



## **Appliance Administration Manual**

**v7.2**



This document covers all required administration information for Loadbalancer.org appliances

*Copyright © 2002 - 2012 Loadbalancer.org, Inc.*

# Table of Contents

Section A – Introduction.....	7
Appliance Details.....	8
Version 7.x.....	8
Initial Configuration.....	8
Additional Information.....	8
Deployment Guides.....	9
Section B – Load balancing Concepts.....	10
Load Balancing Algorithms.....	11
Weighted Round Robin.....	11
Weighted Least Connection.....	11
Destination Hashing.....	11
Agent Based.....	11
Layer 4 vs Layer 7.....	12
Our recommendation.....	12
Section C - Quick Start Guide.....	13
Loadbalancer.org terminology.....	14
What is a Virtual IP address?.....	14
What is a Floating IP address?.....	14
What are Your Objectives?.....	15
What Is The Difference Between a One-Arm and a Two-Arm Configuration?.....	16
What Are The Different Load Balancing Methods Supported?.....	16
Direct Routing (DR).....	18
Network Address Translation (NAT).....	19
Source Network Address Translation (SNAT) .....	20
High-Availability Configuration of Two Loadbalancer.org Appliances.....	21
Clustered Pair Configuration Methods.....	21
Using the Wizard.....	21
Manual configuration.....	21
Unpacking and setting up the Loadbalancer.org appliance.....	22
Configuring The Loadbalancer.org Appliance Using The Web Based Wizard.....	23
Network interface configuration.....	23
Accessing the Web User Interface (WUI).....	23
Example answers using the wizard for a two-arm NAT configuration (single unit).....	24
Additional Appliance Configuration Using The Web Interface.....	25
Adding additional real servers.....	26
Configuring the Real Servers.....	27
Configuring the real servers for NAT mode.....	27
Configuring the real servers for DR mode (Linux).....	27
Detecting the ARP problem.....	27
Solving for Linux – method 1 (using iptables).....	27
Solving for Linux – method 2 (using arp_ignore sysctl values).....	28
Configuring the real servers for DR mode (Windows).....	29
Configuring IIS to respond to both the RIP and VIP.....	29
Resolving ARP issues for Windows server 2000 / 2003 (DR mode only).....	30
Installing the Microsoft loopback adapter.....	30
Configuring the loopback adapter.....	31
Resolving ARP issues for Windows server 2008 (DR mode only).....	33
Installing the Microsoft loopback adapter.....	33
Configuring the loopback adapter.....	34
Configuring strong / weak host behavior.....	35
Verifying netsh Settings.....	36
Configuring the real server for SNAT mode.....	37
IPv6 Support.....	37
Testing The Load Balancer Configuration.....	38
Connection error diagnosis.....	38
Health check diagnosis.....	39
Testing high-availability for a Loadbalancer.org Clustered Pair.....	40

Does Your Application Cluster Correctly Handle Its Own State?	41
Replication solutions for shared data	41
Solutions for session data	41
Persistence	41
What do you do if your application is not stateless?	42
Loadbalancer.org persistence methods	42
<b>Section D – Typical Deployment Examples</b>	<b>43</b>
Example 1 – One-Arm DR Mode (Single Appliance)	44
Initial network interface configuration	44
Accessing the Web User Interface (WUI)	44
Configuring the load balancer (using the WUI)	44
Configuration overview	44
Network settings	45
Virtual server (VIP)	46
Real servers (RIP)	46
Real server changes - solve the ARP problem	47
Basic testing & verification	47
Example 2 – Two-Arm NAT Mode (Clustered Pair)	48
Initial network interface configuration	48
Accessing the Web User Interface (WUI)	48
Configuring the load balancer (using the WUI)	48
Configuration overview	48
Slave unit – network settings	49
Master unit – network settings	50
Master & Slave - heartbeat settings	52
Virtual server (VIP)	52
Real servers (RIP)	53
Real server changes – Set the Default Gateway	53
Verify the Slave configuration	54
Restart Heartbeat	54
Basic testing & verification	54
Example 3 - One-Arm SNAT Mode With SSL - HAProxy & Pound (Single Unit)	55
Initial network interface configuration	55
Accessing the Web User Interface (WUI)	55
Configuring the load balancer (using the WUI)	55
Configuration overview	55
Network settings	56
Virtual server (VIP)	57
Real servers (RIP)	57
SSL termination	58
Basic testing & verification	59
<b>Section E – Detailed Configuration Information</b>	<b>60</b>
Appliance Configuration Methods	61
Console access	61
Console access via a serial cable	62
Remote configuration methods	62
Network Configuration	64
IP addresses	64
Setting IP addresses	64
Setting multiple addresses	65
Configuring bonding	66
Bonding configuration modes	67
Example 1: bonding for bandwidth	67
Example 2: bonding for high-availability (the Default mode)	67
Example 3: bonding for high-availability & bandwidth	67
Configuring VLANs	68
Default gateway	69
Hostname & DNS configuration	69
Advanced DR Considerations	70
What Is the ARP problem?	70
Detecting the ARP problem	70
Solving for Linux – method 1 (using iptables)	70

Solving for Linux – method 2 (using arp_ignore sysctl values).....	71
Solving for Solaris.....	72
Solving for Mac OS X or BSD.....	72
Solving for Windows 2000 / 2003.....	73
Windows 2003 R2 / R1 (With SP1) firewall settings.....	75
Solving for Windows 2008.....	76
Windows 2008 R2 firewall settings.....	80
Windows 2008 R1 firewall settings.....	80
Configuring IIS to respond to both the RIP and VIP.....	81
Advanced NAT considerations.....	82
Explaining the RIP & VIP in NAT mode.....	84
Route configuration for Windows Server with one-arm NAT mode.....	86
Route configuration for Linux with one-arm NAT mode.....	86
Advanced Layer 7 Considerations.....	87
Load balancing based on URL match with HAProxy.....	87
HAProxy error codes.....	88
Configuring VIPs & RIPs via Command Line / Script.....	89
Layer 4.....	89
Layer 7.....	90
SSL Termination.....	91
Certificate on the real servers.....	91
Certificate on the load balancer.....	91
Creating a new certificate using a CSR.....	92
Using an existing certificate.....	93
Creating a PEM File.....	93
Adding an Intermediate certificate.....	94
Windows servers.....	95
Import a certificate exported from Windows server.....	96
Converting an encrypted private key to an unencrypted key.....	96
Limiting ciphers.....	96
Using Tproxy.....	97
Health Monitoring.....	99
Load balancer health (clustered pair).....	99
Heartbeat communication method.....	99
Serial cable.....	99
Unicast (ucast).....	100
Broadcast (bcast).....	100
Ping node.....	100
Auto-failback.....	100
Real server health.....	101
Layer 4.....	101
Layer 7.....	105
Fallback server settings.....	106
Advanced firewall considerations.....	108
Firewall marks (Layer 4).....	108
Layer 4 Persistence Considerations.....	111
Persistence state table replication.....	111
Server maintenance when using persistence.....	111
SNMP reporting.....	112
SNMP for layer 4 based services.....	112
SNMP for layer 7 based services.....	112
Feedback agents.....	113
Installing the Windows agent.....	113
Installing the Linux/Unix agent.....	114
Custom HTTP agent.....	114
Changing the local date, time & time zone.....	115
NTP configuration.....	115
Restoring Manufacturer's Settings.....	116
From the console.....	116
From the WUI.....	116
Force Master/Slave Take-Over In A Clustered Pair.....	116
Force the slave to become active & master passive.....	116
Force the master to become active & slave passive.....	116

Active / Active load balancer configuration (Layer 7 - Haproxy).....	117
Application Specific Settings.....	118
FTP.....	118
Changing the FTP port in NAT mode.....	118
FTP negotiate health check.....	119
FTP recommended persistence settings.....	120
Limiting passive FTP ports.....	120
For Linux.....	120
For Windows 2008.....	121
For Windows 2003.....	122
For Windows 2000.....	122
Terminal Services & RDP.....	123
Layer 4 – IP persistence.....	123
Layer 7 - RDP cookies.....	123
Layer 7 – Microsoft Connection Broker / Session Directory.....	124
Software Updates.....	125
Online Update.....	125
Offline Update.....	126
Updating a Clustered Pair.....	126
Adding a Slave Unit after the master has been configured.....	127
Verifying Master / Slave replication & testing failover.....	128
<b>Section F – Disaster Recovery.....</b>	<b>130</b>
Being prepared.....	131
Backing up to a remote location.....	131
Using wget to copy the files.....	131
Backing up to the load balancer.....	131
Appliance recovery using a USB memory stick.....	132
Disaster recovery after master failure.....	133
Disaster recovery after slave failure.....	135
Option 1 – Using the XML Backup.....	135
Option 2 – Synchronizing from the Master.....	136
<b>Section G – Web User Interface Reference.....</b>	<b>137</b>
System Overview.....	138
View Configuration.....	138
XML.....	138
Layer 4.....	138
Layer 7.....	138
SSL termination.....	138
Heartbeat configuration.....	138
Heartbeat resources.....	138
Network configuration.....	138
Routing table.....	138
Firewall rules.....	139
Edit Configuration.....	139
Layer 4 – virtual servers.....	139
Adding a virtual server.....	140
Modifying a virtual server.....	141
Layer 4 – real servers.....	143
Adding / modifying a real server.....	143
Layer 4 – advanced configuration.....	145
Layer 7 - virtual servers.....	147
Adding a virtual server.....	148
Modifying a virtual server.....	149
Layer 7 – real servers.....	150
Adding / modifying a real server.....	150
Layer 7 - advanced configuration.....	151
SSL termination.....	154
Layer 7.....	154
Layer 4.....	154
Adding / modifying an SSL Virtual server.....	155
SSL - advanced configuration.....	156
Heartbeat configuration.....	157

Floating IPs .....	159
Hostname & DNS.....	160
Network interface configuration.....	161
Routing.....	163
System Date & Time.....	163
Physical – Advanced Configuration.....	164
Setup Wizard.....	164
Upgrade Appliance.....	165
Execute shell Command.....	165
Maintenance.....	166
Backup & Restore.....	166
Restart Services.....	167
Restart Ldirectord.....	167
Restart HAProxy.....	167
Restart Pound.....	167
Restart Heartbeat.....	167
System Control.....	167
Software Update.....	168
Fallback Page.....	168
Firewall Script.....	170
Firewall Lock Down Wizard.....	171
Initialize Graphs.....	172
Users & Passwords.....	172
Reports.....	174
Layer 4 Status.....	174
Layer 7 Status.....	174
Layer 4 Traffic Rate.....	175
Layer 4 traffic Counters.....	175
Layer 4 Current Connections.....	175
Layer 4 Current Connections (resolve hostnames).....	175
Graphing.....	176
Reset Packet Counters.....	177
Logs.....	178
Load balancer.....	178
Layer 4.....	178
Layer 7.....	178
SSL Termination.....	178
Heartbeat.....	178
Section H – Appendix.....	179
Front & Rear Panel Layouts.....	180

## **Section A – Introduction**

## Appliance Details

The Loadbalancer.org appliance is an Intel based server running the GNU/Linux operating system with a custom kernel configured for load balancing. Loadbalancer.org strongly recommends that appliances should always be deployed in a fail-over (clustered pair) configuration for maximum reliability.

The core software is based on customized versions of: Centos 5/ RHEL 5, Linux 2.6, LVS, HA-Linux, HAProxy, Pound & Ldirectord.

## *Version 7.x*

The latest version delivers a completely revamped user interface, several new features as well as many improvements to others. A quick summary is shown below:

- Brand new Web User Interface
- Full IPv6 support
- Improved System Overview with real-time graphs showing key appliance statistics
- New master / slave role status display
- Multiple ports per VIP at layer 4
- SIP Call ID Persistence
- New Technical Support Download option that automatically bundles all logs & data to assist Loadbalancer.org staff
- Enhanced WUI data validation checks
- Code optimized throughout

## Initial Configuration

Each load balancer must initially be individually configured. Once this is done, all configuration takes place on the master load balancer and this is automatically replicated to the slave load balancer. This means that if the master load balancer fails, the traffic will be seamlessly transferred to the slave.

The load balancer can be configured at the console by plugging in a keyboard, mouse & monitor or remotely via the http or secure https web based interface.

## Additional Information

This manual should provide you with enough information to be very productive with your Loadbalancer.org appliance. However, if there are aspects of the appliance that have not been covered, or you have any questions, then please contact [support@loadbalancer.org](mailto:support@loadbalancer.org).



## Deployment Guides

Deployment guides have also been written that focus on specific applications. Links to these are included on the Solutions page of our website : <http://www.loadbalancer.org/solutions.php>

At the time of writing, the following deployment guides are available:

- [Load Balancing IIS Web Servers](#)
- [Load Balancing Web Proxies / Filters](#)
- [Load Balancing OCS 2007 R2](#)
- [Load Balancing Terminal Services](#)
- [Load Balancing Exchange 2010](#)
- [Load Balancing Sharepoint 2010](#)

## **Section B – Load balancing Concepts**

## Load Balancing Algorithms

The Loadbalancer.org appliance supports several different load balancing algorithms. Each one has its advantages and disadvantages and it depends on the specific application which is the most appropriate to use. Usually the default method *Weighted Least Connection* is a good solution which works well in most situations. The following sections summarize each method supported.

### *Weighted Round Robin*

With this method incoming requests are distributed to real servers proportionally to the real servers weight. Servers with higher weights receive new jobs first and get more jobs than servers with lower weights. Servers with equal weights get an equal distribution of new jobs. This method addresses the weakness of the simple round robin method. Weightings are relative, so it makes no difference if real server #1 and #2 have weightings of 50 and 100 respectively or 5 and 10 respectively.

### *Weighted Least Connection*

This method distributes incoming requests based on the number of current connections and also the weighting of each server. Again, weightings are relative, so it makes no difference if real server #1 and #2 have weightings of 50 and 100 respectively or 5 and 10 respectively.

This is the default method for new VIPs.

### *Destination Hashing*

This algorithm assign jobs to servers through looking up a statically assigned hash table by their destination IP addresses.

### *Agent Based*

In addition to the methods above, Loadbalancer.org appliances also support real server agents. This permits the load balancing algorithm to be modified based on the real servers actual running characteristics. For example, a real server could have a run-away process that is consuming excessive CPU resources. Normally the previous algorithms would have no way of knowing this but with the agent installed on the real server, feedback can be provided to the load balancer and the algorithm adjusted accordingly.

## Layer 4 vs Layer 7

A fundamental choice when setting up the load balancer is whether to configure the services at layer 4 or layer 7.

### The basics

At layer 4 the primary protocols used are TCP and UDP. These protocols are not aware of upper level protocols such as FTP, HTTP, DNS, RDP etc. Therefore the load balancer can only make load balancing decisions based on details available at layers 4 and below such as the port numbers and IP addresses. At layer 7, the load balancer effectively has more information to make load balancing related decisions since more information about upper levels protocols is available.

Layer 7 load balancing uses a proxy at the application layer (HAProxy). HTTP requests are terminated on the load balancer, and the proxy generates a new request which is passed to the chosen real server.

### Performance

Due to the increased amount of information at layer 7, performance is not as fast as at layer 4. If raw throughput is a primary concern, then layer 4 is probably the better choice.

### Persistence

Persistence (sticky connections) is the ability to ensure that a specific client connects back to the same server within a specific time limit. It is normally required when the session state is stored locally to the web server as opposed to a database. At Layer 4, Source IP persistence is available. At layer 7, additional methods such as HTTP cookie persistence where the load balancer sets a cookie to identify the same session and RDP cookie persistence which is used to ensure RDP Terminal Server clients are reconnected to existing sessions.

### Real server changes

At Layer 4, either the ARP problem (please refer to section C page 27-36 or section E page 70-79 for more details) has to be solved or the default gateway on the real servers must be set to point at the load balancer. At Layer 7, the connection is fully proxied and therefore the real servers do not need to be changed.

### Transparency

Transparency refers to the ability to see the originating IP address of the client. Connections at Layer 4 are always transparent where as at layer 7 the IP address of the load balancer is recorded as the source address unless additional configuration steps are taken (such as using TPROXY or utilizing the X-Forwarded-For headers, please see page 152 for more details).

## *Our recommendation*

Where possible we recommend that Layer 4 Direct Routing (DR) mode is used. This offers the best possible performance since replies go direct 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.

## **Section C - Quick Start Guide**

*(Also available as a separate download)*

## Loadbalancer.org terminology

<b><u>Acronym</u></b>	<b><u>Terminology</u></b>
<b>Load Balancer</b>	An IP based traffic manager for clusters
<b>VIP</b>	The Virtual IP address that a cluster is contactable on (Virtual Server)
<b>RIP</b>	The Real IP address of a back-end server in the cluster (Real Server)
<b>GW</b>	The Default Gateway for a back-end server in the cluster
<b>WUI</b>	Web User Interface
<b>Floating IP</b>	An IP address shared by the master & slave load balancer when in a high-availability configuration (shared IP)
<b>Layer 4</b>	Part of the seven layer OSI model, descriptive term for a network device that can route packets based on TCP/IP header information
<b>Layer 7</b>	Part of the seven layer OSI model, descriptive term for a network device that can read and write the entire TCP/IP header and payload information at the application layer
<b>DR</b>	Direct Routing is a standard load balancing technique that distributes packets by altering only the destination MAC address of the packet
<b>NAT</b>	Network Address Translation – Standard load balancing technique that changes the destination of packets to and from the VIP (external subnet to internal cluster subnet)
<b>SNAT</b> (HAProxy)	Source Network Address Translation – Load balancer acts as a proxy for all incoming & outgoing traffic
<b>SSL Termination</b> (Pound)	The SSL certificate is installed on the load balancer in order to decrypt HTTPS traffic on behalf of the cluster
<b>MASQUERADE</b>	Descriptive term for standard firewall technique where internal servers are represented as an external public IP address. Sometimes referred to as a combination of SNAT & DNAT rules
<b>One-Arm</b>	The load balancer has one physical network card connected to one subnet
<b>Two-Arm</b>	The load balancer has two network interfaces connected to two subnets - this may be achieved by using two physical network cards or by assigning two addresses to one physical network card
<b>Eth0</b>	Usually the internal interface also known as Gb0
<b>Eth1</b>	Usually the external interface also known as Gb1

### *What is a Virtual IP address?*

Most load balancer vendors use the term virtual IP address (VIP) to describe the address that the cluster is accessed from. It is important to understand that the virtual IP (VIP) refers both to the physical IP address and also to the logical load balancer configuration. Likewise the real IP (RIP) address refers both to the real servers physical IP address and its representation in the logical load balancer configuration.

### *What is a Floating IP address?*

The floating IP address is shared by the master and slave load balancer when in a high-availability configuration. The network knows that the master controls the floating IP address and all traffic will be sent to this address. The logical VIP matches this address and is used to load balance the traffic to the application cluster. If the master has a hardware failure then the slave will take over the floating IP address and seamlessly handle the load balancing for the cluster. In scenarios that only have a master load balancer there can still be a floating IP address, but in this case it would remain active on the master unit only.

## What are Your Objectives?

It is important to have a clear focus on your objectives and the required outcome of the successful implementation of your load balancing solution. If the objective is clear and measurable, you know when you have achieved the goal.

Load balancers have a number of flexible features and benefits for your technical infrastructure and applications. The first question to ask is:

*Are you looking for increased performance, reliability, ease of maintenance or all three?*

<b>Performance</b>	A load balancer can increase performance by allowing you to utilize several commodity servers to handle the workload of one application
<b>Reliability</b>	Running an application on one server gives you a single point of failure. Utilizing a load balancer moves the point of failure to the load balancer. At Loadbalancer.org we advise that you only deploy load balancers as clustered pairs to remove this single point of failure
<b>Maintenance</b>	Using the appliance, you can easily bring servers on and off line to perform maintenance tasks, without disrupting your users



In order to achieve all three objectives of performance, reliability & maintenance in a web based application, your application must handle persistence correctly (see page 41 for more details).

## What Is The Difference Between a One-Arm and a Two-Arm Configuration?

The number of 'arms' is a normally descriptive term for how many physical connections (Ethernet interfaces) are used to connect a device to a network. It is very common for a load balancer that uses a routing method (NAT) to have a two-arm configuration. Proxy based load balancers (SNAT) commonly use a one-arm configuration.

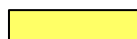
<b>One-Arm</b>	The load balancer has <b>one</b> physical network card connected to <b>one</b> subnet
<b>Two-Arm</b>	The load balancer has <b>two</b> network interfaces connected to <b>two</b> subnets - this may be achieved by using two physical network cards or by assigning two addresses to one physical network card

## What Are The Different Load Balancing Methods Supported?

The Loadbalancer.org appliance is one of the most flexible load balancers on the market. The design of the appliance allows different load balancing modules to utilize the core high availability framework of the appliance. Multiple load balancing methods can be used at the same time or in combination with each other.

Layer 4	DR (Direct Routing)	Ultra-fast local server based load balancing <i>Requires handling the ARP issue on the real servers</i>	1 ARM
Layer 4	NAT (Network Address Translation)	Fast Layer 4 load balancing, the appliance becomes the default gateway for the real servers	2 ARM
Layer 4	TUN	Similar to DR but works across IP encapsulated tunnels	1 ARM
Layer 7	SSL Termination (Pound)	Usually required in order to process cookie persistence in HTTPS streams on the load balancer <i>Processor intensive</i>	1 or 2 ARM
Layer 7	SNAT (Source Network Address Translation: HAProxy)	Layer 7 allows great flexibility including full SNAT and WAN load balancing, cookie insertion and URL switching <i>Not as fast as Layer 4</i>	1 or 2 ARM

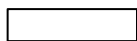
### **Key:**



Recommended for high performance fully transparent and scalable solutions



Recommended if HTTP cookie persistence is required, also used for numerous Microsoft applications such as Terminal Services (RDP cookie persistence) and Exchange, that require SNAT mode



Only required for Direct Routing implementation across routed networks (rarely used)

### **Loadbalancer.org Recommendation:**

Where feasible, one-arm direct routing (DR) mode is our recommended method because it's a very high performance solution with little change to your existing infrastructure.





Sometimes it's not possible to use DR mode. The two most common reasons being: if the application cannot bind to the RIP & VIP at the same time; or if the host operating system cannot be modified to handle the ARP problem (see page 27-36 for more details)

A second option is Network Address Translation (NAT) mode. This is a fairly high performance solution but it requires the implementation of a two-arm infrastructure with an internal and external subnet to carry out the translation (the same way a firewall works). Network engineers with experience of hardware load balancers will have often used this method.

The third option is Source Network Address Translation (SNAT) mode using HAproxy. If your application requires that the load balancer handles cookie insertion, RDP cookies, Session Broker integration or SSL termination then this option is appropriate. This can be deployed in one-arm or two-arm mode and does not require any changes to the application servers. HAproxy is a high-performance solution that operates as a full proxy, but due to this it cannot perform as fast as the layer 4 solutions.

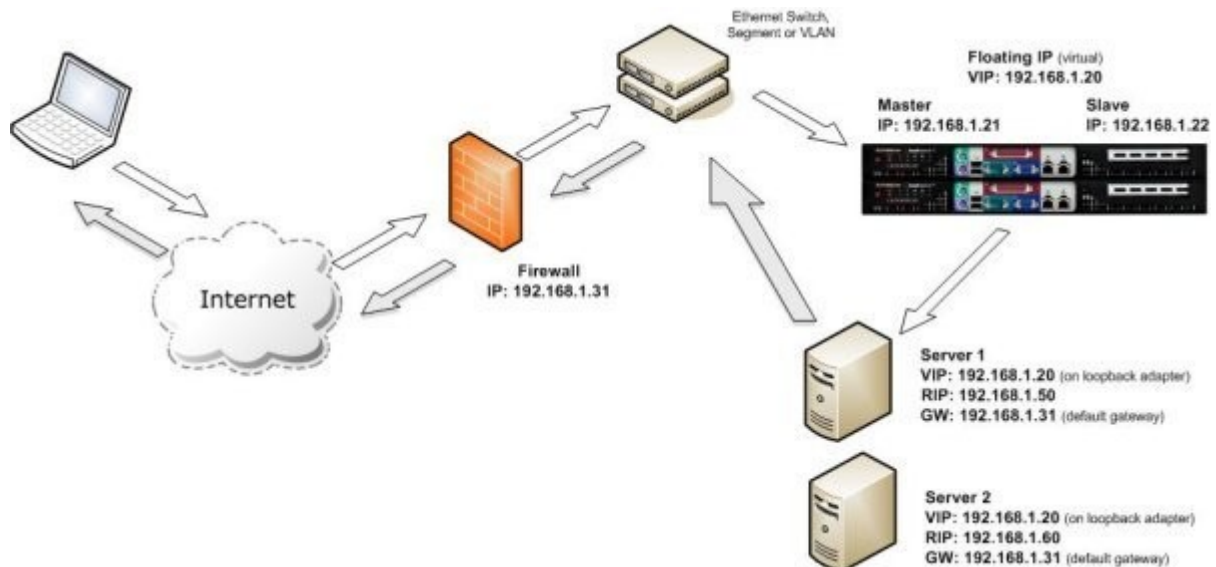


If your application doesn't maintain its own state information then you may need to use cookie insertion to maintain server persistence (affinity)

The following sections describe these configurations in more details.

## Direct Routing (DR)

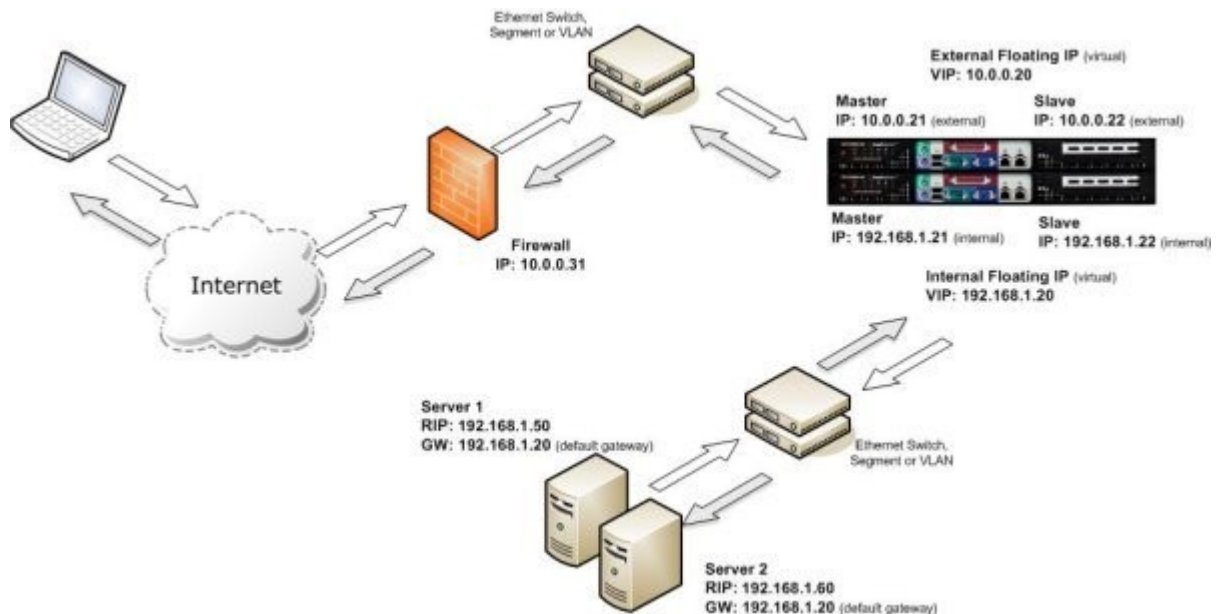
The one-arm direct routing (DR) mode is the recommended mode because it's a very high performance solution with little change to your existing infrastructure. *NB. Foundry networks call this Direct Server Return and F5 call it N-Path.*



- Direct routing works by changing the destination MAC address of the incoming packet on the fly which is very fast
- However, this means that when the packet reaches the real server it expects it to own the VIP. This means you need to make sure the real server responds to both its own IP and the VIP, but does not respond to ARP requests for the VIP. Please refer to page 27-36 for more details on resolving the ARP problem
- On average, DR mode is 8 times quicker than NAT for HTTP, 50 times quicker for Terminal Services and much, much faster for streaming media or FTP
- Load balanced services can be configured directly on the interface (normally eth0) with no additional IP address. However, when using a clustered pair, all load balanced virtual services **must** be configured on a floating IP to enable failover & failback between master & slave
- The virtual server and real servers must be in the same switch fabric / logical network. They can be on different subnets, provided there are no router hops between them. If multiple subnets are used, an IP address in each subnet must be defined on the load balancer
- Port translation is not possible in DR mode i.e. having a different RIP port than the VIP port
- DR mode is transparent, i.e. the real server will see the source IP address of the client
- Administration of the load balancer is via any active IP address (on port 9080)

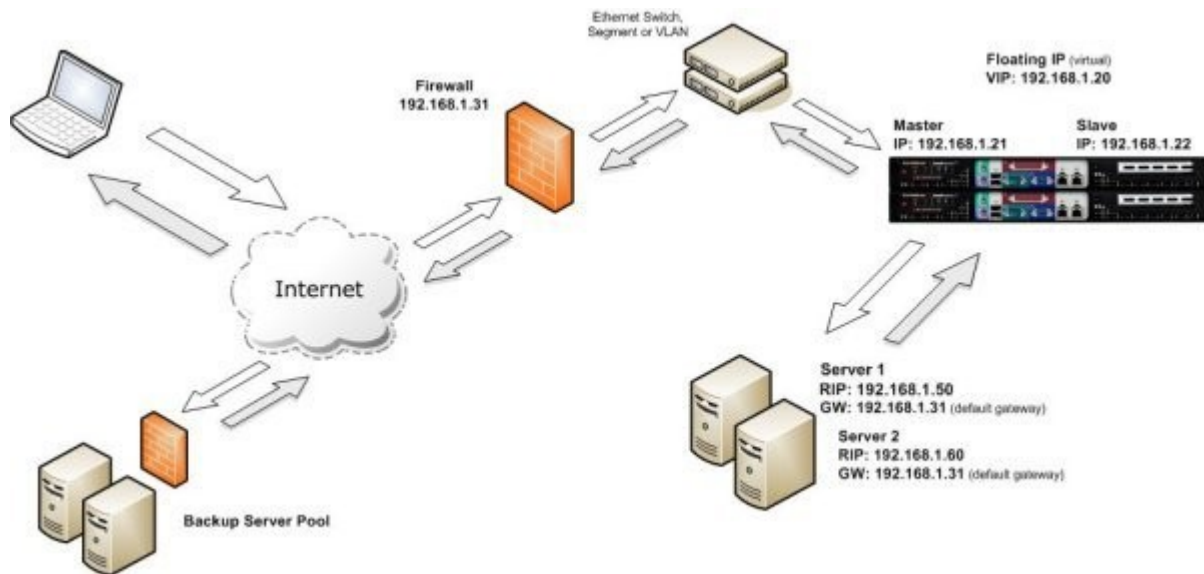
## Network Address Translation (NAT)

Sometimes it is not possible to use DR mode. The two most common reasons being: if the application cannot bind to RIP & VIP at the same time; or if the host operating system cannot be modified to handle the ARP problem. The second choice is Network Address Translation (NAT) mode. This is also a fairly high performance solution but it requires the implementation of a two arm infrastructure with an internal and external subnet to carry out the translation (the same way a firewall works).



