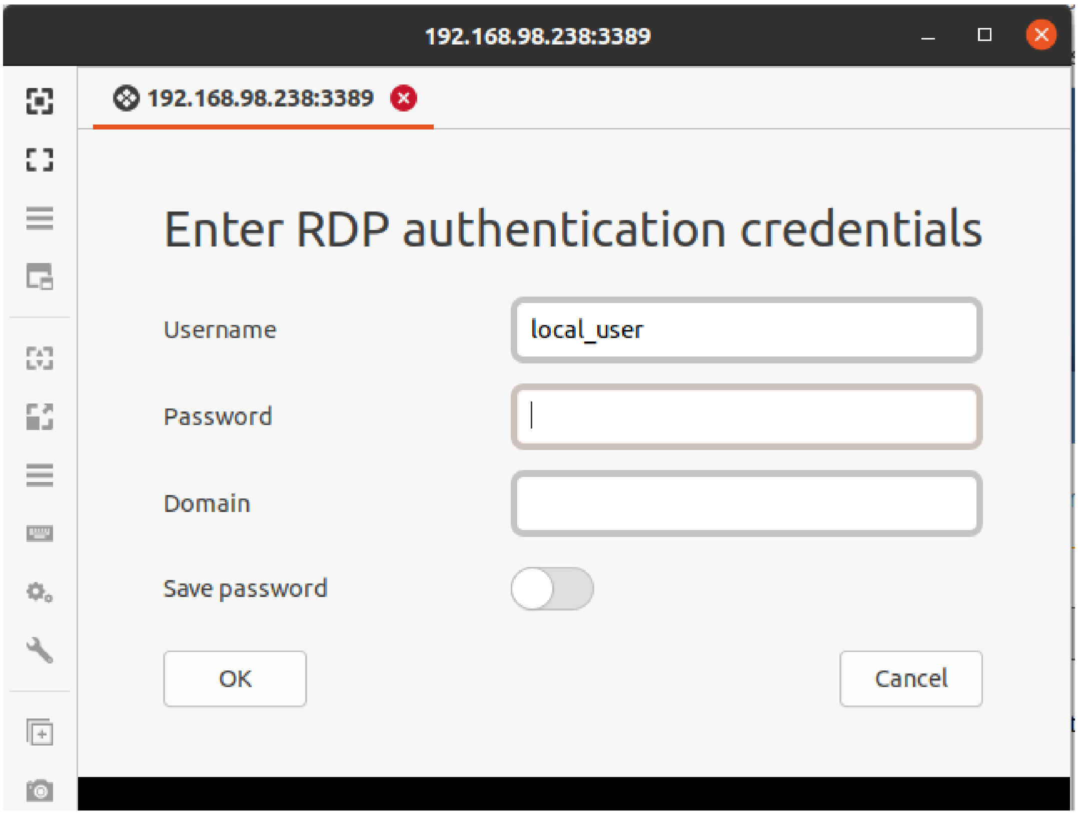
```
[root@localhost ~]# leostream-gateway --conn
No connections
```

3. Use a web browser to connect to the Leostream gateway virtual service and log in using appropriate authorised credentials.