- In two-arm NAT mode the load balancer translates all requests from the external virtual server to the internal real servers
- It is a good idea to use *eth1* as your external network and *eth0* as internal, otherwise you will need to change the *autonat* setting in *Edit Configuration > Layer 4 – Advanced Configuration*
- The real servers must have their default gateway configured to point at the load balancer. When master & slave units are used, a floating IP must be used to enable failover
- Real servers are automatically given access to the Internet through the load balancer (via *autonat*)
- Load balanced services can be configured directly on the interface (normally *eth0*) with no additional IP address. However, when using a clustered pair all load balanced virtual services must be configured on a floating IP to enable failover & failback between master & slave
- Normally the virtual server and real servers should be located on different subnets within the same logical network (i.e. no router hops) and the load balancer should have an IP address in each subnet.  
*Note-1: It is possible to have real and virtual servers in the same subnet – please refer to the Advanced NAT topic in Section F. Note-2: It is possible to have the real servers located on routed subnets, but this would require a customized routing configuration on the real servers and is not recommended*
- If you want real servers to be accessible on their own IP address for non-load balanced services, e.g. SMTP or RDP, you will need to setup individual SNAT and DNAT firewall script rules for each real server. Please refer to the Advanced NAT Considerations section for more details
- You can also configure the load balancers in one-arm NAT mode, but in order to make the servers accessible from the local network you need to change the routing configuration on the real servers. Please refer to the Advanced NAT Considerations section for more details.
- NAT mode is transparent , i.e. the real server will see the source IP address of the client
- Administration of the load balancer is via any active IP address (on port 9080)

## Source Network Address Translation (SNAT)



If your application requires that the load balancer handles cookie insertion then you need to use the SNAT configuration. This also has the advantage of a one-arm configuration and does not require any changes to the application servers. However, as the load balancer is acting as a full proxy it doesn't have the same raw throughput as the routing based methods.

The network diagram for the Layer 7 HAProxy SNAT mode is very similar to the Direct Routing example except that no re-configuration of the real servers is required. The load balancer proxies the application traffic to the servers so that the source of all traffic becomes the load balancer.

- As with other modes a single unit does not require a Floating IP, although it is recommended to make adding a slave unit easier
- SNAT is a full proxy and therefore load balanced real servers do not need to be changed in any way
- Because SNAT is a full proxy any server in the cluster can be on any accessible subnet including across the Internet or WAN
- SNAT is not transparent by default, i.e. the real servers will not see the source IP address of the client, they will see the load balancers IP address. If required, this can be solved by either enabling TPROXY on the loadbalancer, or leveraging the X-forwarded-For header.



For detailed configuration examples, please refer to section D starting on page 43

## High-Availability Configuration of Two Loadbalancer.org Appliances

Loadbalancer.org's recommended configuration is to use a clustered pair of load balancers to provide a highly available and resilient load balancing solution. In this configuration, the pair communicates via a heartbeat to determine if the master node is active. Should the master node suffer a failure, the slave will immediately take over any resources hosted on the shared floating IP addresses.



Using a single load balancer introduces a single point of failure for your infrastructure so it is strongly recommended to use two appliances in a clustered pair

### *Clustered Pair Configuration Methods*

There are two ways to configure a clustered pair; either by using the wizard or configuring the units manually.

#### Using the Wizard

If the wizard is used, the slave is configured first and then the master. This ensures that both units can first communicate via the selected link (via a serial cable or over the network) , and also that settings that are setup on the master and correctly replicated to the slave.



For more details on using the wizard and an example, refer to page 23

#### Manual configuration

If the master is configured first without using the wizard and the slave is added later, the following points should be considered:

- The IP address for the slave must be configured in the master using *Edit Configuration > Hostname DNS* in the WUI
- The **Force full slave sync** option in *Edit Configuration > Hostname DNS* should be checked prior to clicking **Update** - this will ensure that all configured services are correctly replicated over to the slave unit
- Once the IP address is set and synchronization has occurred, its important to restart heartbeat on both units to ensure heartbeat starts cleanly. This can be done via *Maintenance > Restart Services* in the WUI



For more details refer to configuration Example-2 on page 48, for more details on adding a slave and verifying failover refer to page 127 and for more details on setting up heartbeat, refer to page 99

## Unpacking and setting up the Loadbalancer.org appliance

- Remove all packaging
- Rack mount the appliance if required
- The power supply is an auto sensing unit (100v to 240v)
- Connect the power lead from the power socket to the mains or UPS
- Connect your network cable from your switch or hub to the internal network port (*eth0*)
- If using a two-armed configuration connect a second network cable to the external port (*eth1*)

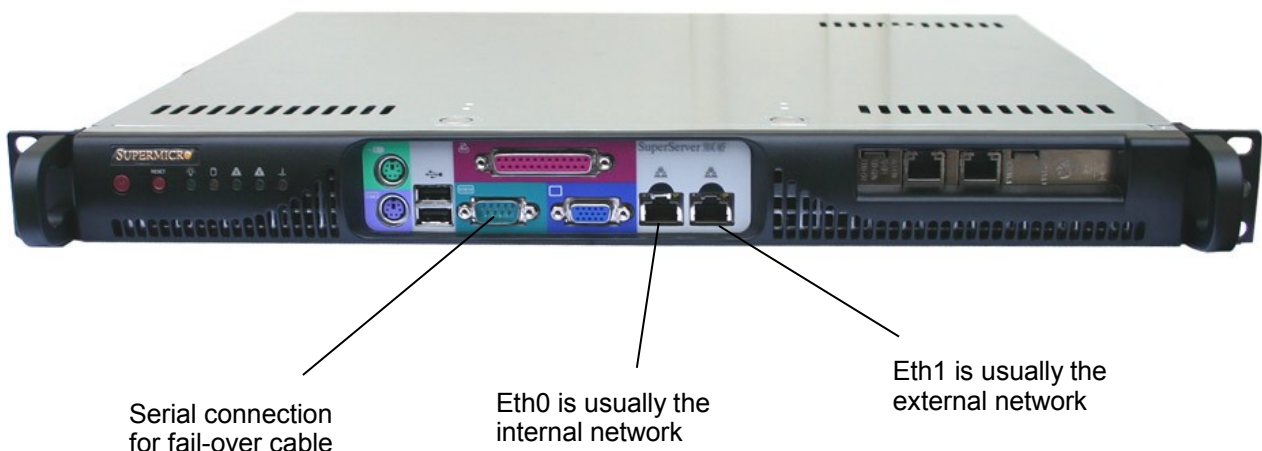


If two load balancers (recommended) are being used, connect a null modem cable (one cable is supplied with each appliance) between the two serial ports, then configure the slave first

- Attach a monitor to the VGA port
- Attach a keyboard to the USB or PS/2 port
- Check mains power is on
- Press the power switch to start the appliance (fans should start & front panel LEDs should light)
- Allow a minute for booting

The next few pages of this document detail the following steps:

- Configuring the load balancer using the web based wizard
- Additional appliance configuration using the web interface
- Testing the load balancer configuration



## Configuring The Loadbalancer.org Appliance Using The Web Based Wizard

This section deals with the process of configuring a single load balancer appliance via the web based wizard. The web based wizard enables you to configure a complete working configuration with one virtual server and one real server. You can then continue in the web interface to make modifications to this basic configuration, add additional Virtual IP's (VIPs), additional Real Servers (RIPs) etc.

### *Network interface configuration*

log in to the console:      **Username:** root  
                                 **Password:** loadbalancer

You can access the web interface either via Links at the console or from a web browser on a client connected to the same network (**recommended**). By default the IP address for eth0 on the physical appliance is set to 192.168.2.21/24. If another device already has this IP address then no address will be assigned. If you want to change or assign an IP address, the following command should be used once logged in as root:

```
ip addr add <IP address>/<mask> dev eth0
```

e.g.

```
ip addr add 192.168.1.100/24 dev eth0
```

**NB. This is temporary, the IP address MUST be set via the WUI to make this permanent**

### *Accessing the Web User Interface (WUI)*

With a web browser, access access the WUI : **http://192.168.2.21:9080/lbadmin/**  
(replace 192.168.2.21 with the correct address if this has been changed)

log in to the WUI:      **Username:** loadbalancer  
                                 **Password:** loadbalancer

*NOTE: If you prefer you can use the HTTPS administration address: **https://192.168.2.21:9443/lbadmin/***

This will take you to the Loadbalancer.org web interface, where the web based configuration wizard will start by default the first time it is accessed. This wizard will ask a series of questions in order to configure the appliance with a basic configuration.

#### **EDIT CONFIGURATION > SETUP WIZARD**

The Loadbalancer.org Setup Wizard has not been run yet. You can run it now or anytime later with  
Edit Configuration > Setup Wizard

Do you want to run it now?

☐ yes ☐ no



## Example answers using the wizard for a two-arm NAT configuration (single unit)

Once you have decided on your load balancing configuration, completing the wizard should be fairly self explanatory. The following example is for a two-arm NAT configuration:

### EDIT CONFIGURATION > SETUP WIZARD

---

Is this unit part of an HA-pair? ☐ yes ☒ no

---

Will the load balancer form part of a one armed set-up (i.e. same subnet as servers)? ☐ yes ☒ no

---

Then the load balancer will form part of a two-armed set-up. (See Quickstart guide for further explanation.)

We will now configure the load balancer's network interfaces:

Enter the IP address for the INTERNAL interface eth0 (CIDR format):

Enter the IP address for the EXTERNAL interface eth1 (CIDR format):

Now we will configure the DNS and gateway settings for the load balancer.

Enter the IP address of the default gateway IP v4:

Enter the IP address of the default gateway IP v6:

Enter the IP address of the nameserver:

Enter the IP address of the second nameserver:

Now we will configure the first Virtual Service.

Enter the port number for the Virtual Service:

Enter the IP address of the first Real Server (backend):

Please check that all your settings are correct!

Check that your settings are correct and click **Submit**. Once the wizard is complete the load balancer is configured and ready to use.

For NAT mode, you also need to configure the real server to ensure that it uses the internal IP address of the load balancer as its default gateway. Once this is done you can test the virtual server from the external network. By default, the wizard uses the IP address of the external interface for the first virtual server, 10.0.0.120 in this example.

You can now use the *Edit Configuration* menu in the WUI to easily add more virtual or real servers to your configuration.



To restore manufacturer's settings – at the console use the command **lbrestore** or in the WUI goto *Maintenance > Backup & Restore > Restore Manufacturer's Defaults*. This will set the address to 192.168.2.21 if this address is available.



## Additional Appliance Configuration Using The Web Interface



When using a Clustered Pair, all configuration must be done via the master unit, the slave unit will then be synchronized automatically. If for some reason the master is down and changes are required to the slave's setup, these changes will not be automatically replicated to the master when its brought back online and therefore must be handled manually.

This section deals with the configuration of the load balancers via the web interface. The wizard will enable you to get up and running very quickly with a virtual server and a single configured real (back-end) server . You can use the web interface to add or modify existing virtual and real servers as required.

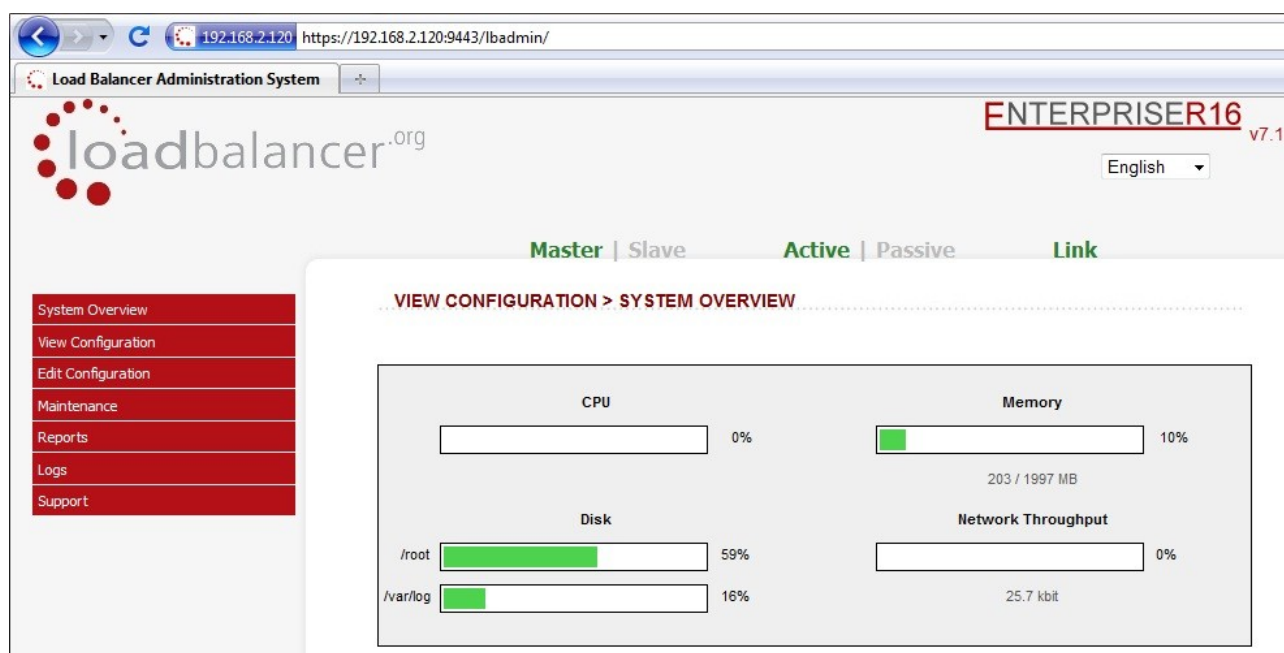
If you have already used the web based wizard, then you will already be using the WUI. From here all administration tasks can be carried out. If not, access the WUI as follows:

With a web browser access the web interface: ***http://192.168.2.21:9080/lbadmin/***

*(replace 192.168.2.21 with the correct address)*

log in to the WUI: ***Username:*** loadbalancer  
***Password:*** loadbalancer

***NOTE:*** If you prefer you can use the HTTPS administration address: ***https://192.168.2.21:9443/lbadmin/***



All administration tasks can be carried out through the web interface.

## Adding additional real servers

The wizard sets up one virtual server with one real server (back-end server) to send the traffic to. You will need to add any extra servers through the Web User Interface:



- Use *Edit Configuration > Layer 4 Configuration > Real Servers* , you'll see the first Real Server that was created by the wizard

### EDIT CONFIGURATION > REAL SERVERS

VIP1	10.0.0.120	Ports 80	NAT	[ Add a new Real Server ]	
RIP1	192.168.2.60	Port 80	Weight 1	[ Modify ]	[ Delete ]

- Click [Add a new Real Server]

### EDIT CONFIGURATION > ADD A NEW REAL SERVER

Label	<input type="text" value="RIP2"/>	
Real Server IP Address	<input type="text" value="192.168.2.70"/>	
Real Server Port	<input type="text" value="80"/>	
Weight	<input type="text" value="1"/>	
Minimum Connections	<input type="text" value="0"/>	
Maximum Connections	<input type="text" value="0"/>	
<input type="button" value="Update"/>		

- Enter the label, IP address and port number of your additional real server
- The weight defaults to 1 making the real server active immediately and equal weight to the first real server added by the wizard. If the real servers have different performance specifications, then the weight can be adjusted – a higher number means more traffic is sent to that server
- Leave the minimum & maximum connections as 0 for unrestricted

## Configuring the Real Servers

Depending on the deployment method (DR, NAT or SNAT) used, the actual physical servers may need to be configured to allow the load balancer to operate correctly. The following sections define what is needed for the various modes.

### Configuring the real servers for NAT mode

If you are using a two-arm NAT load balancing method, the real server configuration is a simple case of configuring the load balancer as the default gateway. Normally, a floating IP address is added using *Edit Configuration > Floating IPs*. This is important when a master / slave configuration is used to allow failover & fallback of the default gateway address.



Failure to correctly configure the real servers default gateway is the most common mistake when using NAT mode

### Configuring the real servers for DR mode (Linux)

If you are using the one-arm DR load balancing method, each real server requires the ARP problem to be solved. All real servers must be configured to respond to the VIP address as well as the RIP address. This is because in DR mode load balanced traffic arrives on the VIP address, whilst other traffic such as health-checks, administration traffic etc. use the real server's IP address.

#### Detecting the ARP problem

You can use *Reports > Layer 4 Current Connections* to check whether the ARP problem has been solved. If not, the connection state will be SYN\_RECV as shown below when a client connection to the VIP is attempted:

#### REPORTS > LAYER 4 CURRENT CONNECTIONS

```
IPVS connection entries
pro expire state      source          virtual         destination
TCP 00:51 SYN_RECV    192.168.2.7:64763 192.168.2.109:80 192.168.2.99:80
```

#### Solving for Linux – method 1 (using iptables)

You can use iptables (netfilter) on each real server to re-direct incoming packets destined for the virtual server IP address. To make this permanent, simply add the command to an appropriate start-up script such as /etc/rc.local. If the real server is serving multiple VIPs, add additional iptables rules for each VIP.

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

e.g.

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

this means redirect any incoming packets destined for 10.0.0.21 (the virtual server) locally.



Method 1 does not work with IPv6 Virtual Servers, use method 2 below instead

### **Solving for Linux – method 2 (using arp\_ignore sysctl values)**

This is the preferred method as it supports both IPv4 and IPv6. Each real server needs the loopback adapter to be configured with the Virtual Servers IP address. This address must not respond to ARP requests and the web server also needs to be configured to respond to this address. To set this up follow steps 1-3 below.

#### **Step 1 : re-configure ARP on the real servers (this step can be skipped for IPv6 virtual servers)**

To do this add the following lines to /etc/sysctl.conf:

```
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
```

#### **Step 2 : apply these settings**

Either reboot the real server or run the following command to apply these settings:

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

#### **Step 3 : add the virtual servers' IP address to the loopback adapter**

run the following command for each VIP. To make this permanent, simply add the command to an appropriate startup script such as /etc/rc.local.

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

*for IPv6 addresses use:*

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

Alternatively, modify the appropriate interface script to add the additional IP address(es).



Failure to correctly configure the real servers to handle the ARP problem is the most common mistake in DR mode configurations

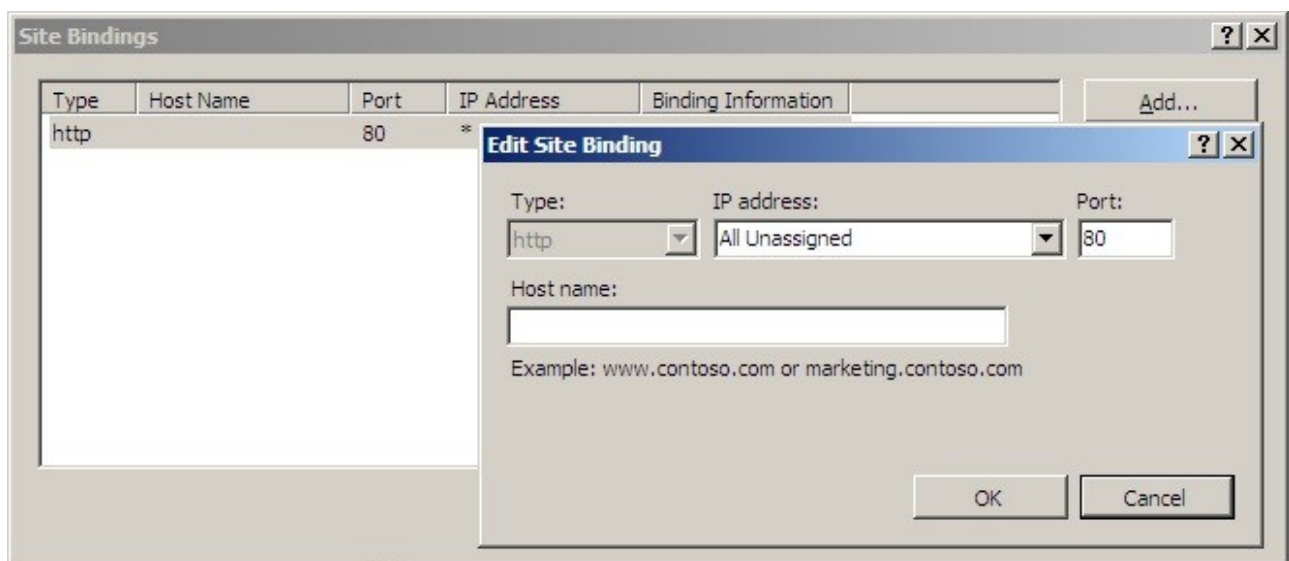
## Configuring the real servers for DR mode (Windows)

If you are using a one-arm DR load balancing method, each web server requires the ARP problem to be handled:

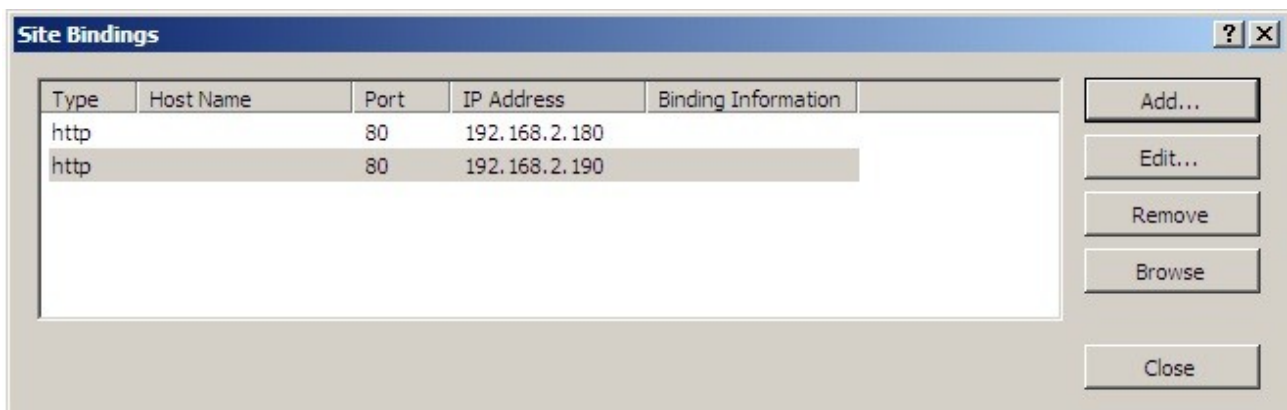
- for all real servers in Direct Routing mode the load balanced application must respond to both the virtual IP as well as the servers real IP. With Windows IIS the IP address must either be set to (All Unassigned) or use the Advanced tab to add a second IP address as shown below
- Each real server must have the Microsoft loopback adapter installed and configured
- The Microsoft loopback adapter must be configured to deal with the ARP problem

### Configuring IIS to respond to both the RIP and VIP

By default, IIS listens on all configured IP addresses, this is shown in the example below. As can be seen the IP address field is set to "All Unassigned".



If the default configuration is left, no further IIS configuration is required. If you do change the IP address in the bindings from "All Unassigned" to a specific IP address, then you need to make sure that you also add a binding for the Virtual Server IP address (VIP) as shown below:

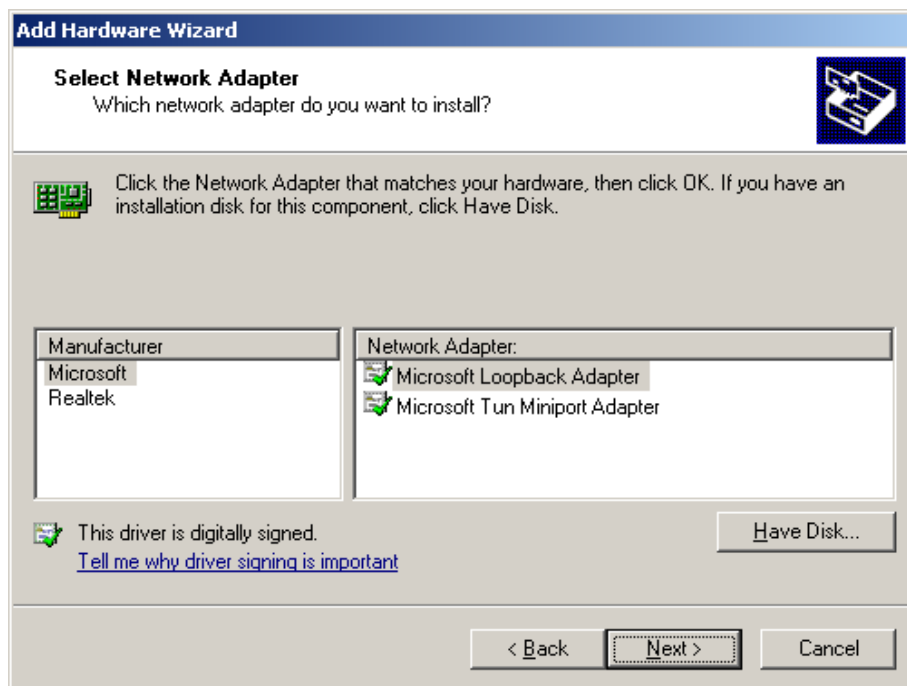


## **Resolving ARP issues for Windows server 2000 / 2003 (DR mode only)**

Windows server supports the direct routing (DR) method through the use of the MS loopback adapter to handle the traffic. The IP address on the loopback adapter must be set to be the same as the Virtual Servers IP address (VIP). If the real server is included in multiple VIPs, you can add additional IP addresses to the loopback adapter that correspond to each VIP.

### **Installing the Microsoft loopback adapter**

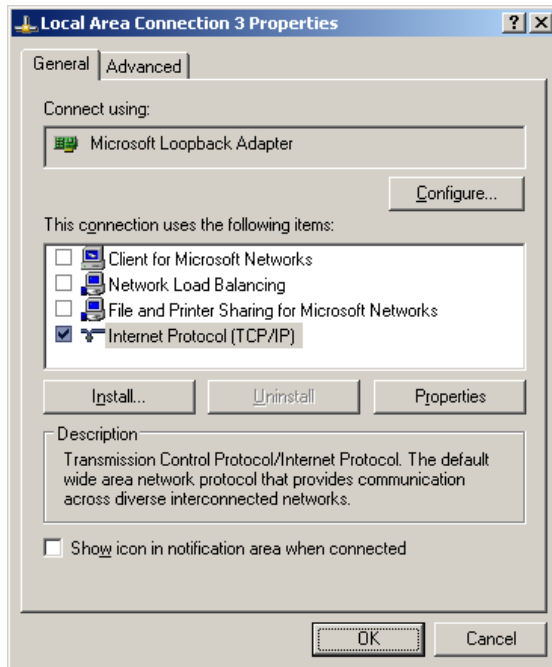
1. Open the Control Panel and double-click Add Hardware
2. Once the Hardware Wizard opens, click Next
3. Select 'Yes, I have already connected the hardware', click Next
4. Scroll to the bottom of the list, select 'Add a new hardware device' and click Next
5. Select 'Install the hardware that I manually select from a list (Advanced)', click Next
6. Select 'Network adapters', click Next
7. Select 'Microsoft' & 'Microsoft Loopback Adapter', click Next



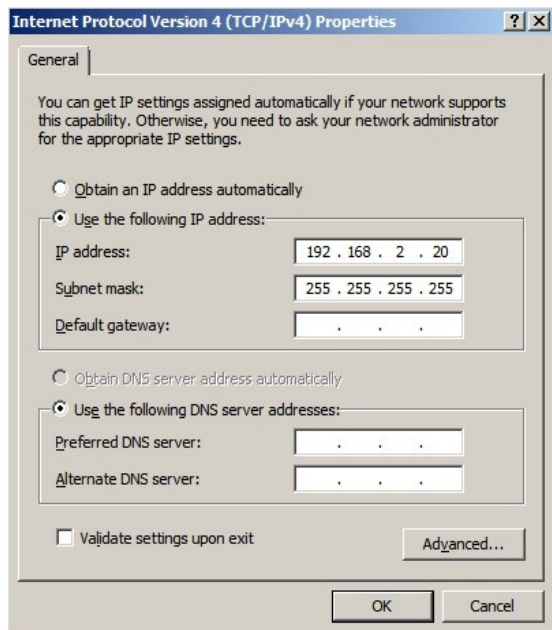
8. Click Next to start the installation, when complete click Finish

## Configuring the loopback adapter

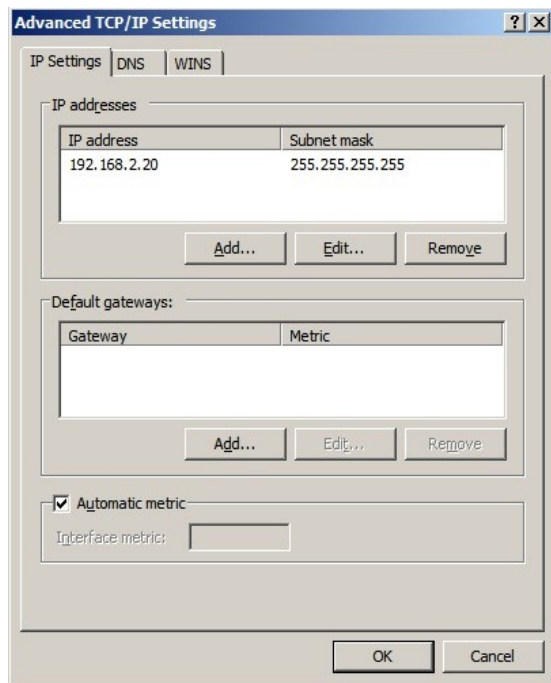
1. Open the Control Panel and double-click Network Connections
2. Right click the new loopback adapter and select properties



3. Un-check all items except Internet Protocol (TCP/IP)
4. Select Internet Protocol (TCP/IP), click Properties and configure the IP address to be the same as the Virtual Server (VIP), with a full subnet mask e.g. 192.168.2.20/32



5. Click on the *Advanced* button and change the Interface Metric to 254 (This stops the adapter responding to ARP requests).



6. Click OK on the Advanced and TCP/IP popup windows, then click Close on the Local Area Connection window to save the new settings
7. Now repeat the above process for all other Windows 2000 / 2003 real servers



For Windows server 2003 SP1 & above, if you have enabled the built-in firewall, you will need to create an exception to enable access to the web server. This exception by default will allow traffic on both the network and loopback adapters

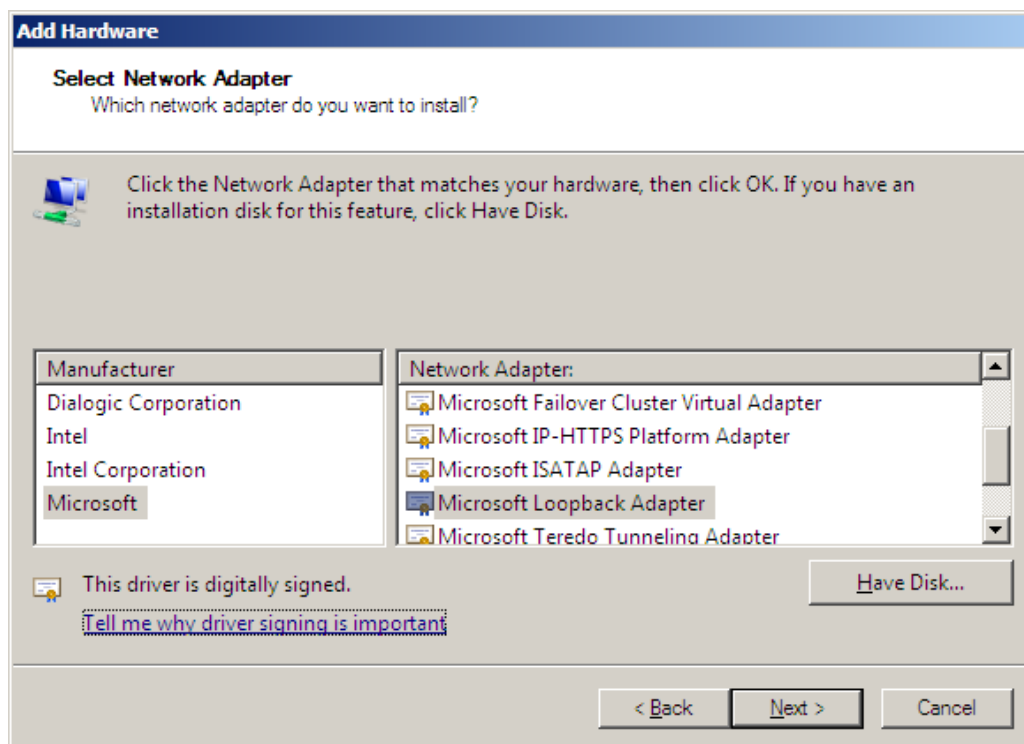


## **Resolving ARP issues for Windows server 2008 (DR mode only)**

The basic concept is the same as for Windows 2000 / 2003. However, additional steps are required to set the strong / weak host behavior. This is used to either block or allow interfaces receiving packets destined for a different interface on the same server.

### Installing the Microsoft loopback adapter

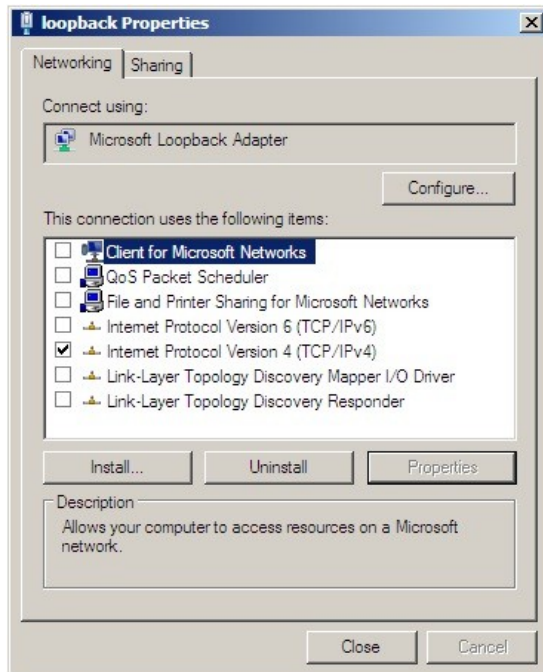
1. Click Start, select Run and enter **hdwwiz** to start the Hardware Installation Wizard
2. When the Wizard has started, click Next
3. Select 'Install the hardware that I manually select from a list (Advanced)', click Next
4. Select 'Network adapters', click Next
5. Select 'Microsoft' & 'Microsoft Loopback Adapter', click Next



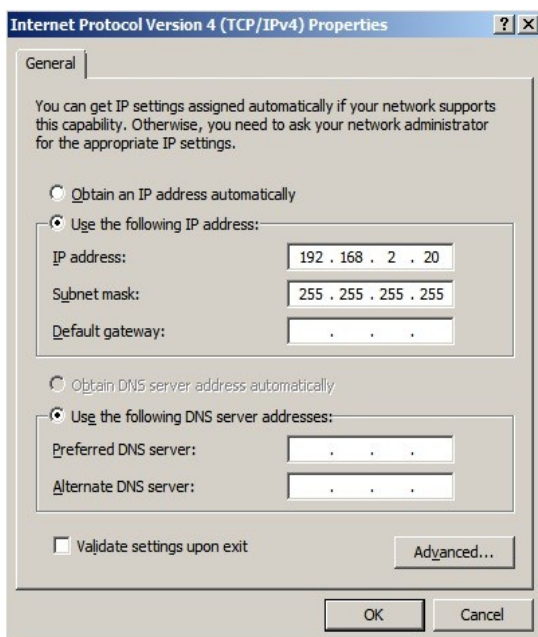
6. Click Next to start the installation, when complete click Finish

### Configuring the loopback adapter

1. Open Control Panel and double-click Network and Sharing Center
2. Click Change adapter settings
3. Right-click the new loopback adapter and select Properties



4. Un-check all items except Internet Protocol Version 4 (TCP/IPv4)
5. Select Internet Protocol Version (TCP/IPv4), click Properties and configure the IP address to be the same as the Virtual Server (VIP), with a full subnet mask e.g. 192.168.2.20/32



6. Click OK on the TCP/IP popup window, then click Close on the Local Area Connection window to save the new settings

7. For Windows 2008, its not necessary to modify the interface metric on the advanced tab and should be left set to Automatic
8. Now repeat the above process for all other Windows 2008 real servers

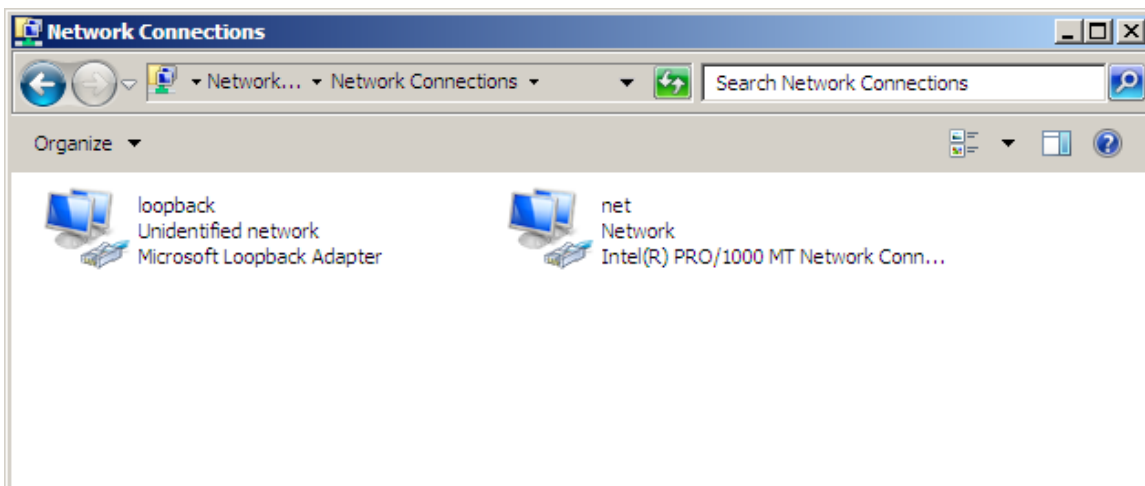
#### Configuring strong /weak host behavior

Windows XP and Windows Server 2003 use the weak host model for sending and receiving for all IPv4 interfaces and the strong host model for sending and receiving for all IPv6 interfaces. You cannot configure this behavior. The Next Generation TCP/IP stack in Windows Vista and Windows Server 2008 supports strong host sends and receives for both IPv4 and IPv6 by default.

To ensure that the Windows 2008 is running in the correct mode to respond to the VIP, the following commands must be run in a command window on the real server :

```
netsh interface ipv4 set interface "net" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostsend=enabled
```

For these commands to work, the LAN connection NIC must be named “net” and the loopback NIC must be named “loopback”. If you prefer to leave your current NIC names, then the commands above must be modified accordingly.



*N.B. The names for the NICs are case sensitive, so make sure that the name used for the interface and the name used in the commands match exactly.*

If you prefer to use the index number for the interface, you can look up the index number using the following command:

```
netsh interface ipv4 show interface
```

then substitute the relevant index number for “net” and “loopback” in the three netsh commands

## Verifying netsh Settings

To verify that settings have been configured correctly, run the following command on each real server to clearly list the settings that have been applied to the interface:

```
netsh interface ipv4 show interface <interface name>
```

i.e. for the 'loopback' adapter run : netsh interface ipv4 show interface loopback

i.e. for the 'net' adapter run : netsh interface ipv4 show interface net

e.g.

```
C:\Users\Administrator>netsh interface ipv4 show interface loopback
```

Interface loopback Parameters

```
-----
IfLuid                      : ethernet_9
IfIndex                     : 15
State                       : connected
Metric                     : 30
Link MTU                   : 1500 bytes
Reachable Time              : 28500 ms
Base Reachable Time         : 30000 ms
Retransmission Interval    : 1000 ms
DAD Transmits               : 3
Site Prefix Length         : 64
Site Id                    : 1
Forwarding                  : disabled
Advertising                 : disabled
Neighbor Discovery          : enabled
Neighbor Unreachability Detection : enabled
Router Discovery            : dhcp
Managed Address Configuration : enabled
Other Stateful Configuration : enabled
Weak Host Sends              : enabled
Weak Host Receives          : enabled
Use Automatic Metric        : enabled
Ignore Default Routes       : disabled
Advertised Router Lifetime  : 1800 seconds
Advertise Default Route     : disabled
Current Hop Limit           : 0
Force ARPND Wake up patterns : disabled
Directed MAC Wake up patterns : disabled
```

```
C:\Users\Administrator>
```

This shows that the settings have been applied correctly.



For Windows server 2008, if you want to leave the built-in firewall enabled, you'll either need to enable the relevant default firewall exceptions or create your own to enable access to the web server. By default these exceptions will allow traffic on both the network and loopback adapters



Failure to correctly configure the real servers to handle the ARP problem is the most common mistake in DR configurations

## Configuring the real server for SNAT mode

When using Layer7 (HAproxy) Virtual Servers, no changes are required to the real servers.

## IPv6 Support

New to v7.x is full IPv6 support. This allows Virtual Servers to be configured using IPv6 addresses. Its also possible to mix IPv4 and IPv6 addresses on a single appliance as illustrated below:

### EDIT CONFIGURATION > NETWORK INTERFACE CONFIGURATION

Bonding	
Bond eth0 & eth1 as bond0:	<input type="checkbox"/> ?
Bond eth2 & eth3 as bond1:	<input type="checkbox"/> ?
<b>Bond Interfaces</b>	
VLAN	
Interface:	eth0 ? <b>Add VLAN</b>
VLAN ID:	1 ?
IP Address Assignment	
eth0	192.168.2.135/24 fde6:d14c:3089:1::382/120
eth1	10.12.1.135/24 fde6:d14c:3089:1::384/120
eth2	
eth3	
<b>Configure Interfaces</b>	

## Testing The Load Balancer Configuration

For testing, add a page to each real web servers root directory e.g. test.html and put the server name on this page for easy identification during your tests.

Now you need a couple of clients to do the testing. Open up a web browser on two different clients and enter the URL for the VIP i.e. <http://192.168.1.20/>.

Each client should see a different server name because of the load balancing algorithm in use i.e. they are being load balanced across the cluster.

**Why test using two clients?** If you use a single client it will most likely keep on hitting the same server for multiple requests. This is to do with the way that the load balancing algorithms are optimized.



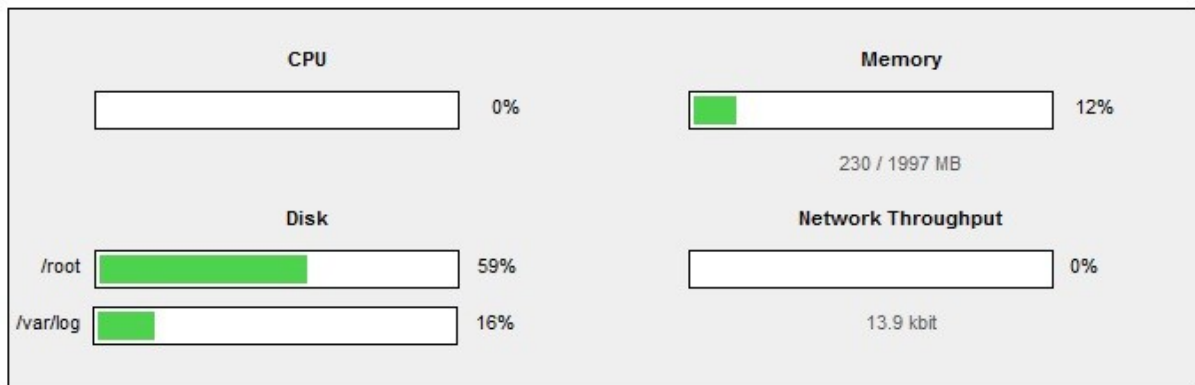
When using a two-arm NAT load balancing method the test client must be in the external subnet

## Connection error diagnosis

If you get a connection error when trying to access the VIP then:

1. Check *View Configuration > Network Configuration* and make sure that the VIP is active on the load balancer, if not check *Logs > Heartbeat* for errors
2. Check *System Overview* and make sure none of your VIPs are highlighted in red. If they are, your cluster is down. Green indicates a healthy cluster, yellow indicates that your cluster may need attention (one real server may be down), and blue indicates a real server has been deliberately taken offline

### VIEW CONFIGURATION > SYSTEM OVERVIEW



Key cluster healthy cluster may need attention cluster is down real server deliberately offline

+	HTTP_Cluster - 192.168.2.182 Ports 80 Protocol TCP	Connections - Active: 0 Inactive: 0
+	FTP_Cluster - 192.168.2.184 Ports 21 Protocol TCP	Connections - Active: 0 Inactive: 0
+	SMTP_Cluster - 192.168.2.186 Ports 25 Protocol TCP	Connections - Active: 0 Inactive: 0

3. If the VIP is still not working then check *Reports > Current Connections* to see the current traffic in detail, any packets marked SYN\_RECV imply incorrect real server configuration. Check that any NAT mode servers have the correct default gateway and any DR mode servers are responding to the VIP as well as their own IP.

## Health check diagnosis

Go to the Maintenance > System Overview section of the web interface and check that when you use 'take offline' the connections are redirected to the rest of the cluster as expected.

Pull the network cable out of one of the web servers, wait a few seconds (for the load balancer to detect the change) and then refresh the browsers on both clients. They should now both switch to the same server (as one has been removed from the load balancing list).

Put the network cable back in to the web server, wait a few seconds and then refresh the browsers again. They should now show different web servers again.

Key cluster healthy cluster may need attention cluster is down real server deliberately offline

HTTP_Cluster - 192.168.2.182 Ports 80 Protocol TCP Connections - Active: 0 Inactive: 0							
Label	IP	Method	Weight	Active conns	Inactive conns		
alpha_server	192.168.2.178	DR	1	0	0	Drain Halt	↑
bravo_server	192.168.2.190	DR	0	0	0	Bring Online	⚙
charlie_server	192.168.2.191	DR	0	0	0	Drain Halt	↓

'**alpha\_server**' is green which indicates that the server is operating normally.

'**bravo\_server**' is blue, this indicates that it is deliberately in maintenance mode. You can use 'Bring Online' to make it active.

'**charlie\_server**' is down (red). This implies that the real server has failed a health check; you can investigate this using *Logs > Layer 4*. If you know the real server should be active, you may need to increase the health check time-out *Edit Configuration > Layer 4 – Advanced Settings* or *Layer 7 – Advanced Settings*.



## Testing high-availability for a Loadbalancer.org Clustered Pair

To test fail-over of a clustered pair, once fully configured power down the master and check that the slave unit takes over all the floating IP(s). If fail-over to the slave unit does not occur correctly, check *Logs > Heartbeat* on both nodes for any errors.



It is very important to verify that master / slave failover occurs correctly before going live. This proves the resilience of the cluster and makes you aware of the failover / failback process. Please refer to page 116 for details of the `hb_takeover` command which can be used to force a failover and refer to page 128 for detailed steps on verifying failover / failback



When testing load balancer fail-over, don't just pull the serial cable and network cable out. This will not cause a fail-over but will cause a split brain (I.e. both units active) to occur. You can configure fail-over on network failure but this is not enabled by default. To enable this, a ping node must be configured under *Edit Configuration > Heartbeat Configuration*

New to v7.x is the role status at the top of each screen. For a working pair, the normal view is shown below:



This shows that the master unit is active and that the heartbeat link is up between master & slave.

Other states:

Master   Slave	Active   Passive	Link	this is a master unit, it's active, no slave unit has been defined
Master   Slave	Active   Passive	Link	this is a master unit, it's active, a slave has been defined but the link to the slave is down. <b>Action:</b> <i>check &amp; verify the heartbeat configuration</i>
Master   Slave	Active   Passive	Link	this is a slave unit, it's active (a failover from the master has occurred) and the heartbeat link to the master has been established
Master   Slave	Active   Passive	Link	this is a master unit, a slave unit has been defined, but the link is down (e.g. serial cable unplugged) so the state cannot be determined. In this case the floating IP's may be active on both units. <b>Action:</b> <i>check &amp; verify the heartbeat configuration, check the serial cable (if applicable), check heartbeat logs &amp; if required restart heartbeat on both units</i>
Master   Slave	Active   Passive	Link	this is the master unit, a slave unit has been defined on the master, but the link is down (e.g. serial cable unplugged) so the state cannot be determined. In this case the heartbeat service has probably stopped on both units. <b>Action:</b> <i>check &amp; verify the heartbeat configuration, check the serial cable (if applicable), check heartbeat logs &amp; if required restart heartbeat on both units</i>

**NOTE:** *Restarting heartbeat will cause a temporary outage of all load balanced services*



## Does Your Application Cluster Correctly Handle Its Own State?



Load balancers work most effectively if the application servers are completely stateless. This means that if a web server fails and is automatically taken out of the cluster; then all the current user sessions will be transferred to other servers in the cluster without the users needing to re-login to the application again. ***If your application doesn't have a persistent data store then you can't have seamless fail over for your back-end servers***

Do your web servers store persistent information on local drives?

- Images (jpeg, png, gif etc.)
- Files (html, php, asp etc.)

If so these files either need to be on shared storage such as an NFS/CIFS mount, or they need to be replicated to all of the nodes in the cluster.

### ***Replication solutions for shared data***

On UNIX you can use the RSYNC command to replicate files, on Windows Server you can use RSYNC as well but you may prefer ROBOCOPY from the Windows Server Resource Kit. Usually you will upload your content to one master server and then replicate it to the other servers in the cluster.

### ***Solutions for session data***

Standard ASP and PHP session data is stored locally by default, leaving your session data in a local store will prevent you from implementing seamless application server fail-over in your cluster. If an application server fails, all of the local session data will be lost and your user will need to re-log in and possibly lose shopping baskets etc.

This problem is easily resolvable by implementing a shared persistent data store for the cluster. This is usually either done with a shared back-end database or a shared memory solution.

### ***Persistence***

Persistence is a feature that is required by many web applications. Once a user has interacted with a particular server all subsequent requests are sent to the same server thus persisting to that particular server. It is normally required when the session state is stored locally to the web server as opposed to a database.

## *What do you do if your application is not stateless?*

Some applications require state to be maintained such as:

- Terminal Services
- SSH
- FTP (upload)
- SMTP (incoming)

You may also find that you are unable to modify your HTTP/HTTPS based application to handle shared session data.

For these cases, you can use persistence based on source IP address. You lose the ability to have transparent fail-over, but you do still get increased capacity and manageability. This persistence problem occurs with all load balancers and all vendors use standard methods and technologies to mitigate the issue.

## *Loadbalancer.org persistence methods*

- Source IP (subnet)
- Cookie (Active or Passive)

The standard Layer 4 persistence method is source IP persistence, you can handle millions of persistent connections at Layer 4. Just modify your virtual server to be persistent if you require source IP persistence.

Cookies are a Layer 7 based persistence method that can offer more even traffic distribution and also handle any clients where the source IP address may change during the session (e.g. mega proxies).

## **Section D – Typical Deployment Examples**

## Example 1 – One-Arm DR Mode (Single Appliance)

This DR (Direct Return) mode example has one virtual server (VIP) with two real servers (RIPs). It's a straight forward deployment mode and can be used in many situations. It also offers the highest performance because return traffic passes directly from the real servers to the client (i.e. not via the load balancer).

### *Initial network interface configuration*

Log in to the console:      **Username:** root  
                                 **Password:** loadbalancer

The default IP address is 192.168.2.21/24. To change this, at the console use:

```
ip addr add <IP address>/<mask> dev eth0
```

***Note: This is temporary, the IP address MUST be set via the web interface to make this permanent***

### *Accessing the Web User Interface (WUI)*

With a web browser, access the WUI : ***http://192.168.2.21:9080/lbadmin/***

*(replace 192.168.2.21 with the correct address if this has been changed)*

**Username:** loadbalancer  
**Password:** loadbalancer

***Note:*** If you prefer you can use the HTTPS administration address : ***https://192.168.2.21:9443/lbadmin/***

### *Configuring the load balancer (using the WUI)*

All configuration is performed via the Web User Interface.

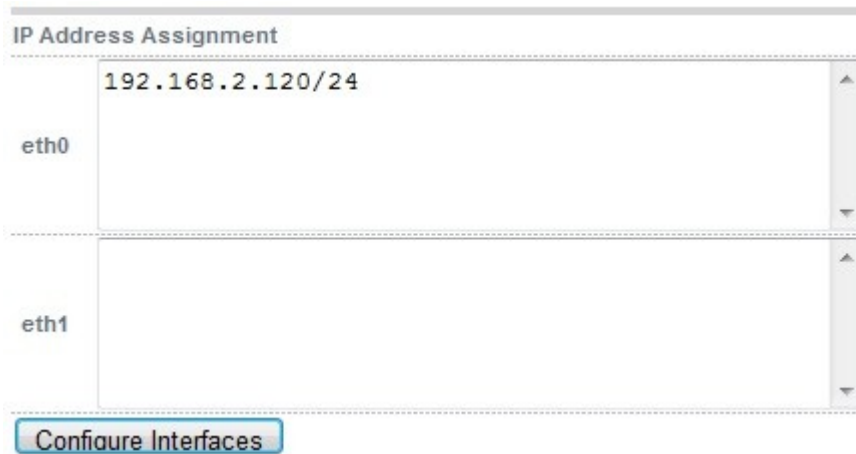
#### Configuration overview

- **Configure Network Settings (WUI)** – A single Interface is needed, eth0 is normally used
- **Define the Virtual Server (WUI)** – All real (back-end) servers are accessed via this IP address
- **Define the Real Servers (WUI)** – Define the real servers
- **Implement the required changes to the real servers** – In DR mode, the ARP issue must be solved

## **Network settings**

Configure the various network settings as outlined below:

- Open *Edit Configuration > Network Interface Configuration*



IP Address Assignment

eth0 192.168.2.120/24

eth1

Configure Interfaces

- Specify the IP address & mask for eth0 (normally eth0 is used for single-arm configurations although this is not mandatory) , e.g. 192.168.2.120/24
- Click **Configure Interfaces**
- Open *Edit Configuration > DNS & Hostname*
- Specify the DNS server(s)



Domain Name Server: 192.168.2.1

Domain Name Server2:

Update

- Click **Update**
- Open *Edit Configuration > Routing*



Routing

Default Gateway IP v4 192.168.2.1

Default Gateway IP v6

Configure Routing

- Specify the default gateway
- Click **Configure Routing**

### Virtual server (VIP)

Next, configure the Virtual Server. This is the IP address that is presented to clients. Any packet arriving at the load balancer with that IP address and port number will be handled by the real servers associated with the virtual server.

- Use *Edit Configuration > Layer 4 Virtual Servers > Add a new Virtual Server*

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="192.168.2.130"/>	?
Virtual Server Ports	<input type="text" value="80"/>	?
Forwarding Method	<input type="text" value="Direct Routing"/>	?
Persistent	<input type="text" value="no"/>	?

- Enter a suitable Label (name) for the VIP
- Enter a valid IP address , e.g. 192.168.2.130
- Enter a valid port , e.g. 80
- Ensure that the Forwarding Method is set to 'Direct Routing' (*Note: this is the default*)

### Real servers (RIP)

Each Virtual Server requires a cluster of real servers (back-end servers) to forward the traffic to.

- Use *Edit Configuration > Layer 4 Real Servers > Add a new Real Server*
- Next to the relevant Virtual Server, click *Add a new Real Server*

Label	<input type="text" value="RIP1"/>	?
Real Server IP Address	<input type="text" value="192.168.2.150"/>	?
Weight	<input type="text" value="1"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?

- Enter a suitable Label (name) for the RIP
- Enter a valid IP address , e.g. 192.168.2.150

*Note:* a port does not need to be specified since port redirection is not possible in DR mode, therefore the port used will be the same as that configured for the VIP

- The weight defaults to 1 making real servers active immediately
- Leave the Minimum & Maximum Connections as 0 which means unrestricted
- Repeat for remaining Real Servers

### **Real server changes - solve the ARP problem**

Since this example uses the one-arm DR mode load balancing method each web server requires the ARP problem to be handled:

- Each server must be configured to respond to the VIP address as well as the RIP address
- Each Windows server must have the MS Loopback Adapter installed and configured
- The MS Loopback Adapter must be configured to deal with the ARP problem



Failure to correctly configure the real servers to handle the ARP problem is the most common problem in DR configurations. Please refer to pages 27-36 or 70-79 for more details.

### **Basic testing & verification**

A few quick checks can be performed to verify the configuration:

- Using *System Overview* , check that the VIP & RIP are shown as active (green)
- Using a browser, navigate to the VIP address , i.e. <http://192.168.2.130> to verify that you can reach the real servers via the Virtual Server
- Check *Reports > Layer 4 Current Connections* to ensure you client connections are reported in state 'ESTABLISHED'. if connections are in state 'SYN\_RECEIVED' , this normally means that the ARP issue on the real servers has not been solved

## Example 2 – Two-Arm NAT Mode (Clustered Pair)

This example covers the process of configuring two load balancers (as a clustered pair) in NAT mode. In this scenario, the slave's network settings must be configured first, followed by the master. This allows the master to successfully communicate with the slave and replicate settings as they are configured.

Using two appliances configured as a clustered pair is Loadbalancer.org's recommended configuration and ensures that no single point of failure is introduced.



When using two-arm NAT mode each web server has to be in the same subnet as the internal interface of the load balancer and the real servers' default gateway must point at an IP address on the load balancer



By default the hardware appliance uses the serial interfaces to transmit / receive heartbeat information , so make sure that you connect the serial cable (one is included with each unit) between the master & slave , by default the Virtual Appliance is configured to use ucast for heartbeat

### *Initial network interface configuration*

Please refer to example 1.

### *Accessing the Web User Interface (WUI)*

Please refer to example 1.

### *Configuring the load balancer (using the WUI)*

#### Configuration overview

- **Configure the Slave's Network Settings** – Two Interfaces are needed, this can be either two physical interfaces such as eth0 and eth1, or one physical interface and an alias/secondary interface such as eth0:0
- **Configure the Master's Network Settings** – Two Interfaces are needed, this can be either two physical interfaces such as eth0 and eth1, or one physical interface and an alias/secondary interface such as eth0:0
- **Configure the Master & Slave Heartbeat Settings** – Set the heartbeat comms method
- **Define the Virtual Server (via the master)** – All IIS servers are accessed via this IP address
- **Define the Real Servers (via the master)** – Define the servers that make up the IIS cluster
- **Implement the required changes to the real servers** – In NAT mode, the IIS servers default gateway must be the load balancer



### **Slave unit – network settings**

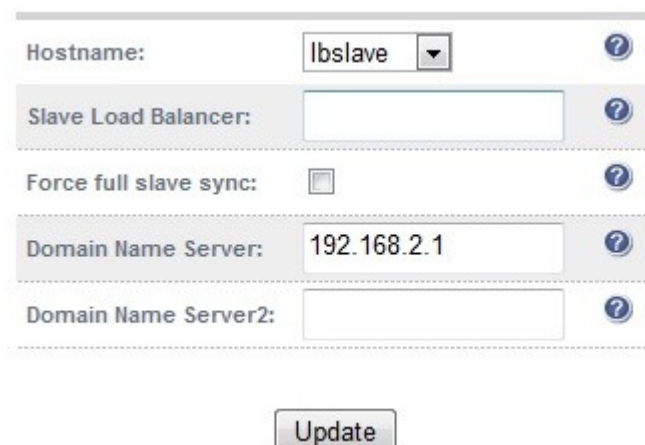
Configure the various network settings as outlined below:

- Open *Edit Configuration > Network Interface Configuration*



The screenshot shows a window titled "IP Address Assignment". It contains two rows of configuration. The first row is for the "eth0" interface, with the IP address "192.168.2.121/24" entered in a text field. The second row is for the "eth1" interface, with the IP address "10.0.0.121/16" entered in a text field. Below the configuration rows is a button labeled "Configure Interfaces".

- Specify the IP address & mask for eth0 (normally eth0 is configured as the internal interface, although this is not mandatory) , e.g. 192.168.2.121/24
- Specify the IP address & mask for eth1 (normally eth1 is configured as the external interface, although this is not mandatory) , e.g. 10.0.0.121/16
- Click **Configure Interfaces**
- Open *Edit Configuration > Hostname & DNS*



The screenshot shows a window for configuring Hostname and DNS. It contains several fields: "Hostname:" with a dropdown menu showing "lslave", "Slave Load Balancer:" with an empty text field, "Force full slave sync:" with an unchecked checkbox, "Domain Name Server:" with the text "192.168.2.1", and "Domain Name Server2:" with an empty text field. Each field has a question mark icon to its right. Below the fields is a button labeled "Update".

- Set the hostname drop-down to '**lslave**'
- Specify the DNS server(s) , e.g. 192.168.2.1
- Click **Update**
- Open *Edit Configuration > Routing*

Routing		
Default Gateway	IP v4	192.168.2.1
	IP v6	

**Configure Routing**

- Specify the default gateway , e.g. 192.168.2.1
- Click **Configure Routing**

### **Master unit – network settings**

Once the slave is configured, continue with the master unit.

- On the master, open *Edit Configuration > Network Interface Configuration*

IP Address Assignment	
eth0	192.168.2.120/24
eth1	10.0.0.120/16

**Configure Interfaces**

- Specify the IP address & mask for eth0 (normally eth0 is configured as the internal interface, although this is not mandatory) , e.g. 192.168.2.120/24
- Specify the IP address & mask for eth1 (normally eth1 is configured as the external interface, although this is not mandatory) , e.g. 10.0.0.120/16
- Click **Configure Interfaces**
- Open *Edit Configuration > Hostname & DNS*

Hostname:	lbmaster ▼	?
Slave Load Balancer:	192.168.2.121	?
Force full slave sync:	<input type="checkbox"/>	?
Domain Name Server:	192.168.2.1	?
Domain Name Server2:		?