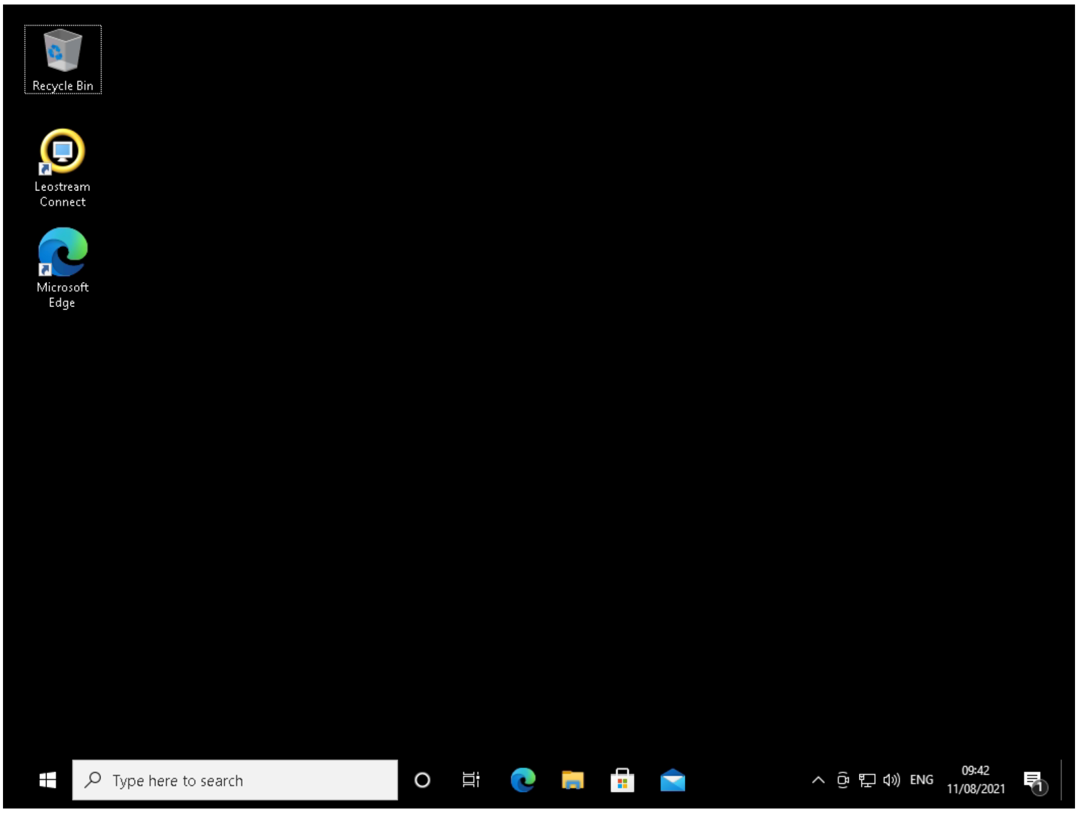


4. Select a client to connect to.

5. Open the downloaded file in an RDP client and enter appropriate credentials (if SSO isn't enabled).



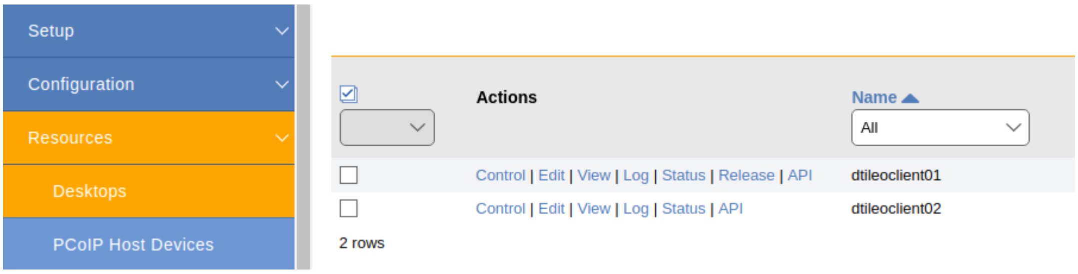
6. A connection should be successfully established to the remote client, via the gateway virtual service configured on the load balancer.



7. On the Leostream gateway hosts, re-execute the command leostream --conn and the active connection should be listed.

```
[root@localhost ~]# leostream-gateway --conn
Desktop          Port  Dport Source address  Key
---------------- ----- ----- --------------- --------------------------------
192.168.98.235    3389  3389 192.168.65.185  de8e9319ea66b1dcb2579ec285f17bb4
```