**Update**

- Using the hostname drop-down, ensure that the hostname is set to '**lbmaster**'
- Specify the Slave Load balancer's IP address , e.g. 192.168.2.121
- Specify the DNS server(s) , e.g. 192.168.2.1
- Click **Update**
- Open *Edit Configuration > Routing*
- Specify the default gateway

<b>Routing</b>		
Default Gateway	IP v4	192.168.2.1
	IP v6	










**Configure Routing**

- Click **Configure Routing**

## Master & Slave - heartbeat settings

- Open *Edit Configuration > Modify Heartbeat Configuration*

### EDIT CONFIGURATION > MODIFY HEARTBEAT CONFIGURATION

Serial	<input checked="" type="checkbox"/>	
Unicast	<input type="checkbox"/>	
Broadcast	None ▾	
UDP Port for broadcast & unicast	6694	
Keepalive	3	
Deadtime	10	
Warntime	5	
Ping node		
Automatic Fail-back	<input checked="" type="checkbox"/>	






Modify Heartbeat configuration

- Set the heartbeat communications method as required. For a hardware load balancer the default is serial, for a VA the default is unicast (i.e. via the network)

## Virtual server (VIP)

Next, configure the virtual server. This is the IP address that is presented to clients. Any packet arriving at the load balancer with that IP address and port number will be handled by the real servers associated with the virtual server.

- Use *Edit Configuration > Layer 4 Virtual Servers > Add a new Virtual Server*

Label	VIP Name	
Virtual Server IP address	192.168.2.130	
Virtual Server Ports	80	
Forwarding Method	NAT ▾	
Persistent	no ▾	

- Enter a suitable label (name) for the VIP
- Enter a valid IP address , e.g. 192.168.2.130
- Enter a valid port, e.g. 80
- Ensure that the Forwarding Method is set to 'NAT'

### **Real servers (RIP)**

Each Virtual Server requires a cluster of real servers (back-end servers) to forward the traffic to.

- Use *Edit Configuration > Layer 4 Real Servers > Add a new Real Server*
- Next to the relevant Virtual server, click *Add a new Real Server*

Label	<input type="text" value="RIP1"/>	?
Real Server IP Address	<input type="text" value="192.168.2.150"/>	?
Weight	<input type="text" value="1"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?

- Enter a valid IP address , e.g. 192.168.2.150
- Enter a valid port , e.g. 80
- The weight defaults to 1 making real servers active immediately
- Leave the Minimum & Maximum connections as 0 which means unrestricted
- Repeat for the remaining real servers

### **Real server changes – Set the Default Gateway**

As we are using NAT mode, each web servers' default gateway must be changed to be the load balancer. When using a clustered pair, you must define an additional floating IP for this purpose. Then, if failover is required, the same IP will also be brought up on the slave.

To add a floating IP, use *Edit Configuration > Floating IP's* enter the IP address that you'd like to use for the default gateway, then click **update**.

#### **EDIT CONFIGURATION > ADD NEW FLOATING IP**

<input type="text" value="192.168.2.254"/>
<input type="button" value="Update"/>

### **Verify the Slave configuration**

To verify that the new VIP & RIP have been replicated correctly, open the WUI on the slave and goto *Edit Configuration > Layer 4 Virtual Servers & Edit Configuration > Layer 4 Real Servers* and check that your configuration appears there also.

If not, double check that both units are configured correctly and that the IP address for the slave defined on the master is correct. Then on the master open *Edit Configuration > Hostname & DNS* , check 'Force full slave sync' and click **update**, this will force all setting to be copied from the master to the slave, then check again.

### **Restart Heartbeat**

Now restart heartbeat on both units using *Maintenance > Restart Services > Restart Heartbeat*. This ensures that heartbeat starts cleanly and is communicating between the two devices correctly. Once the restart is complete, on the active unit (the master) you should see the floating IP for the corresponding VIP displayed as follows under *View Configuration > Network Configuration*

```
inet 192.168.2.120/24 brd 192.168.2.255 scope global eth0
inet 10.0.0.120/16 brd 192.168.2.255 scope global eth0
inet 192.168.2.130/24 brd 192.168.2.255 scope global secondary eth0
```

- the first two lines show the interface IP address (eth0)
- the last line shows the active floating IP address (VIP)

### **Basic testing & verification**

A few quick checks can be performed to verify the configuration:

- Using *System Overview* , check that the VIP & RIP are shown as active (green)
- Using a browser, navigate to the VIP address, i.e. <http://192.168.2.130> to verify that you can reach the real servers via the Virtual Server
- Check *Reports > Layer 4 Current Connections* to ensure you client connections are reported in state 'ESTABLISHED'. If not, double-check that you have set the default gateway on all real servers to be an IP on the load balancer

### Example 3 - One-Arm SNAT Mode With SSL - HAProxy & Pound (Single Unit)

This example uses HAProxy and Pound at layer 7. Pound is used to terminate the SSL connection on the load balancer. Pound then passes traffic to an HAProxy VIP / RIP cluster. HAProxy does not offer the raw throughput of layer 4, but is still a high performance solution that is appropriate in many situations.

In this example it's assumed that the real server application has not been designed to track & share session details between real servers. Therefore, persistence will be enabled on the load balancer to ensure that clients connect to the same real (back-end) server on each subsequent connection (within the persistence timeout window). If persistence is not configured then new connections may be distributed to a different real server which in this case would result in failure of the application.



Because HAProxy is a full proxy, any server in the cluster can be on any accessible subnet including across the Internet or WAN



In this mode, no changes are required to the real (back-end) servers

#### *Initial network interface configuration*

Please refer to example 1.

#### *Accessing the Web User Interface (WUI)*

Please refer to example 1.

#### *Configuring the load balancer (using the WUI)*

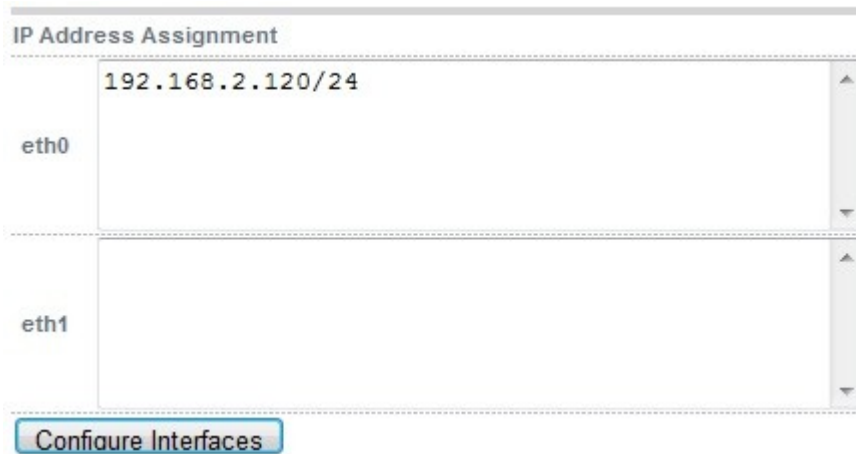
#### Configuration overview

- **Configure Network Settings (WUI)** – A single Interface is needed, eth0 is normally used
- **Define the Virtual Server (WUI)** – All real (back-end) servers are accessed via this IP address
- **Define the Real Servers (WUI)** – Define the real servers
- **Configure SSL Termination** – Configure Pound provide SSL

## **Network settings**

Configure the various network settings as outlined below:

- Open *Edit Configuration > Network Interface Configuration*



IP Address Assignment

eth0 192.168.2.120/24

eth1

Configure Interfaces

- Specify the IP address & mask for eth0 (normally eth0 is used for single-arm configurations although this is not mandatory) , e.g. 192.168.2.120/24
- Click **Configure Interfaces**
- Open *Edit Configuration > DNS & Hostname*
- Specify the DNS server(s)



Domain Name Server: 192.168.2.1 ?

Domain Name Server2: ?

Update

- Click **Update**
- Open *Edit Configuration > Routing*



Routing

Default Gateway IP v4 192.168.2.1

Default Gateway IP v6

Configure Routing

- Specify the default gateway
- Click **Configure Routing**



### Virtual server (VIP)

Next, configure the Virtual Server. This is the IP address that is presented to clients. Any packet arriving at the load balancer with that IP address and port number will be handled by the real servers associated with the virtual server.

- Use *Edit Configuration > Layer 7 Virtual Servers > Add a new Virtual Server*

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="192.168.2.130"/>	?
Virtual Server Ports	<input type="text" value="80"/>	?
Persistence mode	<input type="text" value="HTTP Cookie"/>	?
Fallback Server	<input type="text" value="127.0.0.1:9081"/>	?
<input type="button" value="Update"/>		

- Enter a suitable Label (name) for the VIP
- Enter a valid IP address , e.g. 192.168.2.130
- Enter a valid port , e.g. 80
- Set Persistence mode to '**HTTP Cookie**'
- Restart Haproxy to apply the new settings using the link provided in the yellow box

### Real servers (RIP)

Each Virtual Server requires a cluster of real servers (back-end servers) to forward the traffic to.

- Use *Edit Configuration > Layer 4 Real Servers > Add a new Real Server*
- Next to the relevant Virtual Server, click *Add a new Real Server*

Label	<input type="text" value="RIP1"/>	?
Real Server IP Address	<input type="text" value="192.168.2.150"/>	?
Real Server Port	<input type="text" value="80"/>	?
Weight	<input type="text" value="1"/>	?
<input type="button" value="Update"/>		

- Enter a suitable Label (name) for the RIP

- Enter a valid IP address , e.g. 192.168.2.150

*Note:* in this mode its possible to have a different port for the RIP than was configured for the VIP, in this example both are the same

- Enter a valid port , e.g. 80
- The weight defaults to 1 making real servers active immediately
- Repeat for remaining Real Servers
- Restart Haproxy to apply the new settings using the link provided in the yellow box



The label set for the VIP is used as the name for the HTTP session cookie that is set for use with cookie persistence

### **SSL termination**

Typically, a Pound VIP is configured on port 443 using the same IP address as the Layer 7 VIP created previously. This allows a single IP address to be used.

- Use *Edit Configuration > SSL Termination Virtual Servers*
- Click *Add a New Virtual Server*

Label	VIP Name	?
Virtual Server IP address	192.168.2.130	?
Virtual Server Port	443	?
Backend Virtual Server IP Address	192.168.2.130	?
Backend Virtual Server Port	80	?
Ciphers to use		?
<input type="button" value="Update"/>		

- Enter a suitable Label (name) for the VIP
- Enter the same IP address used for the layer 7 VIP , i.e. 192.168.2.130
- Set the port to 443
- Now define the backend server, as the layer 7 VIP , i.e. 192.168.2.130 , port 80
- Click Update
- Restart Pound to apply the new settings using the link provided in the yellow box

When creating the SSL virtual service, a default self-signed certificate is used. This is ideal for testing but needs to be replaced for live deployments.



For more detailed information on SSL termination, Pound configuration and using Certificates please refer to the SSL Certificates & Pound topic on page 91

### **Basic testing & verification**

A few quick checks can be performed to verify the configuration:

- Using *System Overview* , check that the VIP & RIP are shown as active (green)
- Using a browser, navigate to the VIP address , i.e. <http://192.168.2.130> to verify that you can reach the real servers via the Virtual Server using HTTP
- Using a browser, navigate to the Pound SSL VIP address , i.e. <https://192.168.2.130> to verify that you can reach the real servers via the Virtual Server using HTTPS
- check / verify the certificate details

## **Section E – Detailed Configuration Information**

## Appliance Configuration Methods

The load balancer can be configured in a number of ways as outlined in the following sections.

### Console access

To access the console, connect a monitor and keyboard to the load balancer, power up and you be presented with a login prompt.

Log in to the console:      **Username:** root  
                                 **Password:** loadbalancer



It is recommended to change the default password. To do this type passwd at the console or a terminal window to change the default root password

One of the great advantages of the Loadbalancer.org appliance is that you have full root access and a complete development environment with all of the usual tools you would expect for customizing the installation for your environment.

The following configuration files may be useful:

Physical configuration:	/etc/sysconfig/network-scripts/ifcfg-eth*
Firewall configuration:	/etc/rc.d/rc.firewall
Logical configuration:	/etc/ha.d/conf/loadbalancer.cf
HA-Proxy configuration	/etc/haproxy/haproxy.cfg
Pound SSL configuration	/usr/local/etc/pound.cfg
SSL Certificates	/usr/local/etc/certs
Fail-over (heartbeat) configuration:	/etc/ha.d/ha.cf

For configuration at the console using links, type:

```
links 127.0.0.1:9080/lbadmin
```

Log in using:      **Username:** loadbalancer  
                         **Password:** loadbalancer

This will bring up the text based administration interface by starting the links web browser on the local machine. Use the 'down' cursor key to select a link and the 'right' cursor key to follow a link

## Console access via a serial cable

By default the hardware is shipped with the serial port configured for heartbeat and therefore can't be used for a serial console connection. However if this is your preferred access method then simply go to *Edit Configuration > Heartbeat Configuration* and change the heartbeat to use the network (i.e. ucast or bcast) rather than the serial option. This will automatically activate a console on the serial port.

### **keyboard layout**

to change the keyboard locale edit `/etc/sysconfig/keyboard`

e.g. to change from a UK to a USA layout replace `KEYTABLE="uk"` with `KEYTABLE="us"` , then re-boot.

## Remote configuration methods

Remote configuration is recommended in most cases, but be very cautious if you are changing the network configuration. Make sure you have access to the console if you make a mistake. You can access each load balancer, lbmaster & lbslave remotely via their own IP address using to following tools:

- |                     |                          |
|---------------------|--------------------------|
| • HTTP or HTTPS     | Web based Administration |
| • OpenSSH or PuTTY  | Secure Shell Access      |
| • OpenSCP or WinSCP | Secure File Transfer     |

The default IP address is `192.168.2.21/24`. To change this, at the console use:

```
ip addr add <IP address>/<mask> dev eth0
```

***Note: This is temporary, the IP address MUST be set via the web interface to make this permanent***

For SSH and SCP login as *root* using the password: *loadbalancer*.

For HTTP & HTTPS access, the WUI uses a different set of user accounts and passwords based on the `.htaccess` files. With a web browser, access access the WUI : ***http://192.168.2.21:9080/lbadmin/***

*(replace 192.168.2.21 with the correct address if this has been changed)*

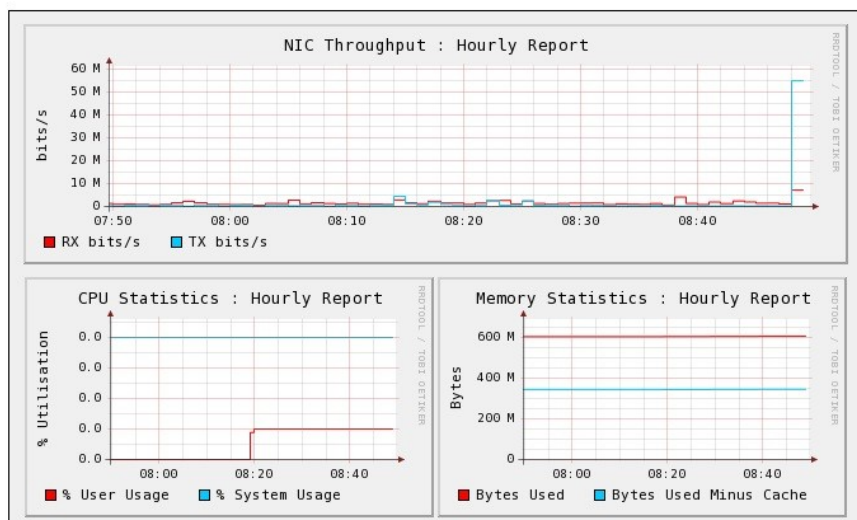
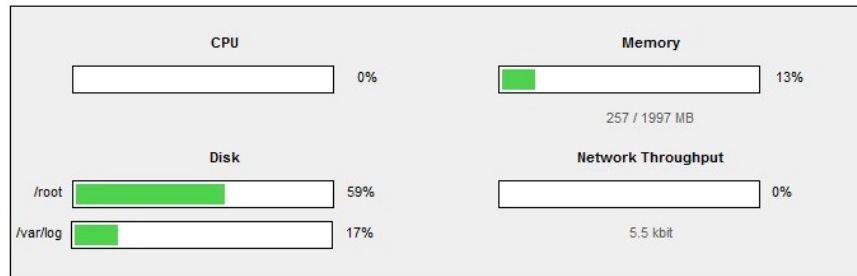
***Username:*** loadbalancer

***Password:*** loadbalancer

***Note:*** *If you prefer you can use the HTTPS administration address : https://192.168.2.21:9443/lbadmin/*

Once logged in, you will be presented with the following screen:

## VIEW CONFIGURATION &gt; SYSTEM OVERVIEW



You can then select an option from one of the main menus. The menu options are as follows:

- **System Overview** : Quickly view system resources, configured VIPs and throughput stats
- **View Configuration** : View the network & load balancer configuration
- **Edit Configuration** : Set up or modify the physical and virtual configuration
- **Maintenance** : Take servers offline or bring them back online
- **Reports** : View the actual live status of the load balancer or historical statistics
- **Logs** : View Ldirectord, Lbadmin, Heartbeat, HAProxy and Pound (SSL)
- **Support** : Create a support download bundle and contact loadbalancer.org support



The first time you access the web interface you will be prompted to run the configuration wizard

## Network Configuration

### IP addresses

New to version 7.x is full IPv6 support. This allows IPv6 services to be configured in the same way as IPv4 services.

Depending on the type of appliance you are using you may have either 2 or 4 network ports. You can manually change the physical IP addresses on the load balancer using *Edit Configuration > Network Interface configuration*.

Normally *eth0* is used as the internal interface and *eth1* is used as the external interface. However, unlike other appliances on the market you can use any interface for any purpose.

In a standard one-arm configuration you would just need to configure *eth0*, the netmask and the default gateway.

### Setting IP addresses

To set the IP address, in the WUI use *Edit Configuration > Network Interface Configuration* shown below:

**EDIT CONFIGURATION > NETWORK INTERFACE CONFIGURATION**

---

**Bonding**

Bond eth0 & eth1 as bond0: ☐ [?](#) [Bond Interfaces](#)

---

**VLAN**

Interface:  [?](#) [Add VLAN](#)

VLAN ID:  [?](#)

---

**IP Address Assignment**

eth0	<div>192.168.2.21/24</div>
eth1	<div></div>

[Configure Interfaces](#)



## Setting multiple addresses

Multiple addresses can be configured per interface as shown below:

### EDIT CONFIGURATION > NETWORK INTERFACE CONFIGURATION

Bonding		
Bond eth0 & eth1 as bond0:	<input type="checkbox"/>	<a href="#">?</a>
Bond eth2 & eth3 as bond1:	<input type="checkbox"/>	<a href="#">?</a>
<a href="#">Bond Interfaces</a>		
VLAN		
Interface:	<input type="text" value="eth0"/>	<a href="#">?</a>
VLAN ID:	<input type="text" value="1"/>	<a href="#">?</a>
<a href="#">Add VLAN</a>		
IP Address Assignment		
eth0	<div>192.168.2.120/24 192.168.8.120/24</div>	
eth1	<div>10.20.1.1/16 fde6:d14c:3089:1::360/64</div>	
<a href="#">Configure Interfaces</a>		

## Configuring bonding

- In the WUI, open *Edit Configuration > Network Configuration*
- If you want to bond eth0 and eth1, check the box named **Bond eth0 & eth1 as bond0**
- Click **Bond Interfaces**
- The eth0 and eth1 fields will be replaced with bond0
- Enter the IP address for bond0 and click **Configure Interfaces**

The screenshot displays a network configuration interface with three main sections:

- Bonding:** Contains two rows. The first row, "Bond eth0 & eth1 as bond0:", has a checked checkbox and a help icon. The second row, "Bond eth2 & eth3 as bond1:", has an unchecked checkbox and a help icon. A "Bond Interfaces" button is located to the right of these rows.
- VLAN:** Features a dropdown menu for "Interface:" set to "bond0" and a text input for "VLAN ID:" set to "1". Both fields have help icons. An "Add VLAN" button is positioned to the right.
- IP Address Assignment:** A table with three rows for "bond0", "eth2", and "eth3". The "bond0" row contains the IP address "192.168.2.74/24". Each row has a vertical scrollbar on the right. Below the table is a "Configure Interfaces" button.

By default, the bond is configured for high-availability. This can be changed by editing `/etc/modprobe.conf` as described in the following section.

## **Bonding configuration modes**

Ideally you want to remove any single point of failure in your network. You can achieve this with a cross-wired switch environment. Every single server including the load balancers is cross wired into two switch fabrics. Then, if a network switch fails the servers & load balancers will activate the connection to the second switch.

Loadbalancer.org appliances support this using the standard Linux bonding driver. Once you have set up the load balancer using a single network card and are happy with the configuration then you can set up bonding.

*NB. You can configure the bonding of network cards using Edit Configuration > Network Interface Configuration.*

If required you can change the bonding mode in the `/etc/modprobe.conf` file:

### Example 1: bonding for bandwidth

```
alias bond0 bonding
options bond0 miimon=100 mode=0
```

Are you really doing 1Gb/s+?

### Example 2: bonding for high-availability (the Default mode)

```
alias bond0 bonding
options bond0 miimon=100 mode=1
```

This works with any switch.

### Example 3: bonding for high-availability & bandwidth

```
alias bond0 bonding
options bond0 miimon=100 mode=4
```

This requires the ports on the switch to be configured as a TRUNK with 802.3ad support.

## Configuring VLANs

Native 8021q VLAN support can be enabled to load balance clusters on multiple VLANs.

In access mode, the switch port is dedicated to one VLAN. The switch handles all the tagging and detagging of frames - the station connected to the port does not need to be configured for the VLAN at all. In trunk mode, the switch passes on the raw VLAN frames, and the station must be configured to handle them. Trunk mode is usually used to connect two VLAN-carrying switches, or to connect a server or router to a switch.

If the load balancer is connected to an access mode switch port there is no VLAN configuration needed. If the load balancer is connected to a trunk port, then all the required VLANs will need to be configured under Network Config.

To configure a VLAN:

- Select the required interface (e.g. eth0)
- Enter the VLAN ID (e.g. 100)
- Click **Add VLAN**
- An extra IP Address Assignment field named eth0.100 will be created as shown below. The required IP address should be entered

IP Address Assignment	
eth0	192.168.1.1/24
eth0.100	192.168.100.1/24
eth1	

[Delete eth0.100](#)

[Configure Interfaces](#)

- Click **Configure Interfaces**

To delete the VLAN definition, click the appropriate **Delete** button



If you have a clustered pair, don't forget to configure any VLANs on the slave as these will not be replicated / created automatically

## Default gateway

The default gateway can be set for both IPv4 and IPv6 as shown below.

### EDIT CONFIGURATION > ROUTING

Routing		
Default Gateway	IP v4	192.168.2.1
	IP v6	fde6:d14c:3089:1::1
<button>Configure Routing</button>		

## Hostname & DNS configuration

The hostname of each appliance must be set to either 'lbmaster' or 'lbslave', the default is 'lbmaster'. To change this, use the Hostname drop-down as shown below.

DNS servers are defined in using the Domain **Name Server** and **Domain Name Server2** fields.

### EDIT CONFIGURATION > HOSTNAME & DNS

Hostname:	lbmaster ▼	?
Slave Load Balancer:	192.168.2.121	?
Force full slave sync:	<input type="checkbox"/>	?
Domain Name Server:	192.168.2.1	?
Domain Name Server2:		?
<button>Update</button>		

This screen is also used to:

- Setup the IP address of the slave unit
- Force a full sync from master appliance to the slave appliance

## Advanced DR Considerations

The most important consideration with DR is how to handle the ARP problem.



The ARP problem only effects layer 4 DR (Direct Return) mode VIPs and therefore it is only necessary to implement the changes to the real servers described below when using this mode

### What Is the ARP problem?

It is important that your web servers do not fight with the load balancer for control of the shared VIP. If they do then request will be sent directly to the web servers rather than hitting the load balancer VIP as intended.

- You only need to resolve the ARP issue on the real servers when you are using the default DR (Direct Routing) load balancing method or IPIP (TUN or IP encapsulation).
- If you are using NAT mode you don't need to make any changes to the real servers except to make sure the load balancers IP address needs to be set as the default gateway.
- SSL termination and Layer 7 SNAT modes do not require any changes to the Real Servers.

### Detecting the ARP problem

You can use *Reports > Layer 4 Current Connections* to check whether the ARP problem has been solved. If not, the connection state will be SYN\_RECV as shown below when a client connection to the VIP is attempted:

#### REPORTS > LAYER 4 CURRENT CONNECTIONS

```
IPVS connection entries
pro expire state      source          virtual         destination
TCP 00:51 SYN_RECV    192.168.2.7:64763 192.168.2.109:80 192.168.2.99:80
```

### Solving for Linux – method 1 (using iptables)

You can use iptables (netfilter) on each real server to re-direct incoming packets destined for the virtual server IP address. To make this permanent, simply add the command to an appropriate start-up script such as /etc/rc.local. If the real server is serving multiple VIPs, add additional iptables rules for each VIP.

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

e.g.

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

this means redirect any incoming packets destined for 10.0.0.21 (the virtual server) locally.



Method 1 does not work with IPv6 Virtual Servers, use method 2 below instead

### **Solving for Linux – method 2 (using arp ignore sysctl values)**

This is the preferred method as it supports both IPv4 and IPv6. Each real server needs the loopback adapter to be configured with the Virtual Servers IP address. This address must not respond to ARP requests and the web server also needs to be configured to respond to this address. To set this up follow steps 1-3 below.

#### **Step 1 : re-configure ARP on the real servers (this step can be skipped for IPv6 virtual servers)**

To do this add the following lines to /etc/sysctl.conf:

```
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
```

#### **Step 2 : apply these settings**

Either reboot the real server or run the following command to apply these settings:

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

#### **Step 3 : add the virtual servers' IP address to the loopback adapter**

run the following command for each VIP. To make this permanent, simply add the command to an appropriate startup script such as /etc/rc.local.

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

*for IPv6 addresses use:*

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

Alternatively, modify the appropriate interface script to add the additional IP address(es).



Failure to correctly configure the real servers to handle the ARP problem is the most common mistake in DR mode configurations

### **Solving for Solaris**

With Solaris the loopback interface does not respond to ARP requests so you just add your VIPs to it.

```
ifconfig lo0:1 plumb  
ifconfig lo0:1 VIP netmask 255.255.255.255 up
```

You will need add this to your start up scripts for your server.

### **Solving for Mac OS X or BSD**

OS X is BSDish, so you need to use BSDish syntax:

```
ifconfig lo0 alias VIP netmask 255.255.255.255 -arp up
```

You will need add this to your start up scripts for your server.



Failure to correctly configure the real servers to handle the ARP problem is the most common mistake in DR mode configurations

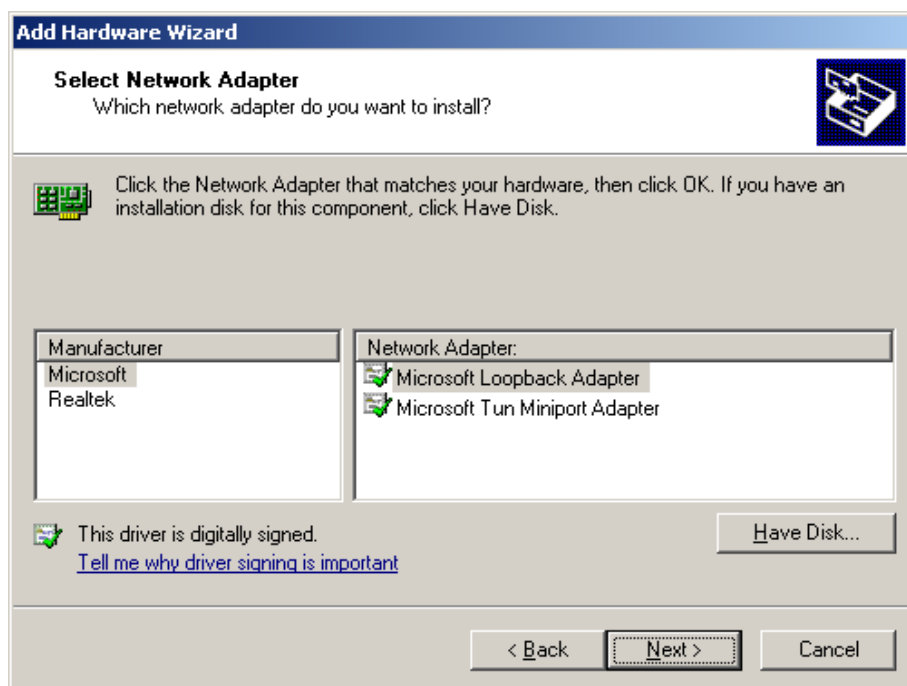


## **Solving for Windows 2000 / 2003**

Windows server supports the direct routing (DR) method through the use of the MS loopback adapter to handle the traffic. The IP address on the loopback adapter must be set to be the same as the Virtual Servers IP address (VIP). If the real server is included in multiple VIPs, you can add additional IP addresses to the loopback adapter that correspond to each VIP.

### **Installing the Microsoft loopback adapter**

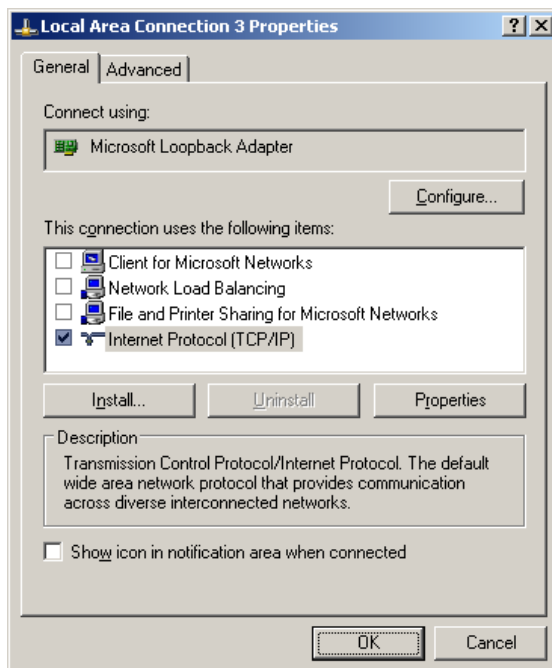
1. Open the Control Panel and double-click Add Hardware
2. Once the Hardware Wizard opens, click Next
3. Select 'Yes, I have already connected the hardware', click Next
4. Scroll to the bottom of the list, select 'Add a new hardware device' and click Next
5. Select 'Install the hardware that I manually select from a list (Advanced)', click Next
6. Select 'Network adapters', click Next
7. Select 'Microsoft' & 'Microsoft Loopback Adapter', click Next



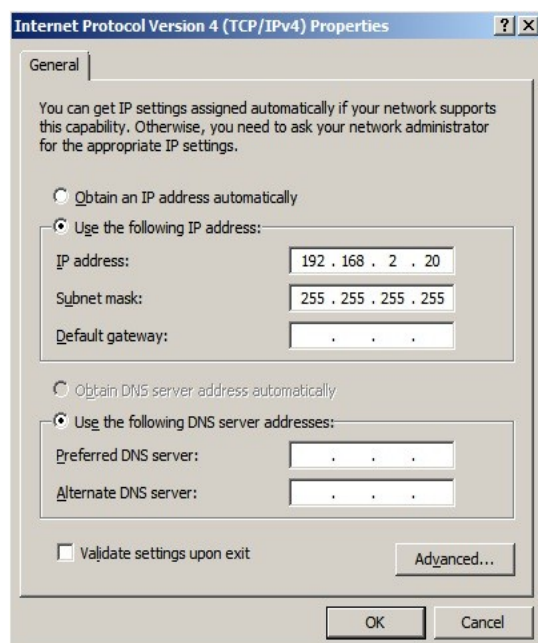
8. Click Next to start the installation, when complete click Finish

## Configuring the loopback adapter

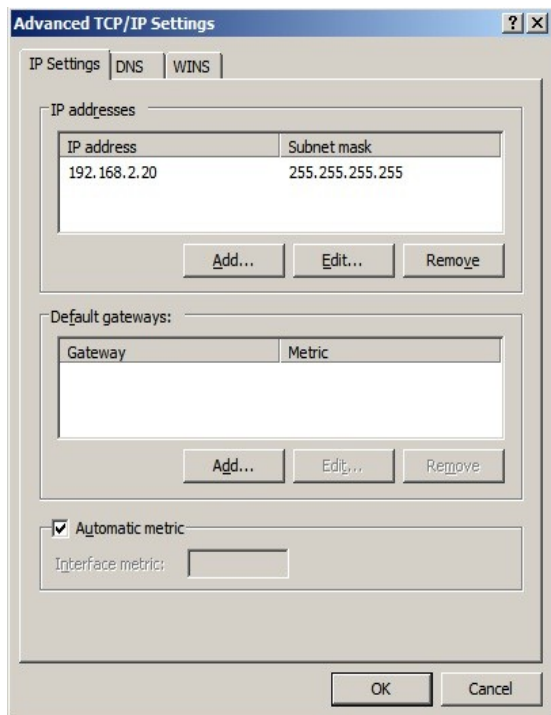
1. Open the Control Panel and double-click Network Connections



2. Right click the new loopback adapter and select properties
3. Un-check all items except Internet Protocol (TCP/IP)
4. Select Internet Protocol (TCP/IP), click Properties and configure the IP address to be the same as the Virtual Server (VIP) with a full subnet mask, e.g. 192.168.2.20/32



5. Click on the *Advanced* button and change the Interface Metric to 254 (This stops the adapter responding to ARP requests).



6. Click OK on the Advanced and TCP/IP popup windows, then click Close on the Local Area Connection window to save the new settings
7. Now repeat the above process for all other Windows 2000 / 2003 real servers

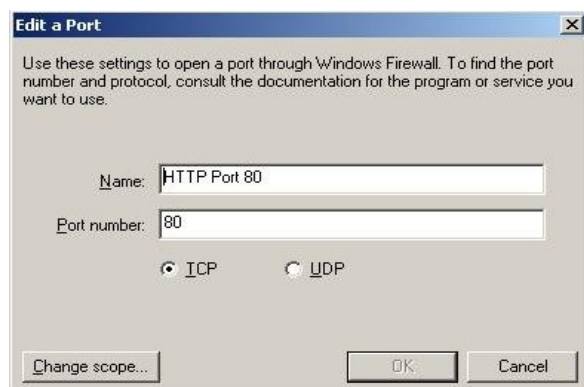


For Windows server 2003 SP1 & above, if you have enabled the built-in firewall, you will need to create an exception to enable access to the web server. This exception by default will allow traffic on both the network and loopback adapters.

#### Windows 2003 R2 / R1 (With SP1) firewall settings

Windows 2003 only allows control of inbound connections.

Enable the firewall for both the LAN connection and the loopback adapter. Check this using the Advanced tab in the Windows Firewall tool. Then add a firewall exception to open the relevant port, e.g. port 80 for http traffic as shown below:

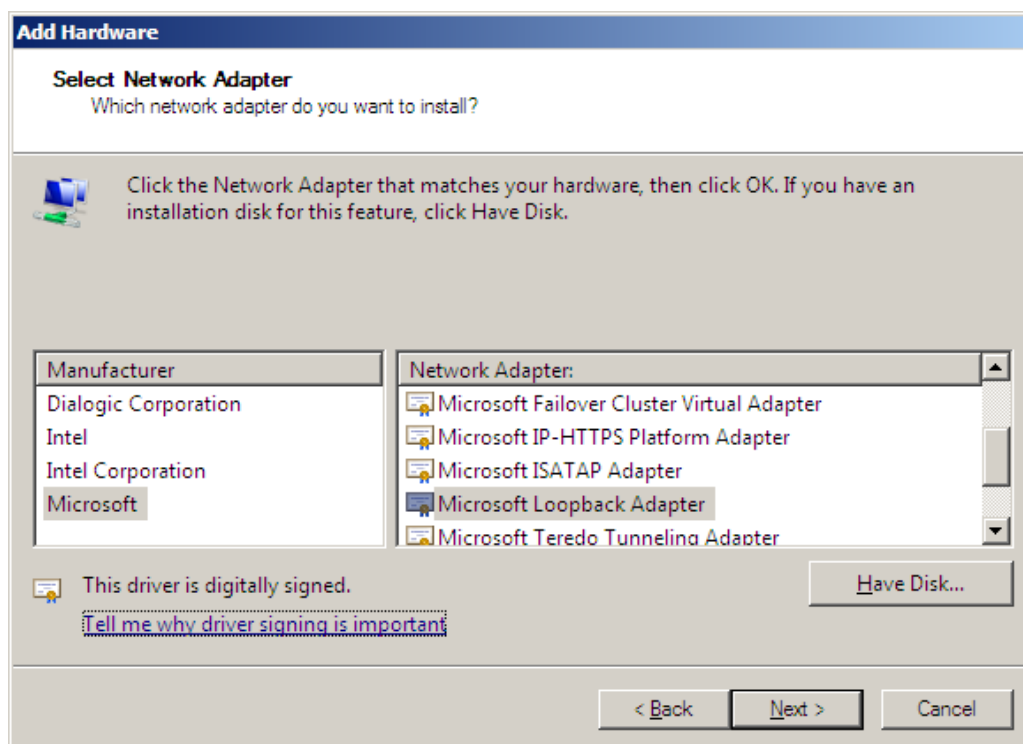


## **Solving for Windows 2008**

The basic concept is the same as for Windows 2000 / 2003. However, additional steps are required to set the strong / weak host behavior. This is used to either block or allow interfaces receiving packets destined for a different interface on the same server.

### **Installing the Microsoft loopback adapter**

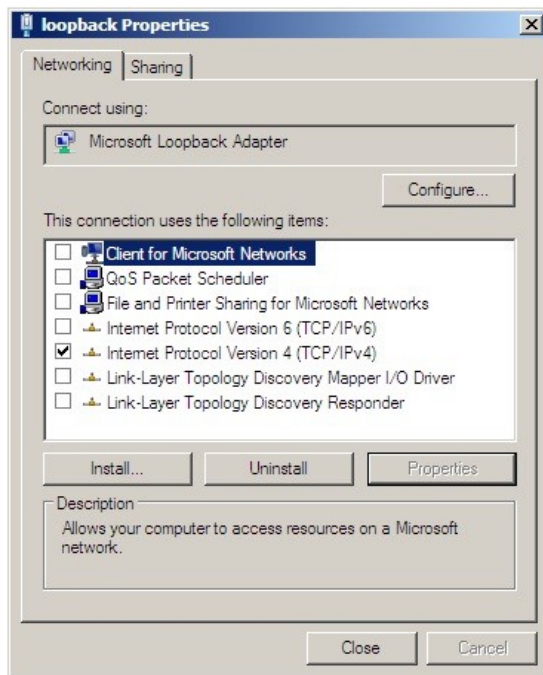
1. Click Start, select Run and enter **hdwwiz** to start the Hardware Installation Wizard
2. When the Wizard has started, click Next
3. Select 'Install the hardware that I manually select from a list (Advanced)', click Next
4. Select 'Network adapters', click Next
5. Select 'Microsoft' & 'Microsoft Loopback Adapter', click Next



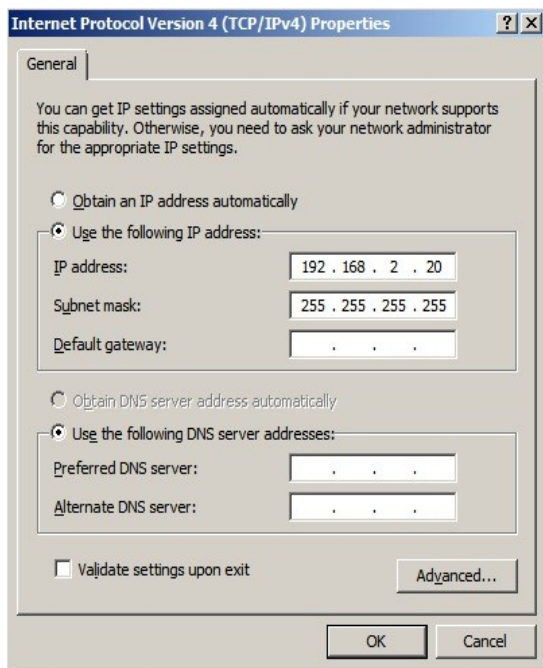
6. Click Next to start the installation, when complete click Finish

## Configuring the loopback adapter

1. Open Control Panel and double-click Network and Sharing Center
2. Click Change adapter settings
3. Right-click the new loopback adapter and select Properties



4. Un-check all items except Internet Protocol Version 4 (TCP/IPv4)
5. Select Internet Protocol Version (TCP/IPv4), click Properties and configure the IP address to be the same as the Virtual Server (VIP) with a full subnet mask, e.g. 192.168.2.20/32



6. Click OK on the TCP/IP popup window, then click Close on the Local Area Connection window to save the new settings

7. For Windows 2008, its not necessary to modify the interface metric on the advanced tab and should be left set to Automatic.
8. Now repeat the above process for all other Windows 2008 real servers

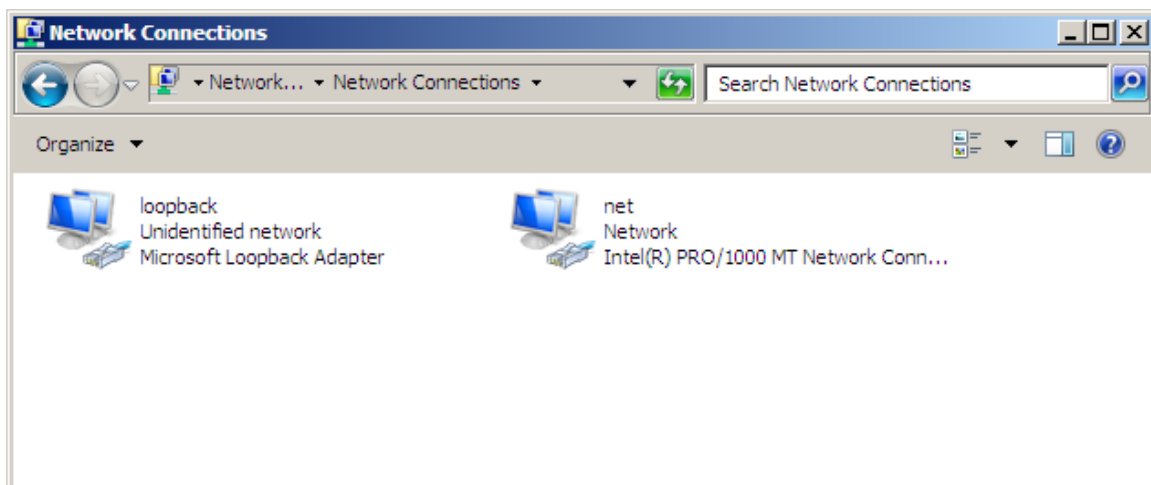
### **Configuring strong / weak host behavior**

Windows XP and Windows Server 2003 use the weak host model for sending and receiving for all IPv4 interfaces and the strong host model for sending and receiving for all IPv6 interfaces. You cannot configure this behavior. The Next Generation TCP/IP stack in Windows Vista and Windows Server 2008 supports strong host sends and receives for both IPv4 and IPv6 by default.

To ensure that the Windows 2008 is running in the correct mode to respond to the VIP, the following commands must be run in a command window on the real server :

```
netsh interface ipv4 set interface "net" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostsend=enabled
```

For these commands to work, the LAN connection NIC must be named “net” and the loopback NIC must be named “loopback”. If you prefer to leave your current NIC names, then the commands above must be modified accordingly.



*N.B. The names for the NICs are case sensitive, so make sure that the name used for the interface and the name used in the commands match exactly.*

If you prefer to use the index number for the interface, you can look up the index number using the following command:

```
netsh interface ipv4 show interface
```

then substitute the relevant index number for “net” and “loopback” in the three netsh commands

## Verifying netsh Settings

To verify that settings have been configured correctly, run the following command on the Windows server to clearly list the settings that have been applied to the interface:

```
netsh interface ipv4 show interface <interface name>
```

i.e. for the 'loopback' adapter run :netsh interface ipv4 show interface loopback

i.e. for the 'net' adapter run :netsh interface ipv4 show interface net

e.g.

```
C:\Users\Administrator>netsh interface ipv4 show interface loopback
```

Interface loopback Parameters

```
-----
IfLuid                : ethernet_9
IfIndex               : 15
State                 : connected
Metric                : 30
Link MTU              : 1500 bytes
Reachable Time        : 28500 ms
Base Reachable Time   : 30000 ms
Retransmission Interval : 1000 ms
DAD Transmits         : 3
Site Prefix Length    : 64
Site Id               : 1
Forwarding             : disabled
Advertising           : disabled
Neighbor Discovery    : enabled
Neighbor Unreachability Detection : enabled
Router Discovery      : dhcp
Managed Address Configuration : enabled
Other Stateful Configuration : enabled
Weak Host Sends       : enabled
Weak Host Receives    : enabled
Use Automatic Metric  : enabled
Ignore Default Routes : disabled
Advertised Router Lifetime : 1800 seconds
Advertise Default Route : disabled
Current Hop Limit     : 0
Force ARPND wake up patterns : disabled
Directed MAC wake up patterns : disabled
```

```
C:\Users\Administrator>
```

the above screen shot shows that the settings have been applied correctly.



For Windows server 2008, if you want to leave the built-in firewall enabled, you'll either need to enable the relevant default firewall exceptions or create your own to enable access to the web server. By default these exceptions will allow traffic on both the network and loopback adapters



Failure to correctly configure the real servers to handle the ARP problem is the most common mistake in DR configurations

## Windows 2008 R2 firewall settings

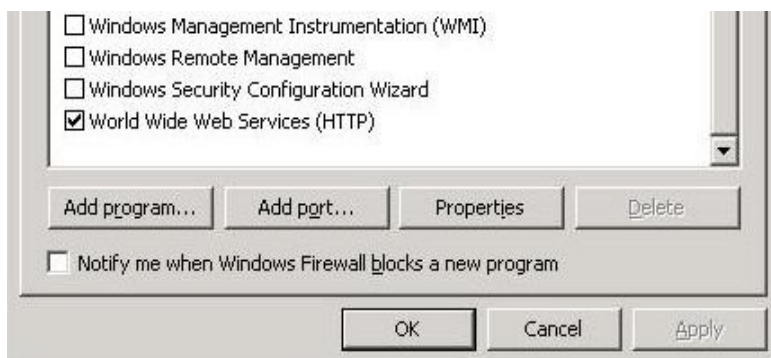
Windows 2008 automatically creates several default firewall rules for both inbound and outbound traffic. By default, all outbound traffic is allowed and all inbound traffic is blocked except where a rule allows it. Outbound rules can also be enabled if necessary. There are 3 firewall policies and interfaces can be associated with one of these 3 policies (domain, private and public) although the loopback adapter automatically gets associated with the public profile and this cannot be changed.

For a web server listening on port 80 the following default http rules need to be enabled as shown below:



## Windows 2008 R1 firewall settings

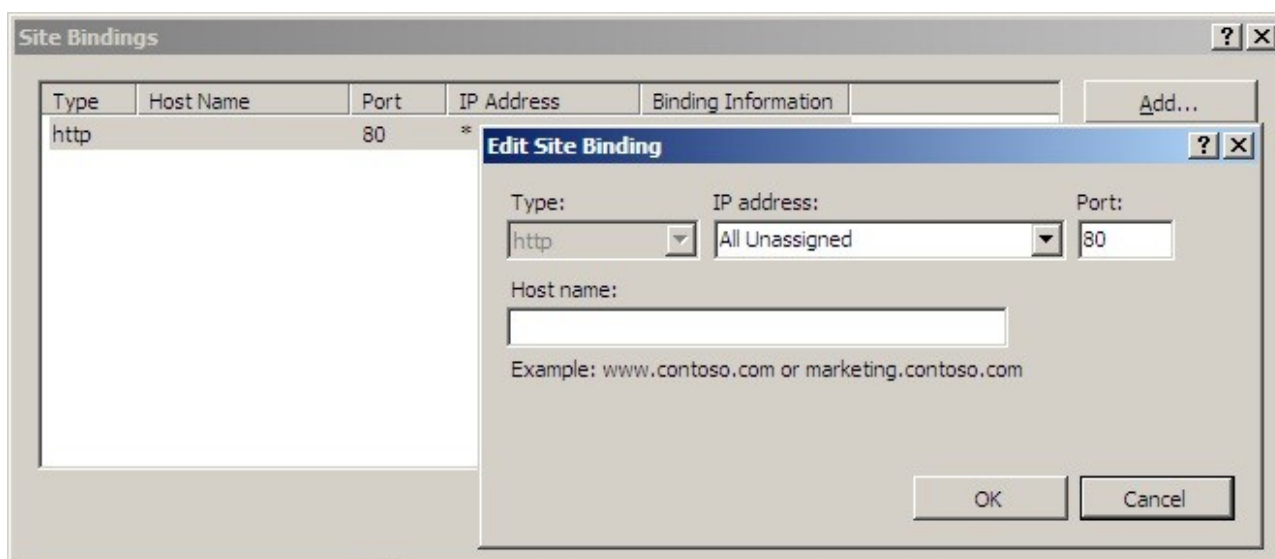
For Windows 2008 R1 the firewall configuration is very similar to windows 2003 R2 except that a default rule gets created automatically that can be enabled to permit port 80 HTTP traffic. You just need to enable the firewall for both interfaces then ensure that the WWW service check-box is ticked as shown below:



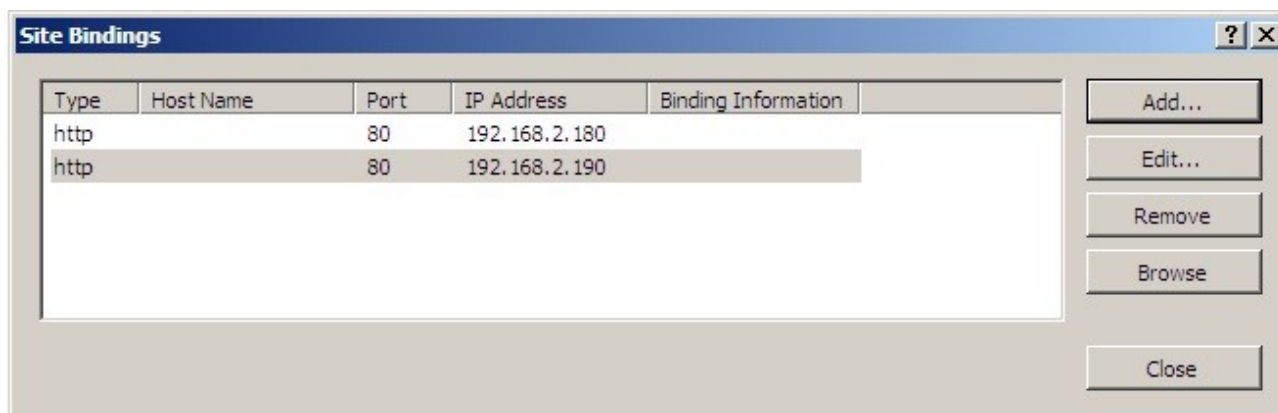


## Configuring IIS to respond to both the RIP and VIP

For DR mode, Its also important to make sure that IIS responds to both the VIP and RIP. By default, IIS listens on all configured IP addresses, this is shown in the example below. As can be seen the IP address field is set to "All Unassigned".



If the default configuration is left, no further IIS configuration is required. If you do change the IP address in the bindings from "All Unassigned" to a specific IP address, then you need to make sure that you also add a binding for the Virtual Server IP address (VIP) as shown below:



## Advanced NAT considerations

The NAT style of load balancing does have the advantage that the only change to the real servers is to modify the default gateway, IP address and subnet. You can also utilize the added security of having your real servers hidden in a subnet behind the load balancer. However, in our honest opinion, we think it is not wise to use your load balancer as a firewall. It adds complexity, and while the Loadbalancer.org appliance can be configured to be rock solid secure, *you should at least be fully aware of what you are doing if it is going to be your bastion host.*

There is no harm in putting a pair of Loadbalancer.org appliances in NAT mode behind your own firewall solution as shown in the example 2 diagram.

In order to use NAT mode on the load balancers you'll need a couple of things:

1. You need an external and internal floating VIP (Floating Virtual IP address)
2. The external one is the one the clients connect to
3. The internal one is the default gateway for the real servers
4. Set your virtual server to use the NAT method and hey presto you are done

BUT :

1. Your real servers won't be able to access the Internet through the new default gateway (except when replying to requests made through the external VIP)
2. Non-load balanced services on the real servers (e.g. RDP for management access to Windows servers) will not be accessible since these have not been exposed via the load balancer

To solve problem #1

When NAT mode is selected in the setup wizard, the autonat feature will be automatically enabled. If you need to do this for a manual configuration, turn autonat on in global options. This activates the rc.nat script that forces external network traffic to be MASQUERADED to and from the external network.

```
#/etc/rc.d/rc.nat
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```



If you have used the wizard to configure the load balancer then this will automatically have generated a MASQUERADE rule in the /etc/rc.d/rc.nat file. This rule will automatically masquerade all traffic from the internal network via eth0 to eth1 (external)

### AUTONAT

*NB. If you configure NAT mode manually don't forget to set the autonat interface to eth1 in global options. If the wizard is used, autonat is turned on automatically.*

To solve problem #2

If you want any specific services to be exposed for your real servers you have two choices:

a) Set up a specific virtual server with a single real server for the service i.e. Just one real server in the FTP group.

or

b) Set up individual public IPs for the services required with individual SNATs and DNATs for each service required as shown below:

```
# SNAT & DNAT all traffic from EXT_ADDR to INT_ADDR
INT_ADDR="192.168.1.13"
EXT_ADDR="234.23.45.236"

iptables -t nat -A POSTROUTING -o $EXT_IFACE -p tcp -s $INT_ADDR -j SNAT --to-source $EXT_ADDR
iptables -t nat -A PREROUTING -i $EXT_IFACE -p tcp -d $EXT_ADDR -j DNAT --to-destination $INT_ADDR
```

Any specific SNAT and DNAT commands must be run before the generic autonat script rc.nat. You should probably disable autonat to stop it interfering with your rules in global options and then put the equivalent command at the end of the firewall script if you also require other internal servers to use autonat. Or you could modify the *rc.nat* script as in the following example:

```
#!/bin/sh
#/etc/rc.d/rc.nat

# SNAT & DNAT all traffic from INT(10.0.0.55) to EXT(192.168.2.43)
iptables -t nat -A POSTROUTING -o eth1 -p tcp -s 10.0.0.55 -j SNAT --to-source 192.168.2.43
iptables -t nat -A PREROUTING -i eth1 -p tcp -d 192.168.2.43 -j DNAT --to-destination 10.0.0.55

# Allow all internal servers to access the external network using NAT
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Example firewall settings output when using a pair of SNAT & DNAT rules followed by autonat:

```
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination
DNAT       tcp  --  0.0.0.0/0              192.168.2.43          to:10.0.0.55

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
SNAT       tcp  --  10.0.0.55              0.0.0.0/0             to:192.168.2.43
MASQUERADE all  --  0.0.0.0/0              0.0.0.0/0
```



Don't hesitate to contact [support@loadbalancer.org](mailto:support@loadbalancer.org) to discuss any specific requirements you may have.

### Explaining the RIP & VIP in NAT mode

RIP is the Real IP address of a back-end server and VIP is the Virtual IP address of the cluster. You can have as many VIPs as you like but for this example we are only using one.

*NB. NAT mode routing is a common and very effective standard routing technique used in firewalls*

The following figure illustrates the rules specified for the load balancer in NAT mode:

Protocol	VIP	Port	RIP	Port
TCP	10.0.0.20	80	192.168.2.50	80

All traffic destined for IP address 10.0.0.20 Port 80 is load-balanced over real IP address 192.168.1.50 Port 80. Packet rewriting works as follows:

The incoming packet for the web service has source and destination addresses as:

SOURCE x.x.x.x:3456 DEST 10.0.0.20:80

The packet would be rewritten and forwarded to the back-end server as:

```
SOURCE      x.x.x.x:3456          DEST      192.168.1.50:80
```

Replies get back to the load balancer as:

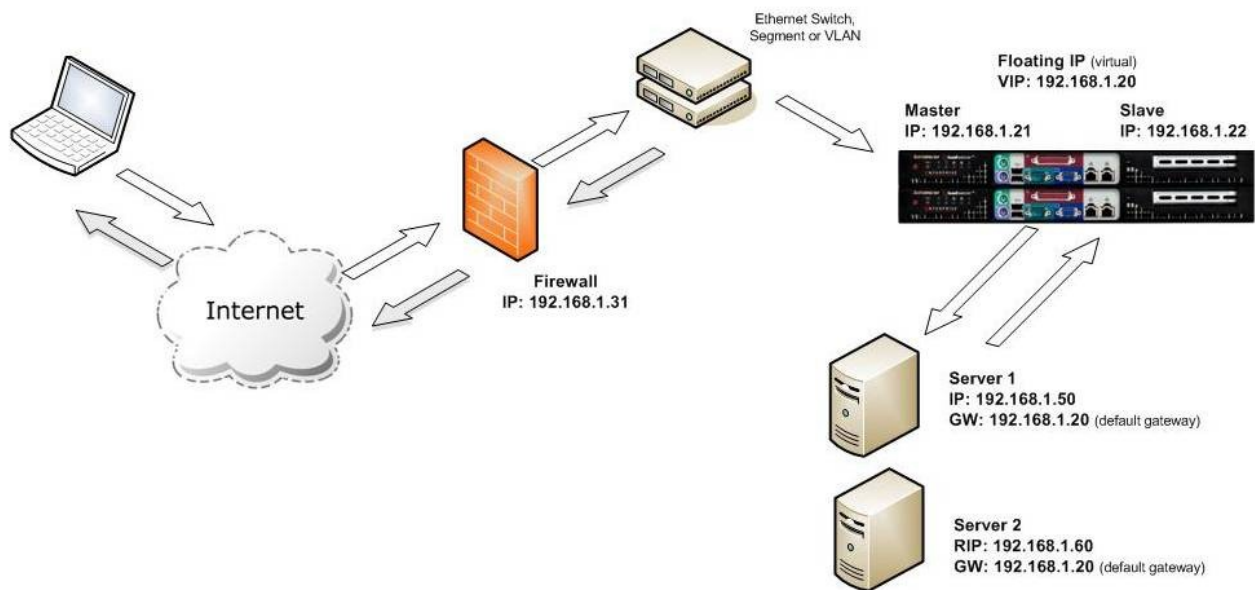
SOURCE 192.168.1.50:80 DEST x.x.x.x:3456

The packets would be written back to the VIP address and returned to the client as:

SOURCE 10.0.0.20:80 DEST x.x.x.x:3456

Notes:

- In NAT mode the source IP address is preserved i.e. back-end server logs client IP address.
- The back-end server RIP must have its default gateway pointing at the load balancer
- The back-end server must be on the internal subnet
- Servers on the internal subnet cannot access the external VIP
- NAT mode allows you to do port translation i.e. have a different RIP port than the VIP port



### Notes:

- One-arm (single subnet) NAT load balancing works well for external clients.
- For internal clients (same subnet) the route table of each real server needs modification.
- Administration of the load balancers is via any active IP address.
- A floating IP must be configured for hosting the virtual server.
- The default gateway of the real servers must point at the load balancers floating IP.



When using a clustered pair of load balancers in one-arm NAT mode all load balanced services must be configured on a floating IP. To access the load balanced services from the same subnet special routing rules must be added to the real servers

## *Route configuration for Windows Server with one-arm NAT mode*

When a client on the same subnet as the real server tries to access the virtual server on the load balancer the request will fail. The real server will try to use the local network to get back to the client rather than going through the load balancer and getting the correct network translation for the connection.

To rectify this issue we need to add a route to the the load balancer that takes priority over Windows default routing rules.

This is a simple case of adding a permanent route:

```
route add -p 192.168.1.0 mask 255.255.255.0 metric 1
```

*NB. Replace 192.168.1.0 with your local subnet address.*

The default route to the local network has a metric of 10, so this new route overrides all local traffic and forces it to go through the load balancer as required.

Any local traffic (same subnet) is handled by this route and any external traffic is handled by the default route (which also points at the load balancer).

## *Route configuration for Linux with one-arm NAT mode*

When a client on the same subnet as the real server tries to access the virtual server on the load balancer the request will fail. The real server will try to use the local network to get back to the client rather than going through the load balancer and getting the correct network translation for the connection.

To rectify this issue we need to modify the local network route to a higher metric:

```
route del -net 192.168.1.0 netmask 255.255.255.0 dev eth0  
route add -net 192.168.1.0 netmask 255.255.255.0 metric 2000 dev eth0
```

*NB. Replace 192.168.1.0 with your local subnet address.*

Then we need to make sure that local network access uses the load balancer as its default route:

```
route add -net 192.168.1.0 netmask 255.255.255.0 gateway 192.168.1.21 metric 0 dev eth0
```

*NB. Replace 192.168.1.21 with your load balancer gateway*

Any local traffic (same subnet) is handled by this manual route and any external traffic is handled by the default route (which also points at the load balancer).

## Advanced Layer 7 Considerations

### *Load balancing based on URL match with HAProxy*

*We're currently building this into the GUI but for now you'll have to edit the config file directly.*

The structure of the HAProxy config file changes quite a lot when you choose to use ACLs. Here's a simple example below:

```
# HAProxy configuration file generated by load balancer appliance
global
uid 99
gid 99
daemon
stats socket /var/run/haproxy.stat mode 600
maxconn 40000
ulimit-n 65536
pidfile /var/run/haproxy.pid
defaults
mode http
contimeout 4000
clitimeout 42000
srvtimeout 43000
balance roundrobin

frontend f1
bind 192.168.2.112:80
acl test_acl1 path_beg /test1
acl test_acl2 path_beg /test2
use_backend b1 if test_acl1
use_backend b2 if test_acl2
default_backend b2
option httpclose

backend b1
cookie SERVERID insert nocache indirect
server s1 192.168.2.99:80 weight 1 cookie s1 check
server s2 192.168.2.10:80 weight 1 cookie s2 check

backend b2
cookie SERVERID insert nocache indirect
server s3 192.168.2.6:80 weight 1 cookie s3 check
```

So instead of the usual 'listen' directive (which groups the virtual server and its real backends together), we now have separate frontend and backend sections.