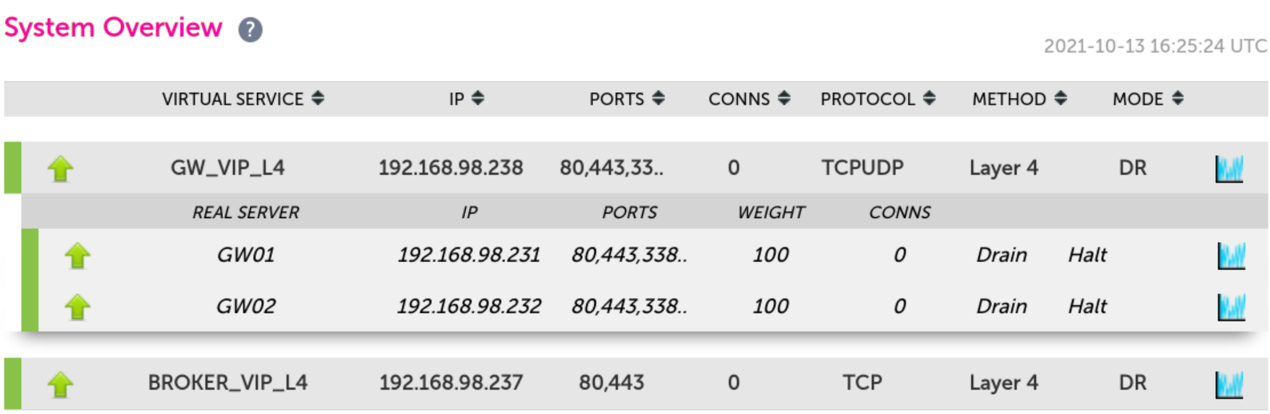
8. Use a web browser to connect to the Leostream connection broker service (if configured).
9. In the menu on the left, navigate to ***Resources > Desktops***.
10. A ***Release*** option should be visible next to the client that has been connected to.



11. Repeat these tests using Leostream connection clients, if applicable.

12.2. Using System Overview

The System Overview can be viewed in the WebUI. It shows a graphical view of all VIPs & RIPs (i.e. the web 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 DR mode deployment where both Leostream 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:

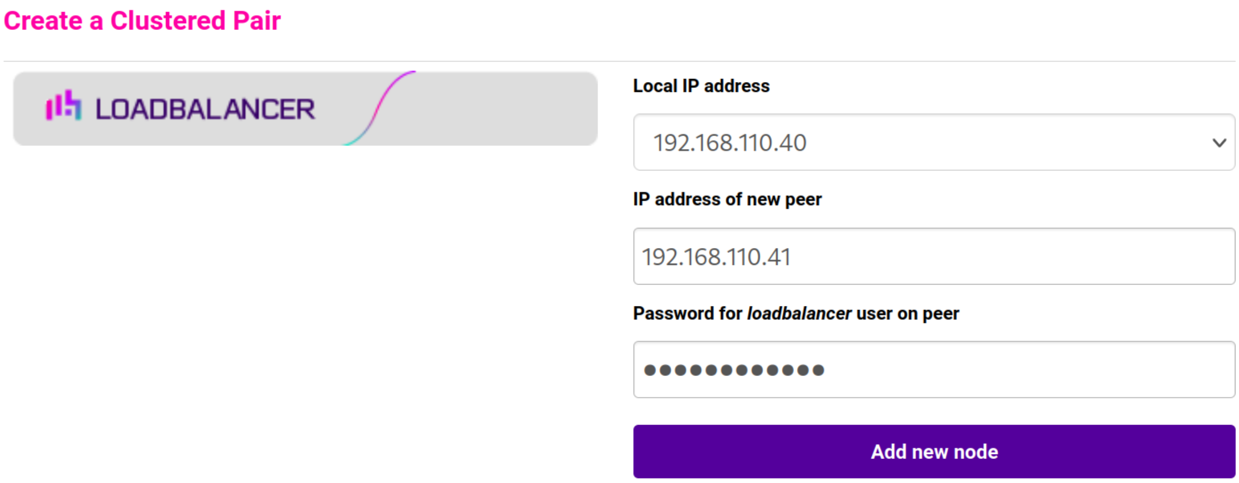
WebUI Main Menu Option	Sub Menu Option	Description
Local Configuration	Hostname & DNS	Hostname and DNS settings
Local Configuration	Network Interface Configuration	Interface IP addresses, bonding configuration and VLANs
Local Configuration	Routing	Default gateways and static routes
Local Configuration	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.