In this example we have 'test\_acl1' <-- just a label, 'path\_beg' <--- i.e. match path beginning with... 'test1'. And similarly for test\_acl2. There are numerous matching options available.

For more details refer to: <http://haproxy.1wt.eu/download/1.5/doc/configuration.txt>

the search that page for 'path\_beg'

## *HAProxy error codes*

For reference, the layer 7 HAProxy error codes are as follows:

Code	When / Reason
200	access to stats page, and when replying to monitoring requests
301	when performing a redirection, depending on the configured code
302	when performing a redirection, depending on the configured code
303	when performing a redirection, depending on the configured code
400	for an invalid or too large request
401	when an authentication is required to perform the action (when accessing the stats page)
403	when a request is forbidden by a "block" ACL or "reqdeny" filter
408	when the request timeout strikes before the request is complete
500	when haproxy encounters an unrecoverable internal error, such as a memory allocation failure, which should never happen
502	when the server returns an empty, invalid or incomplete response, or when an "rspdeny" filter blocks the response.
503	when no server was available to handle the request, or in response to monitoring requests which match the "monitor fail" condition
504	when the response timeout strikes before the server responds

Complete detailed information for HAProxy configuration is available here:

<http://haproxy.1wt.eu/download/1.5/doc/configuration.txt>



## Configuring VIPs & RIPv via Command Line / Script

If required it is possible to add, remove and edit Virtual / Real servers via the command line. This enables loadbalancer configuration changes to be made via script rather than using the WUI. This is not a full API, but it does allow basis manipulation of load balanced services. *NB. An API is currently in development and will be included in a later release.*

### *Layer 4*

For layer 4, the ipvsadm command is used. Several examples are provided below.

*Add a TCP based Virtual Server , use round robin scheduling:*

```
ipvsadm -A -t 192.168.65.192:80 -s rr
```

*Add a TCP based Real Server in DR mode:*

```
ipvsadm -a -t 192.168.65.192:80 -g -r 192.168.70.196:80
```

*Add a TCP based Real Server in NAT mode:*

```
ipvsadm -a -t 192.168.65.192:80 -m -r 192.168.70.196:80
```

*Add a UDP based Virtual Server , use least connection scheduling:*

```
ipvsadm -A -u 192.168.65.192:80 -s lc
```

*Add a UDP based Real Server in DR mode:*

```
ipvsadm -a -u 192.168.65.192:80 -g -r 192.168.70.196:80
```

*Delete a TCP based Virtual Server:*

```
ipvsadm -D -t 192.168.65.180:80
```

*Delete a TCP based Real Server:*

```
ipvsadm -d -t 192.168.65.122:80 -r 192.168.70.134:80
```

*View the current running config:*

```
ipvsadm -ln
```

```
IP Virtual Server version 1.2.1 (size=4096)
```

```
Prot LocalAddress:Port Scheduler Flags
```

	-> RemoteAddress:Port		Forward	Weight	ActiveConn	InActConn
TCP	192.168.65.120:80	rr				
	-> 192.168.70.130:80	Route	1	0	0	
	-> 192.168.70.131:80	Route	1	0	0	
TCP	192.168.65.122:80	rr				
	-> 192.168.70.132:80	Masq	1	0	0	
	-> 192.168.70.133:80	Masq	1	0	0	

## Layer 7

For layer 7 Haproxy VIPs, the socat socket command can be used as shown in the examples below.

*To take a server offline:*

```
echo "disable server VIP_Name/RIP_Name" | socat unix-connect:/var/run/haproxy.stat stdio
```

*To bring a server online:*

```
echo "enable server VIP_Name/RIP_Name" | socat unix-connect:/var/run/haproxy.stat stdio
```

*To set the weight of a real server:*

```
echo "set weight VIP_Name/RIP_Name 0" | socat unix-connect:/var/run/haproxy.stat stdio
```

*NB. Other examples can be found by searching for "Unix Socket Commands" at the following link:*

<http://haproxy.1wt.eu/download/1.5/doc/configuration.txt>

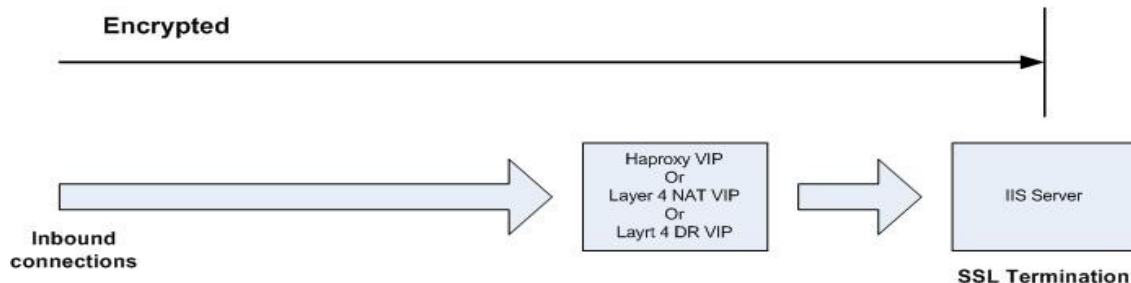


Please note that since these changes are being made directly to the running configuration, the services that are displayed in the System Overview will not match the running configuration

## SSL Termination

When SSL termination is required, the certificate can either be installed on the real servers (e.g. IIS) or directly on the load balancer.

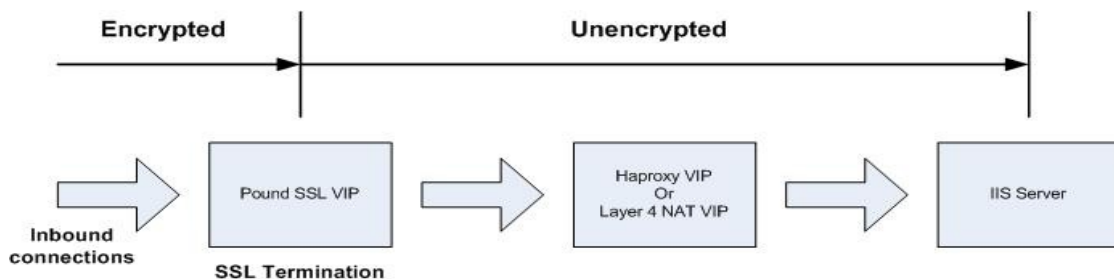
### *Certificate on the real servers*



*Using this method:*

- Data is encrypted from client to server. This provides full end-to-end data encryption as shown in the diagram below
- It's not possible to use HTTP cookie persistence since the packet is encrypted and therefore the cookie cannot be read. If persistence via the load balancer is required, IP persistence must be used

### *Certificate on the load balancer*



*Using this method:*

- Since SSL is terminated on the load balancer, data from the load balancer to the IIS servers is not encrypted as shown in the diagram above. This may or may not be an issue depending on the network structure between the load balancer and IIS servers and your security requirements
- It's possible to use HTTP cookie based persistence
- A Pound SSL Virtual Server is used to terminate SSL. The backend for this Virtual Server can be either a Layer 4 NAT mode Virtual Server or a Layer 7 HAProxy Virtual Server



DR mode cannot be used as the back-end VIP since Pound acts as a proxy, and the real servers see requests with a source IP address of the virtual server. Since the real servers believe that they own the Virtual IP (due to the loopback adapter configured to handle the ARP problem) they are therefore unable to reply to Pound.

## Creating a new certificate using a CSR

By default, when creating the SSL virtual service a self-signed certificate is used. This is ideal for testing but needs to be replaced for live deployments.

In order to obtain a valid signed certificate from a certificate authority such as Verisign or Thawte you'll need to generate a certificate request (CSR).

To customize the certificate configuration, go to *Edit Configuration > SSL Termination*, then click **[Certificate]** next to the relevant Virtual Server.

To generate a CSR, fill in the required details and click **Generate SSL Certificate Request**

Country code (C)	GB	?
State or Province (ST)	Hampshire	?
City (L)	Portsmouth	?
Organisation (O)	Loadbalancer.org	?
Organisation unit (OU)	Support	?
Domain (CN)	www.loadbalancer.org	?
Email address	support@loadbalancer.org	?

**Generate SSL Certificate Request**

Then copy the resulting Certificate Signing Request from the top pane and send this to your chosen Certificate Authority.

Certificate Signing Request	<pre>-----BEGIN CERTIFICATE REQUEST----- MIICuDCCAaACAQAwwczELMAkGA1UEBhMCVUxxDDAKBgNVBAgTA3NhZjENMA5GA1UE BxMEYXNnZjENMA5GA1UEChMEYXNkZzENMA5GA1UECjMEYXNkZzENMA5GA1UEAxME YXNkZzEaMBGCSqGSIb3DQEJARYLYXNkZzEaMBGCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDh/ait7ZjB/1K7TCUvBoByrbhuVrY0/kQtSYU5j4MX 60B3X0x4+OhJvV1SJZ8Nsjsx7m+2rfE+NP2mz/5y0rolyEuHgHT03P/gcOGp5O2N k1z3qKgvsDEpIVjCs+ALz+9arh4hsKf8CmpczFnKf1+/LCA97kkoC6zxLiKSoZ+n ab8Gd7rGfMAf95iqmgwUYUd6oqKNjjBb1AT8x10DBPGRJ+ntFhQb2KDPnYRhRYIf -----</pre>
Signed Certificate from CA	

**Upload Signed Certificate**

Once you receive your signed certificate from the CA, copy this into the lower pane and click **Upload Signed Certificate**



If you need to add intermediate certificates to the chain, this can be done at this stage by appending these certificates to the bottom of the certificate from your CA in the lower pane

## Using an existing certificate

To use an existing certificate, you must first ensure that your certificate and associated files are in PEM format. The file should contain the private key (without a password), the signed certificate issued by a Certificate Authority (CA) and also any additional validation / intermediate certificates that may be required by the CA.

### Creating a PEM File


Using a text editor such as vi or vim under Linux or Notepad under Windows create an empty file called pem.txt for example. Then copy / paste the Certificate and Private Key into the file as follows (shows truncated versions):

```
-----BEGIN CERTIFICATE-----
MIICsDCCAhmgAwIBAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxChA3BgNV
BAYTAkFVMRMwEQYDVQQIEwpTb211LVN0YXRIMSEwHwYDVQQKEWhJbnRlcm5ldCBX
kU6DJupN6U6PRI7+zcKqd8wUiY8+3CyYKHtJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBnkyHYbQTVu8MeR3ilhe2fw+qVE2pgxWYyWaGm8QwTsxQKgbxiG
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbYaAEbcSVympQJdgs6W6ajjLS
-----END CERTIFICATE-----

-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwlm3HyoVxjrymusOeIFgZiWYuaeblreCplo+iy
pSxEruhpmqmdj2tYlpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPPLAfmh88bS7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbYaAEbcSVympQJdgs6W6ajjLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HULrTqwES4Lun
-----END RSA PRIVATE KEY-----
```

Save the file, then using *Edit Configuration > Manage SSL Certificate > Upload prepared PEM file > Browse* , select this file and click **Upload PEM File**

Upload prepared PEM file

Now restart Pound using *Maintenance > Restart Services > Restart Pound*



If your master & slave are correctly configured as a clustered pair, when you upload the PEM file to the master, the file will be automatically copied over to the slave unit



It's very important to backup all of these files. This can be done via the WUI from *Maintenance > Backup & Restore > Download SSL Certificates*



If you have already generated the CSR on your Web Server, you will need to create a PEM file using the Certificate and Private Key, then upload this using the interface - see the following sections

## Adding an Intermediate certificate

Certificate authorities may require that an intermediate CA certificate is installed in your server farm. This can be done by manually pasting the intermediate CA onto the end of your signed server PEM file and then uploading it to the appliance via the upload facility.

*NB. Your current signed key is stored in /usr/local/etc/certs/<vip-name>.pem*

Select the text in the top pane and paste it into a text editor such as notepad (not Word or Wordpad):

```
-----BEGIN CERTIFICATE-----
MIICSDCCAhmgAwIBAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxChAJBgNV
BAYTAkFVMRMwEQYDVQQIEwpTb211LVN0YXRIMSEwHwYDVQQKEzhJbnRlcm5ldCBX
kU6DJupvN6U6PRI7+zcKqd8wUiY8+3CyYKHTJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBKyhYbQTvU8MeR3ilhe2fw+qVE2pgxWYyWaGm8QwTsxQKgbxiG
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbYaAEbcSVympQJdgs6W6ajjLS
-----END CERTIFICATE-----

-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwlm3HyoVxjrymusOeIFgZIWyuablrreCplo+iy
pSxEruhpmqmdj2tYlpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPALafmh88bs7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbYaAEbcSVympQJdgs6W6ajjLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HULrTqwES4Lun
-----END RSA PRIVATE KEY-----
```

Then paste the intermediate CA certificate from your provider onto the end of the file so you get something similar to, but much longer than the following shortened example:

```
-----BEGIN CERTIFICATE-----
MIICSDCCAhmgAwIBAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxChAJBgNV
BAYTAkFVMRMwEQYDVQQIEwpTb211LVN0YXRIMSEwHwYDVQQKEzhJbnRlcm5ldCBX
kU6DJupvN6U6PRI7+zcKqd8wUiY8+3CyYKHTJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBKyhYbQTvU8MeR3ilhe2fw+qVE2pgxWYyWaGm8QwTsxQKgbxiG
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbYaAEbcSVympQJdgs6W6ajjLS
-----END CERTIFICATE-----

-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwlm3HyoVxjrymusOeIFgZIWyuablrreCplo+iy
pSxEruhpmqmdj2tYlpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPALafmh88bs7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbYaAEbcSVympQJdgs6W6ajjLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HULrTqwES4Lun
-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbYaAEbcSVympQJdgs6W6ajjLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HULrTqwES4Lun
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ
-----END CERTIFICATE-----
```

Save this text file and then use the **Upload PEM file** button as to assign this certificate to your Pound virtual server. Once the file is uploaded you will need to restart Pound.

## Windows servers

A fundamental requirement of importing a certificate into Pound is that the certificate file and the private key file must be in PEM format.

Windows Server is only able to export a private key file in .pfx format. Therefore the .pfx file must be converted to PEM format. This can be done using the program 'OpenSSL'.

The conversion can be done either on a Windows server or on any UNIX-like Operating System, such as the load balancer itself.

### **Using Windows:**

OpenSSL is available as a binary package for Windows at the following location:

[http://www.slproweb.com/download/Win32OpenSSL-1\\_0\\_0d.exe](http://www.slproweb.com/download/Win32OpenSSL-1_0_0d.exe)

This should be download and installed on a PC where you'd like to run the conversion process. There are no special instructions for this. You will now have an OpenSSL directory located on your filesystem. Click **START, RUN** then type `cmd.exe`. You need to navigate to the path where you installed your OpenSSL binaries. Within this directory chdir to `bin`

To convert .PFX to .PEM

```
openssl pkcs12 -in <drive:\path\to\cert>.pfx -nodes -out <drive:\path\to\new\cert>.pem
```

To convert .CER file to .PEM format:

```
openssl x509 -in <drive:\path\to\cert>.cer -inform DER -out <drive:\path\to\cert>.pem -outform PEM
```

### **Using UNIX / Linux:**

Once OpenSSL has been installed, you can now use the below command to convert your private key into a format ZXTM can correctly decipher.

To convert .PFX to .PEM

```
openssl pkcs12 -in <path/to/exported/cert>.pfx -nodes -out <path/to/new/cert>.pem
```

To convert .CER file to .PEM format:

```
openssl x509 -in </pat/to/cert>.cer -inform DER -out </path/to/cert>.pem -outform PEM
```

This method can also be used from the Loadbalancer.org appliance console if required.

### Import a certificate exported from Windows server

For Windows, its often easiest to get the certificate working on the server first. The certificate can then be exported from Windows in .pfx format, then converted to .pem format and finally loaded into the relevant Pound Virtual Server on the load balancer. The steps are:

- 1) Once the certificate is working correctly on your Windows server, export the certificate from Windows. The format will be .pfx.
- 2) Download openssl from : [http://www.slproweb.com/download/Win32OpenSSL-1\\_0\\_0d.exe](http://www.slproweb.com/download/Win32OpenSSL-1_0_0d.exe) and install this on your PC.

- 3) Using openssl on your PC, convert the pfx file to a pem file. The command to use is:

```
openssl pkcs12 -in drive:\path to cert\cert.pfx -nodes -out drive:\path to cert\cert.pem
```

(You will be prompted for the password used to create the pfx file)

- 4) Now upload the PEM file using *Edit Configuration > SSL Termination > Certificate > Upload prepared PEM file*

- 5) Finally restart Pound (*Maintenance > Restart Pound-SSL*)

### Converting an encrypted private key to an unencrypted key

If a password has been included in the private key, this should be removed before it is used with your pem file. This can be done using the following method:

```
openssl rsa -in server.key -out server.key.unencrypted
```

(This can be done either on the load balancer or another machine with openssl installed)

### Limiting ciphers

To limit the Ciphers that Pound will respond to, simply enter the cipher string in the Ciphers field. For example, to limit to SSL v3, enter SSLv3 and click update. Multiple Ciphers can be entered separated by commas.

#### **EDIT CONFIGURATION > SSL TERMINATION VIRTUAL SERVERS**

Label	SSL	?
Virtual Server IP address	192.168.2.60	?
Virtual Server Port	443	?
Backend Virtual Server IP Address	192.168.2.60	?
Backend Virtual Server Port	80	?
Ciphers to use		?

Update

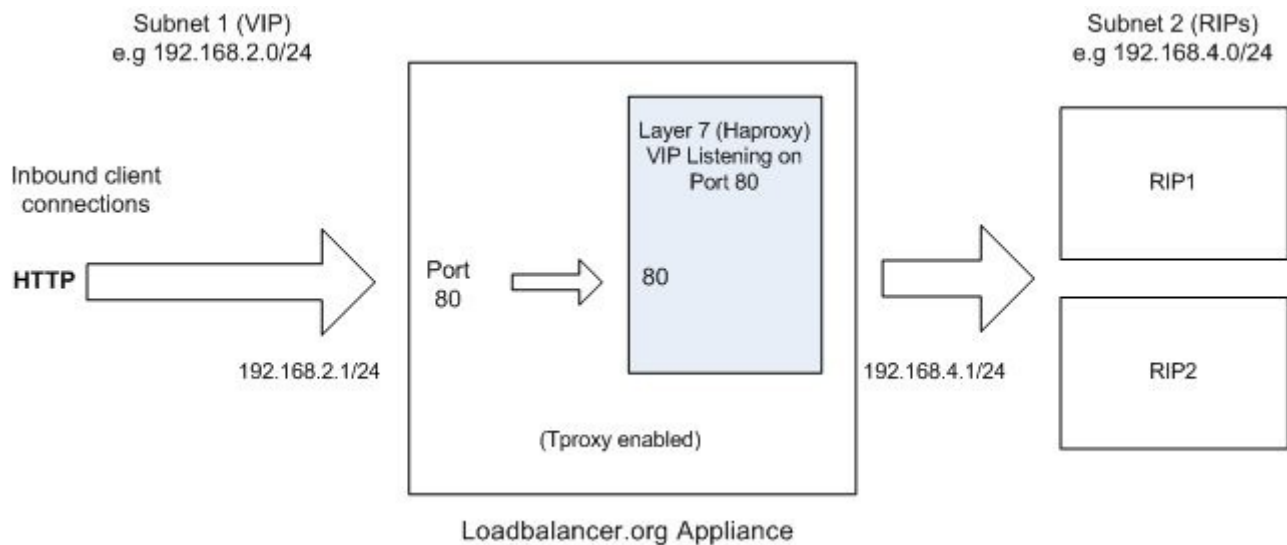


## Using Tproxy

Tproxy can be used with Haproxy and Pound to maintain the actual source IP address of the client rather than the IP address of the proxy itself. When enabling Tproxy, its important to be aware of the topology requirements for Tproxy to work correctly. This is explained in the examples below.

### Example 1 – Layer 7 (Haproxy) VIP with Tproxy enabled:

In this example, a layer 7 VIP is configured in the normal way and Tproxy is enabled. The setup is shown in the following example.

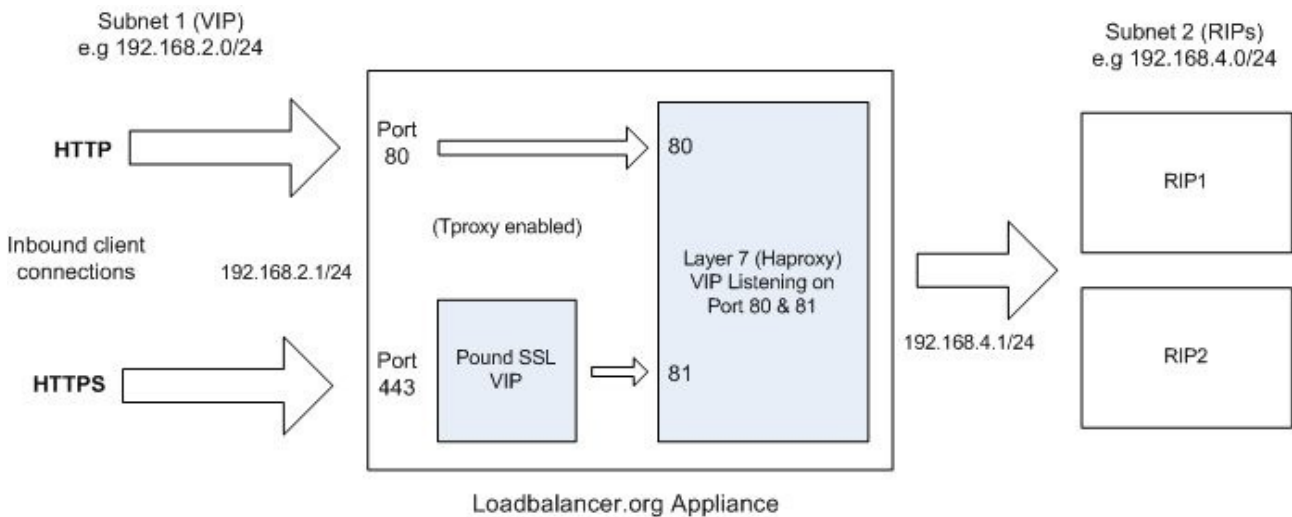


### NOTES:

- The RIPs must be on a different subnet to the VIP - this can be achieved by using 2 IP addresses assigned to a single interface, or two separate interfaces (eth0 & eth1)
- Tproxy must be enabled using the WUI : open *Edit Configuration > Layer 7 – Advanced Configuration* and set **Transparent Proxy** to 'On'
- On the real servers, the default gateway must be configured to be an IP address on the load balancer. Its best to add an additional floating IP for this to allow failover to the slave

### Example 2 – SSL (Pound) Termination with Tproxy enabled:

In this example Pound is also used to terminate SSL. Here, Pound passes the decrypted traffic to a layer 7 backend VIP where the real servers are configured. This setup is shown in the following diagram.



### NOTES:

- The RIPs must be on a different subnet to the VIP - this can be achieved by using 2 IP addresses assigned to a single interface, or two separate interfaces (eth0 & eth1)
- Configure the Layer 7 VIP to listen on 2 ports - e.g. 80 & 81, then use port 81 for the Pound backend and port 80 for client connections as shown above
- Tproxy must be enabled using the WUI : open *Edit Configuration* > *SSL – Advanced Configuration* and set **Transparent Proxy** to 'On' (this will also automatically enable Tproxy for Haproxy)
- On the real servers, the default gateway must be configured to be an IP address on the load balancer. It's best to add an additional floating IP for this to allow failover to the slave

## Health Monitoring

The loadbalancer.org appliance supports both real server (back-end) and load balancer health checks.

### *Load balancer health (clustered pair)*

When a clustered pair is deployed rather than a single appliance, the load balancers are configured by default to use a serial connection to check the health of each other. This permits failover to the slave unit if the master unit fails. Multiple checks can be configured between the appliances using the serial cable and network cables, as well as checks to a common node such as the default gateway. This allows a number of checks to be configured to ensure that failover only occurs when needed and 'split brain' (i.e. master and slave are both active) scenarios are avoided.

#### Heartbeat communication method

##### EDIT CONFIGURATION > MODIFY HEARTBEAT CONFIGURATION

Serial	<input checked="" type="checkbox"/>	<a href="#">?</a>
Unicast	<input type="checkbox"/>	<a href="#">?</a>
Broadcast	None ▾	<a href="#">?</a>
UDP Port for broadcast & unicast	6689	<a href="#">?</a>
Keepalive	3	<a href="#">?</a>
Deadtime	10	<a href="#">?</a>
Warntime	5	<a href="#">?</a>
Ping node		<a href="#">?</a>
Automatic Fail-back	<input checked="" type="checkbox"/>	<a href="#">?</a>

Modify Heartbeat configuration

*N.B. The screen shot above shows the configuration screen for the hardware appliance, for the VA the serial option is not available.*

#### Serial cable

This method requires a null modem cable (1 cable is supplied with each appliance) to be connected between the two load balancers in the cluster. This enables heartbeat checks to utilize the serial port (ttys0 / ttys1). This is the only method which is active by default, other methods must be enabled manually.



When the wizard is used to configure a VMware based clustered pair, heartbeat is automatically configured to use the network (ucast) for heartbeat.

### Unicast (ucast)

This method of heartbeat communication uses unicast UDP between master and slave, with a destination port given by the UDP Port for broadcast & unicast parameter. When unicast is enabled, the load balancer determines the correct interface and IP addresses to use based upon the configured slave IP address. Please ensure that the correct slave IP has been entered on the DNS & Hostname page before enabling unicast. Unicast is the preferred communication method if serial cannot be used.

### Broadcast (bcast)

This method of heartbeat communication uses broadcast UDP between master and slave, with a destination port given by the UDP Port for broadcast & unicast parameter. Care must be taken when using broadcast on multiple pairs of load balancers in the same network. Each high-availability pair must operate on a different UDP port if they are not to interfere with each other. If heartbeat communication over the network is required, it is recommended that unicast be used in preference to broadcast.

### **Ping node**

Specify a mutually accessible IP address to test network availability. A good ping node to specify is the IP address of a router that both the master and slave can access (e.g. the default gateway). If the active appliance loses access to the ping node, but the passive appliance still has access, then a failover will occur. However, if both nodes lose access failure will not occur.

### **Auto-failback**

When the master returns to service after a failure do we transfer resources back to it. Sometimes it is useful to always fall back to the master. If you prefer to manually control this process, un-check this option.



If the master unit is configured first and a slave unit is added later, heartbeat will need to be restarted on both units for synchronization to occur. This can be done using : *Maintenance > Restart Services*

## Real server health

Real server health checking is provided by Ldirectord at layer 4. This is integrated into Loadbalancer.org appliances and allows a full range of options to check that real servers are operational, and if not what steps to take.

### Layer 4

*Edit Configuration > Virtual Servers > Modify*

Check Type	Connect to port	?
Service to check	HTTP	?
Check Port		?
Check Command		?
Virtual Host		?
Login		?
Password		?
Protocol	TCP	?
Granularity	255.255.255.255	?
Request to send	check.txt	?
Response expected	OK	?
Email Alerts		?

### Check types

**Negotiate connection** – Sends a request and looks for a specific response (see service to check below)

**Connect to port** - Just do a simple connect to the specified port/service & verify that its able to accept a connection

**Ping server** – Sends an ICMP echo request packet to the real server

**External check** - Use a custom file for the health check. Specify the filepath in the 'Check Command' field.

**No checks, always Off** - All real servers are off

**No checks, always On** - All real servers are on (no checking)

**5 Connects, 1 Negotiate** - Do 5 connect checks and then 1 negotiate

**10 Connects, 1 Negotiate** - Do 10 connect checks and then 1 negotiate

### Service to check

If negotiate is selected as the check type, the following methods are available:

**HTTP** – use HTTP as the negotiate protocol (also requires filename, path + text expected)

**HTTPS** – use HTTPS as the negotiate protocol (also requires filename, path + text expected)

**HTTP Proxy** – Use an http proxy check

**FTP** – use FTP as the negotiate protocol (also requires login/password, filename in the default folder)

**IMAP (IPv4 only)** – use IMAP as the negotiate protocol (requires login/password)

**IMAPS (IPv4 only)** - use IMAPs as the negotiate protocol (requires login/password)

**POP** – use POP as the negotiate protocol (also requires login/password)

**POPS** - use POPs as the negotiate protocol (also requires login/password)

**LDAP (IPv4 only)**– use LDAP as the negotiate protocol (also requires username/password)

**SMTP** – use SMTP as the negotiate protocol

**NNTP (IPv4 only)** – use NNTP as the negotiate protocol

**DNS** – use DNS as the negotiate protocol

**MySQL (IPv4 only)** – use MySQL as the negotiate protocol (also requires username/password)

**SIP** – use SIP as the negotiate protocol (also requires username/password)

**Simple TCP** – Sends a request string to the server and checks the response

**RADIUS (IPv4 only)** - use RADIUS as the negotiate protocol (also requires username/password)

**none**

### Check port

This can be used if the port to check is non standard, e.g., the service to check is https, but the port used is 4443 instead of the standard 443.

### Check command

The custom check script, used with the external check type. The script should be placed in /var/lib/loadbalancer.org, and given world read and execute permissions.

### Virtual host

If the real server will only respond to a URL or 'virtualhost' rather than an ip address. You can specify the virtual host to request here.

### Login

The login name to use with negotiate checks where authentication is required.

### Password

The password to use with negotiate checks where authentication is required.

### Request to send

This is used with negotiate checks and specifies the request to send to the server. The use of this parameter varies with the protocol selected in Service to Check. With protocols such as HTTP and FTP, this should be the object to request from the server. Bare filenames will be requested from the web or FTP root. With DNS, this should be either a name to look up in an A record, or an IP address to look up in a PTR record. With databases, this should be an SQL query. With LDAP, this should be the search base for the query. The load balancer will perform an (ObjectClass=\*) search relative to this base. With Simple TCP, this should be a string to send verbatim to the server.

### Response expected

This is the response that must be received for the negotiate to be a success. The negotiate check succeeds if the specified text (response) is found anywhere in the response from the web server when the file specified in the File to Check field is requested.











For example, a file called 'check.txt' could be placed in the default folder of the web server, this text file could just have the text **OK** in the file, then when the negotiate check runs, it would look for a file called 'check.txt' containing **OK**. If found, the test would succeed, if not found it would fail and no new sessions will be sent to that server.

### Email alerts

Specify the email alert address. This can also be configured at a global level to apply to all Layer 4 Virtual Servers in the WUI using *Edit Configuration > Layer 4 Advanced Configuration*.