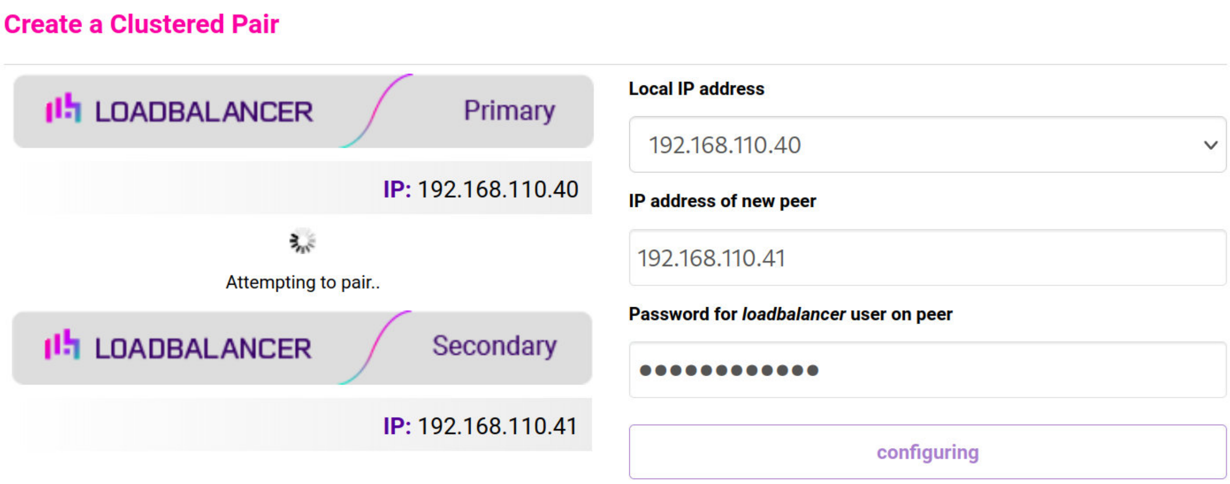
15.1.2. Configuring the HA Clustered Pair

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.

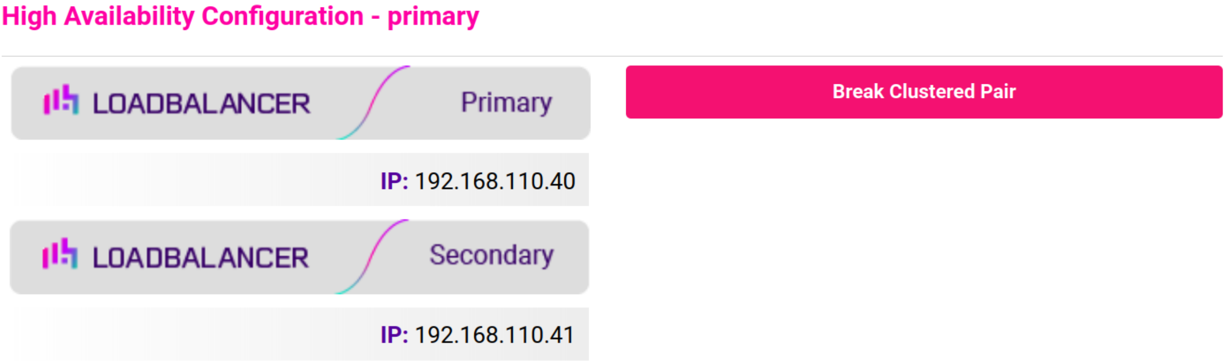
1. Deploy a second appliance that will be the Secondary and configure initial network settings.
2. Using the WebUI on the Primary appliance, navigate to: *Cluster Configuration > High-Availability Configuration*.



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
```

ℹ 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
```

ℹ 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	1 Jan 2021	Initial version		DT, AH
1.1.0	26 May 2022	Added NAT mode deployment method	NAT mode validated as a working deployment method	AH
1.1.1	5 January 2023	Combined software version information into one section Added one level of section numbering Added software update instructions Added table of ports used by the appliance Reworded 'Further Documentation' section	Housekeeping across all documentation	AH
1.1.2	2 February 2023	Updated screenshots	Branding update	AH
1.1.3	7 March 2023	Removed conclusion section	Updates across all documentation	AH
1.2.0	24 March 2023	New document theme Modified diagram colours	Branding update	AH

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.