### Additional health check settings

**EDIT CONFIGURATION > ADVANCED CONFIGURATION**

Layer 4		
Check Interval	<input type="text" value="6"/>	
Check Timeout	<input type="text" value="3"/>	
Negotiate Timeout	<input type="text" value="5"/>	
Failure Count	<input type="text" value="1"/>	
Quiescent	<input type="text" value="no"/>	
Email Alerts	<input type="text"/>	
Auto NAT	<input type="text" value="off"/>	
Multi-threaded	<input type="text" value="yes"/>	
Fallback	<input type="text" value="yes"/>	
Disable Write	<input type="text" value="off"/>	

Update

### Check interval

Layer 4 (Ldirectord) health check interval in seconds. If this setting is too low, you may induce un-expected real server downtime. For slower servers, this may need to be increased.

### Check timeout

Layer 4 (Ldirectord) health check timeout in seconds. If this setting is too low, you may induce un-expected real server downtime. For slower servers, this may need to be increased.

### Negotiate timeout

Layer 4 (Ldirectord) negotiate health check timeout in seconds. The negotiate checks may take longer to process as they involve more server side processing than a simple TCP socket connect check. If this setting is too low, you may induce unexpected real server downtime. For slower servers, this may need to be increased.

### Failure count

Layer 4 (Ldirectord) number of times a check has to fail before taking server offline. The time to detect a failure and take down a server will be (check interval + check timeout ) \* failure count.

### Quiescent

When Quiescent is yes, on a health check failure the real server is not removed from the load balancing table, but the weight is set to 0. Persistent connections will continue to be routed to the failed server, but no new connections will be accepted. When Quiescent is no, the server is completely removed from the load balancing table on a health check failure. Persistent connections will be broken and sent to a different real server.

Quiescent only applies to health checks - it has no effect on taking real servers offline in System Overview. To manually force a real server to be removed from the table, set Quiescent to no and arrange for the server to fail its health check. This may be done, for example, by shutting down the daemon or service, changing the negotiate check value, or shutting down the server.

### Email alerts

This is the default global setting for email alerts and is used for all Layer 4 Virtual Servers if no other address is specified at the individual VIP level.

### Multi-threaded

Perform health checks with multiple threads. Using multiple-threads for health checks will increase performance when you have a large number of virtual servers.

### Fallback

Local Fallback server on / off switch . Configure whether the local (nginx) fallback server is active or not, sometimes you may want the local fallback server switched off so that it doesn't change the SNMP results table when activated. You may also want to disable it for security purposes.



## Layer 7

*Edit Configuration > Virtual Servers (HA Proxy) > Modify*

Check Port	<input type="text"/>	?
Request to send	<input type="text"/>	?
Response expected	<input type="text"/>	?

### Check port

Specify a different port for health checks. If specified this setting overrides the default checkport, useful when you are balancing multiple ports.

### Request to send

Specify a specific file for the health check. Open the specified file and check for the response expected, useful for checking a server sided script to check the health of the back-end application.

### Response expected

The content expected for a valid health check on the specified file. The response expected can be any valid regular expression statement. For example, if the server has a virtual directory configured called /customers, the required setup would be as follows:

Request to send: /customers/

Response expected: welcome

For this to work and the health-check to pass, the text 'welcome' (case sensitive) would need to exist on the default page of the /customers virtual directory.

### Additional health check settings

Interval	<input type="text" value="2000"/>	?
Rise	<input type="text" value="2"/>	?
Fall	<input type="text" value="3"/>	?

### Interval

Interval between health checks. This is the time interval between real server health checks in milliseconds.

### Rise

Number of health checks to Rise. The number of positive health checks required before re-activating a real server.

### Fall

Number of health checks to Fall. The number of negative health checks required before de-activating a real server.

## Fallback server settings

This section allows you to view and modify the local holding page on the load balancer. If you have a master and slave load balancer then you must change this on both servers. The fallback server on the load balancer is an implementation of NGINX.

### Layer 4

The fallback page is displayed when all real servers fail. The fallback page is NOT displayed when servers are taken offline manually via the WUI.

At layer 4, to cause the fallback page to be displayed when real servers are taken offline, you need to force all real servers to fail their health check by for example disabling the relevant service on each real server.

### Layer 7

For layer 7 VIPs the fallback page is displayed when all real servers are unavailable *and also* when all are taken offline via the WUI. The page can be hosted on the load balancer or on an external server. Set the Fallback Server option of the VIP accordingly.

**MAINTENANCE > FALLBACK PAGE**

```
<html>
<head>
<title>The page is temporarily unavailable</title>
<style>
body { font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body bgcolor="white" text="black">
<table width="100%" height="100%">
<tr>
<td align="center" valign="middle">
The page you are looking for is temporarily unavailable.<br/>
Please try again later.<br/>
(port reminder 9080)
</td>
</tr>
```

Update

- The local fallback server is an NGINX instance by default on port 9081
- Use the command **lb2ports** for layer 4 environments to put NGINX on both port 80 & 9081
- Use the command **lb1port** to move it back to port 9081 only, for compatibility with HAProxy on port 80.
- You can use any valid HTML for the default page, simply cut and paste from your favorite editor.

### Holding page on the load balancer

For layer 4 DR & NAT modes, to ensure NGINX is listening on the correct port, run the following command at the console:

```
lb2ports
```

(this can also be run via the WUI if preferred using : *Edit Configuration > Execute a shell script*)

For layer 7, to ensure NGINX is listening on the correct port (9081 only), run the following command at the console:

```
Lb1port
```

(this can also be run via the WUI if preferred using : *Edit Configuration > Execute a shell script*)



If you are using the load balancer for your holding page and your web servers are offline then the local NGINX server is exposed to hacking attempts, if you are concerned about this you can change the fallback server to one of your internal servers.

### Holding page on a dedicated server

The fallback server can also be any appropriate external server. To use this, set the Fallback Server and Fallback Server Ports field of the VIP to the IP address & port of the fallback server, e.g. 192.168.2.10 & 80.



For DR mode the fallback server must be listening on the same port as the VIP. Also, don't forget to solve the ARP problem for the dedicated fallback server if used



For NAT mode don't forget to set the default gateway of the fallback server to the internal IP of the load balancer or when you have 2 appliances in a cluster, to a floating IP

## Advanced firewall considerations

Understanding what you are trying to achieve and how to go about it in the *rc.firewall* script may look a bit scary but it uses Linux netfilter which is an excellent transferable skill to learn.

If you want a quick and simple firewall script then use the firewall lock down wizard. However be very wary of locking yourself out of the system if you are accessing the unit remotely.

If you want to set up a complex NAT solution, or use the Loadbalancer.org appliances as bastion hosts then here are a couple of pointers:

1. All virtual server connections are dealt with on the INPUT chain not the FORWARD chain
2. The SNAT & DNAT is handled automatically for all the Virtual/Real load balanced services
3. HTTP, HTTPS & SSH are by default OPEN on the INPUT chain i.e. If you have a public IP for your VIP someone can use HTTP to get to the local Apache installation on the load balancer, unless you:
  - a) Set up a real server group for HTTP (and HTTPS & SSH)
  - b) Firewall the appliance! (*either using your firewall or the rc.firewall script or both*)
4. You can use the standard Linux filters against spoofing attacks and syn floods
5. LVS has built in DOS attack filters that can be implemented
6. Plenty of extra information is available on the Internet relating to Netfilter and LVS (Linux Virtual Server)

### *Firewall marks (Layer 4)*

Using firewall marks enables multiple ports to be combined into a single virtual service. A common use of this feature is to aggregate port 80 (http) and port 443 (https) so that when a client fills their shopping cart on via http, then move to https to give their credit card information, they will stay on the same real server.



New to version 7.x is the auto-configuration of firewall marks for certain requirements. Previous versions of the software required that this be configured manually.

### Firewall Marks – Auto Configuration

For example, to configure an HTTP/HTTPS virtual server, simply specify port 80 & 443 separated by a comma in the 'Virtual Server Ports' field. This will automatically configure the load balancer for firewall marks.

Label	<input type="text" value="HTTP_Cluster"/>	?
Virtual Server IP address	<input type="text" value="192.168.50.1"/>	?
Virtual Server Ports	<input type="text" value="80,443"/>	?

**Note:** If creating a layer 4 NAT mode VIP with multiple ports, specify '0' as the port for the real servers. Also make sure that for both DR & NAT mode VIPs the check-port is set to a valid port - otherwise health checks will fail and the servers will be marked as down.

## Firewall Marks – Manual Configuration

As with previous versions, firewall marks can also be configured manually if required. The basic concept is to create a firewall rule that matches incoming packets to a particular IP address and port, and mark them with an arbitrary integer. A Virtual Server is then configured or modified, specifying the firewall mark instead of the IP address.

### Step 1 – modify the firewall script

The *Maintenance > Firewall Script* page in the WUI includes some examples under the "FIREWALL MARKS" section as shown below. The example firewall mark shown can be uncommented and edited to suit your requirements.

```
MAINTENANCE > FIREWALL SCRIPT
##### FIREWALL MARKS #####

# Now setup any Firewall marks that are required
# Firewall marks allows you to associate multiple ports with one VIP
# This is useful if you need to keep HTTP & HTTPS persistent

# This example marks HTTP & HTTPS connections only

#VIP1="10.0.0.66"
# iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1
# iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 443 -j MARK --set-mark 1

# If you then add a virtual service with an address of '1' rather
# than 'IP:port' it will balance HTTPS & HTTP,
# this would usually be set as persistent...
```

Update

e.g. to mark incoming packets destined to both 192.168.2.165:80 and 192.168.2.165:443 with the same value:

change the default script from:

```
#VIP1="10.0.0.66"
# iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1
# iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 443 -j MARK --set-mark 1
```

to:

```
VIP1="192.168.2.165"
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 443 -j MARK --set-mark 1
```

click **Update**

## Step 2 – create / modify the virtual server

1. Create a new Virtual Server or edit an existing Virtual Server. The IP address must be set to be the same integer value used for the mark in step 1 (in this example a “1”)

### EDIT CONFIGURATION > LOGICAL LAYER 4 CONFIGURATION

Label	VIP2	?
Virtual Server IP address	192.168.70.208	?
Virtual Server Ports	80	?

change this to:

### EDIT CONFIGURATION > LOGICAL LAYER 4 CONFIGURATION

Label	VIP2	?
Virtual Server IP address	1	?
Virtual Server Ports		?

2. Clear the port field
3. Make sure the Check Port field is set to a valid port for the VIP , e.g. 80
4. Make sure the 'Protocol' is set to *Firewall Marks*
5. Set 'Persistence' to Yes
6. Click **Update**
7. If the Firewall Mark VIP is being created from scratch rather than by modifying an existing VIP, then also add a floating IP that corresponds to the IP address used in step 1 using *Edit Configuration > Floating IPs* . If the Firewall Mark is setup by modifying an existing VIP, the floating IP will already exist

All the other fields can be left as default values and real servers can be associated with the VIP in the normal way. With a firewall mark Virtual Server, the load balancer forwards traffic to the selected real server without changing the destination port. So, incoming traffic to port 80 on the Virtual IP will be forwarded to port 80 on one of the real servers. Likewise, incoming traffic to port 443 will be forwarded to port 443 on the same real server.

You can only have one health check port assigned, so even if you are grouping port 80 and 443 traffic together you would normally run health checks on port 80.

You can also state a range of ports using ':' for the --dport option. For example to specify destination ports from 1024 to 1090, you can use:

--dport 1024:1090

To specify all destination ports of 1024 and above, you can use:

--dport 1024: (the end of the range is assumed if you omit that value)

## Layer 4 Persistence Considerations

### *Persistence state table replication*

If you want the current persistent connection table to work when the master load balancer swaps over to the slave then you can start the synchronization daemons on each load balancer to replicate the data in real time.

First login to lbmaster using SSH or the console, then as root run the following command:

```
ipvsadm --start-daemon master  
ipvsadm --start-daemon backup
```

Then login to lbslave using SSH or the console, then as root run the following command:

```
ipvsadm --start-daemon master  
ipvsadm --start-daemon backup
```

After a few seconds you can confirm that it is working by seeing the output from:

```
ipvsadm -Lnc
```

This should give the same output as running the same command on lbmaster i.e. The state table is being replicated.

*NB. This is the same command that the 'status' report is based on.*

*NB. Obviously you should put these commands in the rc.firewall script to ensure that the sync daemons are started on each re-boot.*



Setting this option can generate a high level of connection state synchronization data between the master and slave load balancers.

### *Server maintenance when using persistence*

A protocol with a long session & persistence enabled such as Terminal Server RDP maintenance can become problematic because clients that disconnect and re-connect will still go to the same server for the length of the persistence timeout. This behavior has already been modified on the Loadbalancer.org appliances (from v6.5) so that when a client disconnects the persistence template is cleared forcing them to re-connect to a different server.

In the unlikely event that you wish to disable this feature globally use the following commands from the console:

```
echo 0 > /proc/sys/net/ipv4/vs/expire_quiescent_template  
echo 0 > /proc/sys/net/ipv4/vs/expire_nodest_conn
```

*NB. This can be made a permanent setting on both load balancers by adding it to the /etc/sysctl.conf file.*

If you are using negotiate checks you may also want to use the quiescent=no global option to ensure that if a server fails a negotiate check but is still technically working the connections are forced to fail over rather than being drained gradually.

## SNMP reporting

Native SNMP support can be enabled on the appliance. This is a simple case of enabling the service:

```
service snmpd start
chkconfig snmpd on
```

*('chkconfig snmpd on' forces snmp to start on appliance reboot)*

The dedicated load balancing mib oid is: 1.3.6.1.4.1.8225.4711

### *SNMP for layer 4 based services*

You can test if everything works by invoking:

```
shell> snmpwalk -c public -v 2c -m LVS-MIB localhost 1.3.6.1.4.1.8225.4711
LVS-MIB::lvsVersion.0 = STRING: "1.2.0"
LVS-MIB::lvsNumServices.0 = INTEGER: 2
LVS-MIB::lvsHashTableSize.0 = INTEGER: 4096
LVS-MIB::lvsTcpTimeOut.0 = INTEGER: 900
LVS-MIB::lvsTcpFinTimeOut.0 = INTEGER: 120
LVS-MIB::lvsUdpTimeOut.0 = INTEGER: 300
LVS-MIB::lvsDaemonState.0 = INTEGER: none(0)
...
etc.
```

Note: LVS-MIB.txt can be downloaded from : <http://www.loadbalancer.org/download/SNMP/>

You can also use all the usual MIB2 counters and gauges such as network and CPU etc.

### *SNMP for layer 7 based services*

Front end stats are returned by invoking:

```
[root@lbmaster ~]# snmpwalk -c public -v2c 127.0.0.1 1.3.6.1.4.1.29385.106.1.0
SNMPv2-SMI::enterprises.29385.106.1.0.0.1.0 = STRING: "stats"
SNMPv2-SMI::enterprises.29385.106.1.0.1.1.0 = STRING: "FRONTEND"
SNMPv2-SMI::enterprises.29385.106.1.0.2.1.0 = ""
SNMPv2-SMI::enterprises.29385.106.1.0.3.1.0 = ""
SNMPv2-SMI::enterprises.29385.106.1.0.4.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.0.5.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.0.6.1.0 = STRING: "2000"
...
etc.
```

Back end stats are returned by invoking:

```
[root@lbmaster ~]# snmpwalk -c public -v2c 127.0.0.1 1.3.6.1.4.1.29385.106.1.1
SNMPv2-SMI::enterprises.29385.106.1.1.0.1.0 = STRING: "stats"
SNMPv2-SMI::enterprises.29385.106.1.1.1.1.0 = STRING: "BACKEND"
SNMPv2-SMI::enterprises.29385.106.1.1.2.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.1.3.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.1.4.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.1.5.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.1.6.1.0 = STRING: "2000"
SNMPv2-SMI::enterprises.29385.106.1.1.7.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.1.8.1.0 = STRING: "0"
...
etc.
```



## Feedback agents

The load balancer can modify the weight (amount of traffic) of each server by gathering data from either a custom agent or an HTTP server. Just set the virtual servers feedback method to agent or http as required. A telnet to port 3333 on a real server with the agent installed will return the current CPU idle as an integer 0-100

The load balancer expects a 0-99 integer response from the agent usually relating to the CPU idle i.e. a response of 92 would imply that the real servers CPU is 92% idle. The load balancer will then use the formula  $(92/10 * \text{requested\_weight})$  to find the new optimized weight. Using this method an idle real server will get 10 times as many new connections as an overloaded server.

*NB. The feedback agent will never offline a server only the standard health check can take a server offline.*

## *Installing the Windows agent*

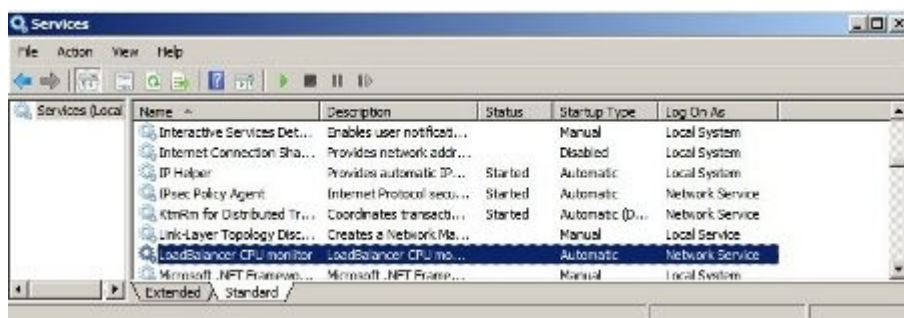
Download the agent from:

<http://www.loadbalancer.org/download/agent/Windows/LBCPUMonInstallation.msi>

Run the installer and follow the wizard to install the service correctly on each real server.



Once the service is installed you will need to start the service:



## *Installing the Linux/Unix agent*

Download the agent from <http://www.loadbalancer.org/download/agent/>

```
apt-get install xinetd (if not already installed)

Insert this line into /etc/services
lb-feedback      3333/tcp                                # Loadbalancer.org feedback daemon

Then:
cp lb-feedback.sh /usr/bin/lb-feedback.sh
chmod +x /usr/bin/lb-feedback.sh
cp lb-feedback /etc/xinetd.d/lb-feedback
chmod 644 /etc/xinetd.d/lb-feedback

/etc/init.d/xinetd restart

Testing:
telnet 127.0.0.1 3333

Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
95
Connection closed by foreign host.
Connection closed by foreign host.
```

## *Custom HTTP agent*

You can use any HTTP server responding on port 3333 to give feedback information to the load balancer. The format of this information must be an integer number of 0-100 without any header information. Using this method you can generate a custom response based on your applications requirements i.e. a mixture of memory usage, IO, CPU etc.

## Changing the local date, time & time zone

You can change the time & time zone from the web interface using:

*Edit Configuration > System date & time*

To set the date and time at the console use the following commands:

```
date --set 1998-11-02 (yyyy-mm-dd)
date --set 21:08:00 (hh:mm:ss)
```

To set the hardware clock to the system time do a:

```
hwclock --systohc
```

## *NTP configuration*

If the load balancer has ntp access to the Internet you can do a:

```
ntpdate time.nist.gov
```

*NB. This is already in the root cron job in /etc/crontab.*

The load balancers local clock is updated once a day using ntp, this requires that your default gateway and DNS are set correctly.

Timezone can be Coordinated Universal Time (UTC) or GMT based like GMT, GMT + 1 hour, GMT - 1 hour, and so on. Please consider that the GMT+/-X format as it is returned by the system differs from the GMT +/- X hours format. The GMT+/-X based statement follows the POSIX standard which means that GMT+X is X hours west of Greenwich. GMT-X means X hours east of Greenwich. So GMT+X means GMT - X hours and vice-verse.



When using a clustered pair (i.e. master & slave) manual time & time zone changes on the master will not be automatically replicated to the slave, therefore the slave must also be set manually

## Restoring Manufacturer's Settings

The load balancer settings can be reset to factory default values in two ways:

### *From the console*

```
lbrestore
```

### *From the WUI*

*Maintenance > Backup & Restore > Restore Manufacturer's Defaults*

Running this will remove all custom configuration from the load balancer. All VIPs and RIPs will be removed and the IP address configured for eth0 will be set to 192.168.2.21 provided that no other device has this address, if it does, then the current IP address will remain.

## Force Master/Slave Take-Over In A Clustered Pair

### *Force the slave to become active & master passive*

*On the slave:*

```
/usr/local/sbin/hb_takeover.php all
```

### *Force the master to become active & slave passive*

*On the master:*

```
/usr/local/sbin/hb_takeover.php all
```

NOTE: these commands can either be run on the console, at a terminal session or via the WUI using : *Edit Configuration > Execute Shell Command*

## Active / Active load balancer configuration (Layer 7 - Haproxy)

Normally load balancer clustered pairs are deployed in an active / passive configuration. In this mode only one device hosts the Virtual Services (VIPs) at any one time. However, it is possible to configure the cluster so that the VIPs are shared between both devices at the same time. For example, if there are a total of 4 VIPs, 2 could be configured to run on the master and 2 could be configured to run on the slave.

To achieve this, the heartbeat configuration file `/etc/ha.d/haresources` must be edited on the master, then copied over to the slave.

An example of a typical active / passive heartbeat configuration file:

```
lbmaster 192.168.36.34 192.168.36.35 192.16.36.36 192.168.36.37 ldirectord haproxy pound
```

To change this to an active / active configuration with 2 VIPs hosted on each load balancer, on the Master load balancer the configuration file would be changed to:

```
lbmaster 192.168.36.34 192.168.36.35 ldirectord haproxy pound
```

```
lbslave 192.168.36.36 192.168.36.37 ldirectord haproxy pound
```

In this file, the format is :

<preferred device> <ip address> <resources>

Here you can see that node `lbmaster` will have the preferred VIPs `192.168.36.34` & `192.168.36.35` and use `haproxy`, `ldirectord` (layer 4) and `pound` (SSL termination) for its load balanced services, whereas the node `lbslave` will host the preferred VIPs `192.168.36.36` & `192.168.36.37` and associated services.

If one of the nodes becomes inactive, its VIPs are transferred to the active node, until such time as the inactive node becomes active again.

The configuration file would also need to be copied over to the slave by running the following command on the master:

```
scp /etc/ha.d/haresources root@lbslave:/etc/ha.d/
```

A configuration change is also needed to make sure that Layer 7 services are able to start up when they are not able to bind to a local VIP. This is done by executing the following command (on both nodes):

```
echo 1 >/proc/sys/net/ipv4/ip_nonlocal_bind
```

If you wish to make this change permanent then the following line should be added to `/etc/sysctl.conf`

```
# Allow binding to non local addresses
```

```
net.ipv4.ip_nonlocal_bind = 1
```

## Application Specific Settings

### *FTP*

FTP is a multi-port service in both active and passive modes:

**active** 20,21

**passive** 21,high\_port

Most firewalls handle this insecure protocol by stateful inspection of the traffic in order to open up the required data port on demand. LVS has a built in helper module (that loads on demand) in order to handle the correct port translation when in MASQ/NAT mode. Therefore, if you set up a Virtual Server on port 21 in MASQ/NAT configuration it should work without a hitch.

However, in DR mode the load balancer cannot see the return packets. One of the simplest ways of dealing with this is to allow your real server to have outgoing FTP access for return traffic to the client from it's RIP and configure only the incoming traffic on the load balancer. So set up a VIP on port 21 for the incoming traffic and allow the server to do the rest of the communication directly with the client. *NB. Your firewall will need to allow FTP connections to all the RIPs as well as the VIP.*

The second direct routing method is to effectively open up all ports and group them together to allow the connections to always talk to the same server. This is best done with a Firewall Mark:

```
# This example marks groups the active FTP ports
VIP1="192.168.0.66"
# First two rule are for Active connections
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 21 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 20 -j MARK --set-mark 1
# Third additional rule for passive
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 1024: -j MARK --set-mark 1
```

*NB. Your firewall will either need the same rules or preferably stateful inspection for FTP access to the VIP.*



DR mode requires the use of firewall marks for both passive and active FTP. NAT mode works in both FTP modes without any additional firewall marks.



NAT mode works in both FTP modes without any additional firewall marks. If an alternative port to the standard port 21 is required, then follow the steps in the section below.

### Changing the FTP port in NAT mode

This procedure details how to configure the load balancer to respond to FTP on port 2180 rather than the standard port 21 (this only applies when using NAT mode).

Edit the modprobe.conf configuration file:

```
nano /etc/modprobe.conf
```

At the end of the file, on a new line, add the following:

```
options ip_vs_ftp ports=2180
```

now save the file (Ctrl-O, Enter) and exit (Ctrl-X).

To apply the changes to the running system, the ftp module will need to be reloaded. This will only need to be done to test the change - on future system reboots, the option will be applied automatically. On the command line, enter:

```
modprobe -r ip_vs_ftp
```

Check that the module has been unloaded successfully:

```
lsmod | grep ip_vs_ftp
```

*there should be no output from the command above.*

Then load the module again:

```
modprobe ip_vs_ftp
```

Repeat the check:

```
lsmod | grep ip_vs_ftp
```

*This time, you should see output similar to the following:*

```
ip_vs_ftp      6297      0
ip_vs          105965    4 ip_vs_ftp
```

The system is now configured to respond to FTP on port 2180 instead of port 21.

### **FTP negotiate health check**

You can modify the virtual server so that rather than doing a simple socket connect check, it will actually attempt to log into the FTP server and read a file for a specific response:

Check Type	negotiate	?
Service to check	ftp	?
Check Port		?
Check Command		?
Virtual Host		?
Login	username	?
Password	password	?
Protocol	tcp	?
Granularity	255.255.255.255	?
File to check	check.txt	?
Response expected	OK	?

- Change the *check type* to negotiate

- Make sure the *service to check* is FTP
- Specify a *login* and *password*
- Specify the *file to check* (defaults to the root directory)
- The file is parsed for the *Response expected* that you specify

### **FTP recommended persistence settings**

When you start using multiple FTP servers in a cluster you need to be aware of the effects of a client switching server. For sites that are download only, you generally don't need any special settings on the load balancer as the connection will usually stay on the same server for the length of the connection. You may however wish to force persistence to something sensible like 15mins (If you go higher remember to change the global TCP timeouts).

If you are using the FTP servers for upload it is recommended to use a single FTP server for uploads and then replicate the data to the read only cluster for downloads (or use a clustered file system). For upload it is especially important to use persistence.

Automatically resuming a broken download is no problem even if you switch servers in a cluster on re-connect. This is because the FTP resume functionality is client based and does not need any server session information.

### **Limiting passive FTP ports**

To reduce the number of ports that the load balancer marks, the following command could be used instead of the command mentioned earlier:

```
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 50000:50100 -j MARK --set-mark 1
```

This would only allow ports 50000-50100.

This can then be limited in the same way on the real server's firewall and also on the FTP server so that passive connect ports proposed by the ftp server are in this range. The way to limit the passive port ranges on a range of typical systems is shown below:

#### For Linux

*in vsftpd, the following line can be added to the vsftpd.conf file to limit the port range:*

```
pasv_max_port - max is 65535
pasv_min_port - min is 1024
```

*in proftpd, the following line can be added to the proftpd.conf file to limit the port range:*

```
PassivePorts 50000 - 50100
```

*in pureftpd, the following startup switch can be used:*

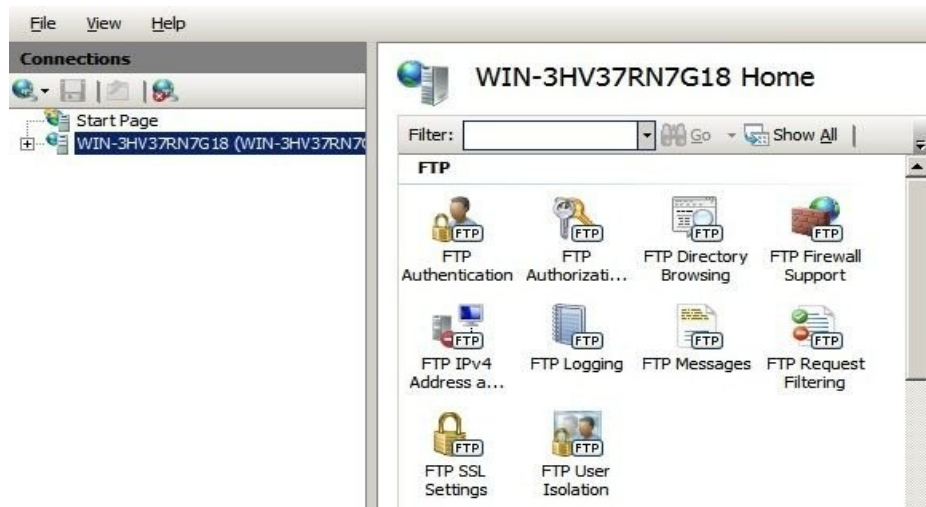
```
-p --passiveportrange <min port:max port>
```

*N.B. The real servers' firewall should also be configured to limit the ports to the same ranges.*

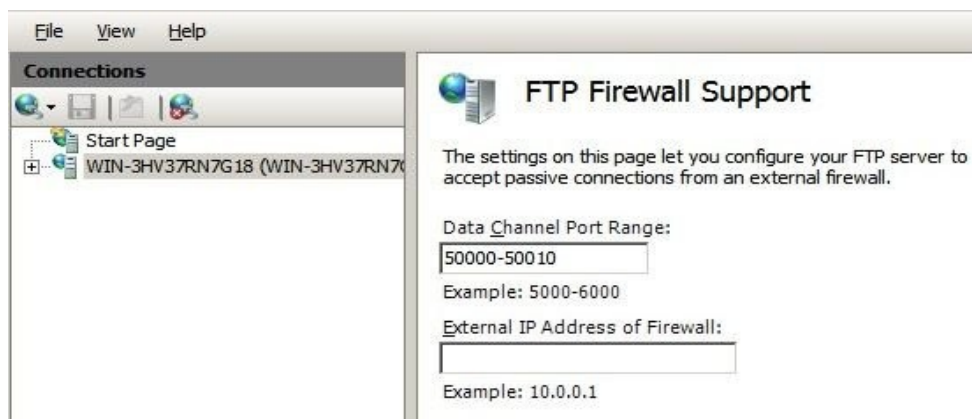


For Windows 2008

Open the IIS Management console, highlight the server node, then double-click the FTP Firewall Support icon.



The following screen will be displayed:



Enter the required port range in the Data Channel Port Range field and apply the changes. These settings will apply to all FTP sites created on the server.

For Windows 2003

**a) Enable Direct Metabase Edit**

1. Open the IIS Management Console
2. Right-click on the Local Computer node
3. Select **Properties**
4. Make sure the **Enable Direct Metabase Edit** checkbox is checked

**b) Configure PassivePortRange via ADSUTIL script**

1. Click **Start**, click **Run**, type cmd, and then click **OK**
2. Type cd Inetpub\AdminScripts and then press ENTER
3. Type the following command from a command prompt  
adsutil.vbs set /MSFTPSVC/**PassivePortRange** "50000-50100"
4. Restart the FTP service

For Windows 2000

**Configure PassivePortRange via the Registry Editor**

1. Start Registry Editor (Regedt32.exe)
2. Locate the following registry key:  
**HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\Msftpsvc\Parameters\**
3. Add a value named "PassivePortRange" (without the quotation marks) of type REG\_SZ
4. Close Registry Editor
5. Restart the FTP service

(SP4 must be installed for this to work)

**Note:** The range that FTP will validate is from 5001 to 65535

## Terminal Services & RDP

### Layer 4 – IP persistence

RDP is a TCP based service usually on port 3389. Because of the nature of a Terminal Server you'll want the clients to reconnect to the same server so that you maintain the session. The common setting to use with Terminal Server is *persistence=3600* (1 hour). This means that when a client reconnects within this time, they will be sent to the same terminal server. If a client is idle for more than 1 hour, then the load balancer will treat the next connection as a new connection and possibly take them to a different server.

Label	<input type="text" value="RDP_Cluster1"/>	<a href="#">?</a>
Virtual Server IP address	<input type="text" value="192.168.2.165"/>	<a href="#">?</a>
Virtual Server Ports	<input type="text" value="3389"/>	<a href="#">?</a>
Persistent	<input type="button" value="Yes"/>	<a href="#">?</a>
Persistence Timeout	<input type="text" value="3600"/>	<a href="#">?</a>
Scheduler	<input type="button" value="Weighted Least Connection"/>	<a href="#">?</a>

### Layer 7 - RDP cookies

In some instances source IP persistence can result in uneven load balancing. This would normally happen if you have a large number of users coming through a corporate firewall or proxy. If a large number of users have the same source IP address they will all hit the same back end server.







If you have this issue the Load balancers also support persistence based on RDP cookies. This method utilizes the cookie sent from the client in the Connection Request PDU. This cookie is created when the username is entered at the first client login prompt (mstsc.exe). Note that if the username is not entered here, the cookie is not created.

Label	<input type="text" value="RDP_Cluster2"/>	<a href="#">?</a>
Virtual Server IP address	<input type="text" value="192.168.2.166"/>	<a href="#">?</a>
Virtual Server Ports	<input type="text" value="3389"/>	<a href="#">?</a>
Layer 7 Protocol	<input type="button" value="Other TCP"/>	<a href="#">?</a>
Persistence mode	<input type="button" value="RDP Cookie"/>	<a href="#">?</a>
Balance mode	<input type="button" value="Round Robin"/>	<a href="#">?</a>

Since this method uses a hash of the username to distribute session, it is not necessary to configure a balance mode.

## Layer 7 – Microsoft Connection Broker / Session Directory

Its also possible to configure the load balancer to interact with Session Directory / Connection Broker by enabling Routing Token Redirection mode. This mode allows the reconnection of disconnected sessions by utilizing a routing token to enable the load balancer to re-connect the client to the correct terminal server.

Label	<input type="text" value="RDP_Cluster3"/>	
Virtual Server IP address	<input type="text" value="192.168.2.167"/>	
Virtual Server Ports	<input type="text" value="3389"/>	
Layer 7 Protocol	<input type="text" value="Other TCP"/>	
Persistence mode	<input type="text" value="MS Connection Broker"/>	
Balance mode	<input type="text" value="Least Connections"/>	



If your back-end servers are running Windows 2008 (R1) / Terminal Services, make sure that the Security Layer setting of the RDP connection properties are set to RDP Security Layer, otherwise the RDP Cookie may be encrypted and will not be readable causing persistence to break.



For additional information, please refer to our Terminal Services Deployment Guide available here : [http://loadbalancer.org/pdffiles/Microsoft\\_Terminal\\_Services\\_Deployment\\_Guide.pdf](http://loadbalancer.org/pdffiles/Microsoft_Terminal_Services_Deployment_Guide.pdf)

## Software Updates

Loadbalancer.org continually develop and add new & improved features to the appliance. To ensure that customers can benefit from this development and can also receive bug and security updates, Loadbalancer.org have an online and an offline update facility that allows customers who have a valid maintenance and support contract to keep their appliance fully up to date.



Since services can be restarted during the update process we recommend performing the update during a maintenance window

### Online Update

This option is used to update the appliance via the Internet.

#### MAINTENANCE > SOFTWARE UPDATE

##### Online Update

Online updates are only available if your organisation has a valid authorisation key.  
An authorisation key can be obtained from Loadbalancer.org Limited.

Version v7.2 is the current version (no updates available).

Auth Key 999

Online Update

To use this, simply enter your Auth-Key and click **Online Update**

Make sure that:

- The load balancer is able to access the Internet – if you have a proxy server, this can be defined using *Edit Configuration > Physical Advanced Configuration*
- The default gateway is set correctly (*Edit Configuration > Routing*)
- The DNS servers are set correctly (*Edit Configuration > Hostname & DNS*)



The auth code is included in your technical support document supplied to you when the appliance was initially purchased or when your support contract was renewed. If you do not have a valid contract, please contact [sales@loadbalancer.org](mailto:sales@loadbalancer.org)

## *Offline Update*

If the load balancer does not have access to the Internet, Offline update can be used.

Please contact [support@loadbalancer.org](mailto:support@loadbalancer.org) to obtain the download(s) required by your appliance.

## *Updating a Clustered Pair*

To update a clustered pair (i.e. a master unit and a slave unit) , follow the steps below:

1. First perform the update to the slave unit using the online or offline update method described previously. Take care to follow any on-screen instructions that are displayed (e.g. service restarts)
2. Now update the master unit in the same way



For a clustered pair , we strongly recommend fully testing & validating the master / slave failover process before going live. If testing was not carried out before go-live, we recommend scheduling a maintenance window to do this. For detailed steps, please refer to page 127.

## Adding a Slave Unit after the master has been configured

To add a second slave unit to an existing master unit to create a highly available clustered pair, follow the steps below:

### 1) On the slave unit:

- set the slave appliance's IP address at the console using the command:  

```
ip addr add dev eth0 <IP address>/<mask>
```

e.g.

```
ip addr add dev eth0 192.168.2.100/24
```
- connect to the WUI using a browser : `http://<IP address>:9080`
- using the WUI option : *Edit Configuration > Network Interface Configuration* set the IP address , this makes the setting permanent (setting via the console is temporary)
- click **Configure Interfaces**
- using the WUI option : *Edit Configuration > Hostname & DNS* , change the hostname to 'lbslave'
- also change the DNS server to a valid IP address
- click **Update**

**\* Now Connect the serial cable between the 2 units (if ucast is used for heartbeat, this is not required)**

### 2) On the master unit:

- using the WUI option : *Edit Configuration > Hostname & DNS* enter the IP address of the slave unit in the 'Slave Load Balancer' field
- check the box 'Force full slave sync'
- click **Update**



**IMPORTANT** – The following steps must be done during a maintenance window since all services will be restarted causing end-user disruption

- using the link presented, restart heartbeat

### 3) On the slave unit:

- using the WUI option : *Maintenance > Restart Services* , click **Restart Heartbeat**

Allow time for heartbeat to restart and synchronise (approx 1min) , then continue to the next section to verify failover to the slave and fallback to the master.

## Verifying Master / Slave replication & testing failover

- 1) On the master click *System Overview* and verify that the system status appears as follows:



- 2) Also check the system status in the same way on the slave unit:



- 3) Verify that settings been replicated to the slave unit, this can be done by using either the *View Configuration* or *Edit Configuration* menus to validate that the same Virtual & Real servers exist on the slave
- 4) During a maintenance window, verify failover to the slave and failback to the master by following the steps below:

1. On the slave using the WUI option : *Edit Configuration > Execute a Shell Command* run the command: **/usr/local/sbin/hb\_takeover.php all**

now verify that the slave's status has changed to Active as follows:



and the master has changed to Passive:



Also, using the WUI option : *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up on the slave unit and brought down on the master

2. On the master using the WUI option : *Edit Configuration > Execute a Shell Command* run the command: **/usr/local/sbin/hb\_takeover.php all**

now verify that the master's status has changed to Active as follows:





3. and the slave has changed to Passive:



Also, using the WUI option : *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up on the master unit and brought down on the slave

- 5) During a maintenance window, verify failover to the slave and failback to the master by following the steps below:

1. Power down the master using the WUI option : *Maintenance > System Control > Halt Server*

now verify that the slave's status has changed to Active as follows:



Also, on the slave using the WUI option : *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up

2. Power up the master

now verify that the master's status has changed to Active as follows:



and the slave has changed to Passive:



Also, using the WUI option : *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up on the master unit and brought down on the slave

**NOTE:** This assumes that the option 'Automatic Fail-back' is enabled. This is enabled by default and can be set using the WUI option : *Edit Configuration > Heartbeat Configuration*. If not enabled, the services will not auto failback to the master when it becomes available, but will remain active on the slave. Services must be moved back to the master manually using the following command on the master: `lusr/local/sbin/hb_takeover.php all`

## **Section F – Disaster Recovery**

## Being prepared

To be able to quickly recover your appliance when a disaster occurs it is important that you create a backup of the XML configuration file and keep it stored in a safe location off the load balancer. Ideally you should keep a backup of both the master and slave configurations. This can easily be done by following the steps below:

### *Backing up to a remote location*

Login to the web interface:

***Username:*** loadbalancer  
***Password:*** loadbalancer

- Select *Maintenance > Backup & Restore > Download XML configuration file*
- Save the file in a secure location

Also for the firewall configuration:

- Select *Maintenance > Backup & Restore > Download Firewall Script*
- Save the file in a secure location

### Using wget to copy the files

It's also possible to use wget from a Linux session on a remote machine to pull the XML configuration file and firewall script:

```
wget --user=loadbalancer --password=loadbalancer http://<IP>:9080/lbadmin/config/getxmlconfig.php
```

```
wget --user=loadbalancer --password=loadbalancer http://<IP>:9080/lbadmin/config/getfirewall.php
```

*NOTE: Replace the password 'loadbalancer' with your password*

### *Backing up to the load balancer*

To create a backup that is stored on the load balancer itself, follow these steps:

Log in to the web interface:

***Username:*** loadbalancer  
***Password:*** loadbalancer

- Select *Maintenance > Backup & Restore > Make local XML backup*
- Select *Maintenance > Backup & Restore > Make local Firewall Script backup*
- A copy of both files will be stored in /etc/loadbalancer.org/userbkup

## Appliance recovery using a USB memory stick

The following instructions detail how to recover a Loadbalancer.org appliance to the latest version using a USB stick (1GB or more in capacity).



This will only work on 64Bit hardware. From version 6.0 onwards, all appliances are 64Bit. If you are running an older version, this may or may not be possible depending on the hardware.

If you are running v5.x and wish to determine whether your appliance is 64Bit, then enter the following command:

```
grep flags /proc/cpuinfo
```

If **lm** (long mode) is present in the output then the CPU is 64Bit and you can proceed. If not then your appliance is 32Bit and you are limited to the latest v5 software.

The latest images require a standard disk (Dell hardware) or a high speed IDE DOM / SATA SSD (Supermicro hardware). If you are already running v6 then you will already have this and should be able to simply re-image your current drive / disk module. If you are upgrading from v5 you will need to purchase a suitable device and then use the following procedure to build it from the USB stick.



It's not currently possible to import an XML file from a v6.x appliance to a v7.x appliance

The latest version of the ISO file is available at <http://www.loadbalancer.org/download/>

You can use [UNetBootin](#) (Windows or Linux) to transfer the ISO onto the USB stick.

Make sure you change the server BIOS to boot from the USB first (the stick must be plugged in at that stage to allow it to be selected as a boot device).

When it boots choose:

Default image

Once the **root@lbmaster:/ #** prompt appears, enter the following commands:

```
# cd /etc/recovery
# ./clone-dsk.sh
```

at the first prompt, press <ENTER>

at the second prompt, select option 1

at the third prompt, select option 1

at "Is the disk /dev/hda already formatted (manually) [Y/N]?" type 'N'

at "do you want to reformat /dev/hda [Y/N]?" type 'Y'

then Yes to all other prompts

The image is then transferred onto the disk / module. This takes around 5 mins to complete depending on the hardware specification of the system.

Once complete, remove the USB stick and reboot the appliance

**\*\*\*\* You now have a fully functioning appliance \*\*\*\***

Now continue with the relevant slave / master recovery steps to configure the device with your particular configuration.

## Disaster recovery after master failure

For a correctly configured clustered pair, if the master fails, the slave will take over automatically. To restore the master load balancer's configuration, a backup copy of the lb\_config.xml file is used. This backup should be created using the steps on page 131.



NOTE: If a backup copy from the master is not available, It's possible to use the lb\_config.xml from the slave instead. If there is no current backup of this, then use the WUI option: *Maintenance > Backup & Restore > Download XML Configuration file* to create the file. A couple of changes need to be made so the file represents the master unit rather than the slave as shown below.

### **Steps to modify a copy of the configuration file from the slave, for use on the master:**

#### **Change:**

```
<network>
  <hostname>lslave</hostname>
  <slave></slave>
```

#### **To:**

```
<network>
  <hostname>lbmaster</hostname>
  <slave>192.168.2.165</slave>
```

*(change to 'lbmaster')*  
*(specify the IP address of your slave unit)*

#### **Change:**

```
<eth0>
  <ip>192.168.2.165</ip>
  <netmask>255.255.255.0</netmask>
```

#### **To:**

```
<eth0>
  <ip>192.168.2.164</ip>
  <netmask>255.255.255.0</netmask>
```

*(change to the IP address of your master unit)*

**NB.** If you use eth1, eth2 (Ent. MAX & 10G only) or eth3 (Ent. MAX & 10G only) these should also be changed in the same way

### **To perform the recovery**

- Locate your copy of lb\_config.xml (either the backup from the master, or the modified slave copy)
- If the failed master is still on, power it down
- Disconnect all cables
- Repair the problems you are having with the master
- Connect the power lead, mouse, monitor and keyboard
- Restore the master from the Load balancer ISO image using a USB stick by following the steps on page 132.
- Log onto the console of the master appliance as:

**Username:** root

**Password:** loadbalancer

- At the console configure the IP address (replace with your IP/mask) :

```
# ip addr add dev eth0 192.168.2.164/24
```



**IMPORTANT** – The following steps must be done during a maintenance window since all services will be restarted

- On the console of the repaired master unit run the following command to stop heartbeat:

```
# service heartbeat stop
```

- On the console of the slave unit run the following command to stop heartbeat:

```
# service heartbeat stop
```

- Once heartbeat has successfully stopped on both units, re-connect the network cable to the repaired master and the serial cable (if used for heartbeat) between the two units

- Open the WUI of the repaired master (replace with your IP address) using:

```
http://192.168.2.164:9080
```

- Login to the WUI:

**Username:** loadbalancer

**Password:** loadbalancer

- Restore your XML file using *Maintenance > Backup & Restore > Upload XML file & Restore*

- Check that all settings have been restored as expected

- To Synchronise heartbeat between master & slave restart Heartbeat on ***both units*** using *Maintenance > Restart Services > Restart Heartbeat*

- If auto-failback is set to 'on' then the repaired master should now take over all services. If it's set to 'off', then you can manually failback to the repaired master using the following command on the master:

```
# /usr/local/sbin/hb_takeover.php all
```

### To Verify the status

After a minute or so your cluster should be restored with the repaired master unit as the active appliance and the slave as the passive appliance.

To verify this, the master units current status is displayed at the top of the WUI as shown below:



This shows that the unit is the **Master**, its currently **Active** and that the **Link** to the slave has been successfully established.

## Disaster recovery after slave failure

If the slave unit has failed, the master will continue to provide load balancing functions as normal. However it is important to recover the slave unit as soon as possible to restore the clustered pair to normal. To restore the slave there are two options:

1. Repair the unit, then restore the slave's XML backup file
2. Repair the unit, then use the 'Force full Sync' option on the master to synchronise the slave

### *Option 1 – Using the XML Backup*

- Locate your up-to-date copy of the lb\_config.xml slave backup file
- If the failed slave is still on, power it down
- Disconnect all cables
- Repair the problems you are having with the slave
- Connect the power lead, mouse, monitor and keyboard
- Restore the slave from the Load balancer ISO image using a USB stick by following the steps on page 132.
- Log onto the console of the slave appliance as:

**Username:** root  
**Password:** loadbalancer

- At the console configure the IP address (replace with your IP/mask) :  

```
# ip addr add dev eth0 192.168.2.164/24
```



**IMPORTANT** – The following steps must be done during a maintenance window since all services will be restarted

- On the console of the repaired slave unit run the following command to stop heartbeat:  

```
# service heartbeat stop
```
- On the console of the master unit run the following command to stop heartbeat:  

```
# service heartbeat stop
```
- Once heartbeat has successfully stopped on both units, re-connect the network cable to the repaired slave and the serial cable (if used for heartbeat) between the two units
- Open the WUI of the repaired slave (replace with your IP address) using:  

```
http://192.168.2.164:9080
```
- Login to the WUI:  
**Username:** loadbalancer  
**Password:** loadbalancer
- Restore your XML file using *Maintenance > Backup & Restore > Upload XML file & Restore*

- Verify the configuration to ensure all settings have been restored as expected
- To Synchronise heartbeat between master & slave restart Heartbeat on ***both units*** using *Maintenance > Restart Services > Restart Heartbeat*

### **To Verify the status**

After a minute or so your cluster should be restored with the master unit as the active appliance and the repaired slave as the passive appliance.

To verify this, the slave units current status is displayed at the top of the WUI as shown below:



This shows that the unit is the **Slave**, its currently **Passive** and that the **Link** to the master has been successfully established.

### ***Option 2 – Synchronizing from the Master***

- If the failed slave is still on, power it down
- Disconnect all cables
- Repair the problems you are having with the slave
- Connect the power lead, mouse, monitor and keyboard and power on the unit

Now follow the steps in the section 'Adding a Slave Unit after the master has been configured' on page 127.



## **Section G – Web User Interface Reference**

## System Overview

View an overview of system performance and cluster status. Also allows servers to be taken offline / online as needed. To take servers offline, there are two options:

- **Drain** – Allows connections to close gracefully
- **Halt** – Drop all connections immediately, do not wait

*NB. If you request to drain / halt all the real servers, the fallback server will NOT be activated. The fallback server only comes into effect when all servers fail their health-check ?? is that correct*

## View Configuration

### *XML*

View the lb\_config.xml configuration file. This details the main configuration for the appliance.

### *Layer 4*

View the layer 4 configuration file.

### *Layer 7*

View the haproxy.cfg configuration file.

### *SSL termination*

View the pound.cfg configuration file.

### *Heartbeat configuration*

View the ha.cfg configuration file.

### *Heartbeat resources*

Displays the contents of the /etc/ha.d/conf/haresources file.

### *Network configuration*

View the running configuration of the network of the load balancer.

### *Routing table*

View the routing table of the appliance.

## Firewall rules

View all firewall rules configured on the appliance.

## Edit Configuration

Set up or modify the physical and virtual configuration of the load balancer appliance.

## Layer 4 – virtual servers

This menu option allows you to add, remove or modify Virtual Servers. Each Virtual Server can have an unlimited number of real servers (except the Enterprise R16 which is limited to 4 x VIPs each with up to 4 RIPs).

You need one Virtual Server for each distinct cluster *AND* protocol that you wish to load balance.

So if you want to serve both HTTP and HTTPS then you will need two virtual servers:

192.168.2.128:80

and

192.168.2.128:443

loadbalancer.org ENTERPRISE R16 v7.1 English

Master | Slave Active | Passive Link

EDIT CONFIGURATION > VIRTUAL SERVERS

[ Add a new Virtual Server ]

HTTP_Cluster	192.168.2.128	Ports 80	Direct Routing	[ Modify ]	[ Delete ]
HTTPS_Cluster	192.168.2.128	Ports 443	Direct Routing	[ Modify ]	[ Delete ]

Adding a Virtual Server is a simple case of specifying the label (name), IP address & port. Other settings can be left at default values which are appropriate in many cases. If you require the client connections to be persistent (i.e. stick to the first real server they hit), then change persistence to 'yes'. This is recommended for HTTPS to stop clients repeatedly re-negotiating SSL keys.

Layer 4 persistence is based on source IP address & destination port. The time out value is in seconds and each time the client makes a connection the timer is reset, so even a 5 minute persistence setting could last for hours if the client is active and regularly refreshes their connection.

The load balancer will automatically add the Virtual Server to the pool of Floating IP(s). The Floating IP should activate instantly. Just check *View Configuration > Network Configuration* to ensure that the Floating IP address has been activated correctly. They will show up as aliases, i.e. eth0:0, eth0:1 etc.

## Adding a virtual server

### EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Server Ports	<input type="text" value="80"/>	?
Forwarding Method	<input type="text" value="Direct Routing"/>	?
Persistent	<input type="text" value="no"/>	?
Protocol	<input type="text" value="TCP"/>	?

**Label** – Set the name for the Virtual Server

**Virtual Server IP address** – Set the IP address of the Virtual Server

**Virtual Server Ports** – Configure the ports for the Virtual Server. For multiple ports separate them by commas, for a range use a dash, for all ports use an asterix. e.g. 80,81,90-95

**Forwarding Method** – The routing method that the load balancer uses to forward packets to the real servers:

*Direct Routing (DR)* - This is the default in 1 Arm mode (Direct Routing). Direct Routing is recommended because it is easy to understand and implement with two load balancers in failover mode - the suggested configuration for all appliances. It only requires one external Floating IP address on the same subnet as your web server cluster and only one network card.

A separate firewall is required, as is a NAT gateway if the network uses private IP addresses.

Direct Routing changes the destination MAC address of the Ethernet frame to redirect it to a server in the cluster, without modifying the IP packet. Each real server must therefore be configured to respond to the Virtual IP, but must not respond to ARP requests for that IP. This is known as the ARP Problem.

The other advantage of Direct Routing is that each web server can reply through its own default gateway at gigabit speeds without needing the packets to return through the loadbalancer.

*NAT* - This is the default in 2 Arm mode (Network Address Translation). This has the advantage that you can load balance any device without having to deal with the ARP problem. The real servers need their default gateway changed to be the internal floating VIP of the load balancer. Because the load balancer handles the return packet you will get more detailed statistics but slower speed than DR or TUN. NAT can also be implemented with a single NIC just use the firewall script to set up an alias on the eth0 interface.

*Tunneling* - This is for WAN links (Tunneling). Tunneling has somewhat limited use as it requires an ip tunnel between the load balancer and the real server as the VIP is the target address many routers will drop the packet assuming that it has been spoofed. However it is useful for private networks with real servers on multiple subnets.

**Persistent** - Enable persistence for this Virtual Server, by Source IP or SIP call ID. Sticky or persistent connections are required for some protocols such as FTP and SIP. It is also kind to clients when using SSL, and unfortunately sometimes required with HTTP if your web application cannot keep state between real servers. NB: If your real servers cannot keep session state persistence themselves, then you will obtain performance but not reliability benefits from a load balancer.

**Protocol** - Select the protocol to load balance (usually TCP):

*TCP* - Transmission Control Protocol (STD0007, RFC0793). The default and most common option.

*UDP* - User Datagram Protocol (STD0006, RF0768). Used for DNS, SIP, etc.

*One Packet Scheduling* - for UDP SIP connections.

*Firewall Marks* - For use when traffic has been tagged in the firewall script using the MARK target.

### **Modifying a virtual server**

When first adding a Virtual Server, only certain values can be configured, others are set at their default setting. These values can be changed after the Virtual Server has been created by clicking **[Modify]** next to the relevant Virtual Server. Settings that can be changed are:

**Persistence Timeout** - How long do you want connections to be sticky? The persistence time is in seconds and is reset on every connection; i.e. 5 minutes persistence will last for ever if the client clicks on a link within that period.

**Scheduler** - Configure how connections are distributed to the real servers:

*Least-Connection* - assign more jobs to real servers with fewer active jobs.

*Weighted Least-Connection* - assign more jobs to servers with fewer jobs, relative to the real servers' weight.

*Robin Robin* - distribute jobs equally amongst the available real servers.

*Weighted Round Robin* - assign jobs to real servers proportionally to the real servers' weight. Servers with higher weights receive new jobs first and get more jobs than servers with lower weights. Servers with equal weights get an equal distribution of new jobs. This is the default.

*Destination Hashing* - assign jobs to servers through looking up a statically assigned hash table by their destination IP addresses.

*Source Hashing* - assign jobs to servers through looking up a statically assigned hash table by their source IP addresses.

**Fallback Server** - The server to route to if all of the real servers in the group fail the health check. The local nginx fallback server is configured for the ports 80 and 9081 (configured to always show the index.html page). You can also configure the the fallback server to be a 'Hot Spare' if required. For example you have one server in the cluster and one fallback they will act as a master / slave pair.

**Fallback Server Port** – Set the port for the fallback server. In DR mode, since port redirection is not possible, the port is automatically set to be the same as the virtual server.

**Check Type** – Specify the type of health-check for the real servers.

**Service to Check** – Specify the protocol to use when check-type is set to negotiate.

**Check Port** - If you want the Service to check to be say HTTPS but not on the default port (443) then you can specify that here.

**Check Command** - The custom check script, used with the external check type. The script should be placed in `/var/lib/loadbalancer.org`, and given world read and execute permissions.

**Virtual Host** - If the real server will only respond to a URL or 'virtualhost' rather than an ip address for its health checks, you can specify the virtualhost to request here.

**Login** - The login name to use with negotiate checks where authentication is required.

**Password** - The password to use with negotiate checks where authentication is required.

**Granularity** - Specify a whole subnet to use instead of source ip for persistence. Some large ISPs use clustered proxies this means that the clients source ip address may keep changing. If you require persistence of HTTP and this is causing a problem then you can set a larger masq on the source ip address match for persistence i.e. `255.255.255.0` for a whole class C subnet. NB. Single IP `255.255.255.255` is the default.

**Request to Send** – Used when Check Type is set to Negotiate. This specifies the request to send to the server. The use of this parameter varies with the protocol selected in Service to Check.

**Response Expected** - This string will be matched against the response to a negotiate check. If the string matches anywhere in the response data, the negotiate check is considered a success.

**Email Alerts** - Specify an email address for server health alerts. Email alerts can be specific to one virtual server or they can be a global setting.

**Feedback Method** - The method the load balancer uses to measure the performance of the real servers:

*Agent* - A simple telnet to port 3333 on the real server

*HTTP* - A simple HTTP GET to port 3333 on the real server

*None* - No feedback (default setting)

The loadbalancer expects a 0-99 integer response from the agent, usually relating to the CPU idle; i.e. a response of 92 would imply that the real servers CPU is 92% idle. The load balancer will then use the formula  $((92 / 10) * \text{requested\_weight})$  to find the new weight. Using this method an idle real server will get 10 times as many new connections as an overloaded server.



For more details on configuring health checks, please refer to page 99

## Layer 4 – real servers

This menu option allows you to add, remove or modify Real Servers. You can add an unlimited number of real servers to each Virtual Server (except the Enterprise R16 which is limited to 4 x VIPs each with up to 4 RIPs).

In DR mode, since port redirection is not possible the Real Server port field is not available and the port is automatically set to be the same as the Virtual Server, whilst for a NAT mode Real Server, its possible to configure the port to be the same or different than the virtual Servers' port.

The screenshot shows the Loadbalancer.org Enterprise R16 v7.1 web interface. The top navigation bar includes the logo, version information, and a language dropdown set to 'English'. Below the navigation bar are tabs for 'Master', 'Slave', 'Active', 'Passive', and 'Link'. The main content area is titled 'EDIT CONFIGURATION > REAL SERVERS'. It displays a table of real servers for two clusters: HTTP\_Cluster1 and HTTP\_Cluster2. Each cluster has two real servers, real\_server1 and real\_server2, with their respective IP addresses, ports, and weights. Action links for 'Add a new Real Server', 'Modify', and 'Delete' are provided for each server. A sidebar on the left contains a menu with options like 'System Overview', 'View Configuration', 'Edit Configuration', and various Layer 4 and Layer 7 configurations. The footer indicates the copyright for Loadbalancer.org Limited from 2002 to 2011.

Cluster	Real Server	IP Address	Port	Routing	Weight	Actions
HTTP_Cluster1	real_server1	192.168.2.150	80	Direct Routing	1	[ Add a new Real Server ] [ Modify ] [ Delete ]
	real_server2	192.168.2.160	80	Direct Routing	1	[ Add a new Real Server ] [ Modify ] [ Delete ]
HTTP_Cluster2	real_server1	192.168.2.110	80	Direct Routing	1	[ Add a new Real Server ] [ Modify ] [ Delete ]
	real_server2	192.168.2.120	80	Direct Routing	1	[ Add a new Real Server ] [ Modify ] [ Delete ]

Adding / modifying a new real server is a simple case of specifying IP address, port number and weight. Other settings can be left at default values which are appropriate in many cases.

The forwarding method defaults to that defined for the virtual server and you will normally leave this as DR. NAT can be used when you have two Floating Virtual IP(s) set up (one internal and one external) and TUN can be used to route through a tunnel across the Internet or WAN.

### Adding / modifying a real server

The screenshot shows the 'EDIT CONFIGURATION > ADD A NEW REAL SERVER' form. It contains several input fields for configuring a new real server. The fields are: 'Label' (with a dropdown menu), 'Real Server IP Address' (with a text input), 'Real Server Port' (with a text input), 'Weight' (with a text input), 'Minimum Connections' (with a text input), and 'Maximum Connections' (with a text input). Each field has a help icon (question mark) to its right. At the bottom of the form is an 'Update' button.

Label	RIP Name	?
Real Server IP Address	IPAddress	?
Real Server Port		?
Weight	1	?
Minimum Connections	0	?
Maximum Connections	0	?

Update

**Label** – Set the name for the Virtual Server.

**Real Server IP address** – Set the IP address of the Real Server.

**Real Server Port** – Configure the port for the Real Server (NAT mode only).

**Weight** - Weight is an integer specifying the capacity of a server relative to the others in the pool. The valid values of weight are 0 through to 65535. The default is 1.

Why would you change the weight of a real server? Say you had a 4 core Xeon web server and a single core Celeron web server, you could increase the weight of the Xeon based server so that it took more of the load.

**Minimum Connections** - An integer specifying the lower connection threshold of a server. The valid values are 0 through to 65535. The default is 0, which means the lower connection threshold is not set.

If Minimum Connections is set with other values, the server will receive new connections when the number of its connections drops below its lower connection threshold. If Minimum Connections is not set but Maximum Connections is set, the server will receive new connections when the number of its connections drops below three fourths of its upper connection threshold.

**Maximum Connections** - An integer specifying the upper connection threshold of a server. The valid values of Maximum Connections are 0 through to 65535. The default is 0, which means the upper connection threshold is not set.

If Maximum Connections is set with other values, no new connections will be sent to the server when the number of its connections exceeds its upper connection threshold.



## Layer 4 – advanced configuration

This section allows you to configure the global timeouts and logging options for the load balancer.

### EDIT CONFIGURATION > ADVANCED CONFIGURATION

Layer 4		
Check Interval	<input type="text" value="6"/>	<a href="#">?</a>
Check Timeout	<input type="text" value="3"/>	<a href="#">?</a>
Negotiate Timeout	<input type="text" value="5"/>	<a href="#">?</a>
Failure Count	<input type="text" value="1"/>	<a href="#">?</a>
Quiescent	<input type="text" value="no"/>	<a href="#">?</a>
Email Alerts	<input type="text"/>	<a href="#">?</a>
Auto NAT	<input type="text" value="off"/>	<a href="#">?</a>
Multi-threaded	<input type="text" value="yes"/>	<a href="#">?</a>
Fallback	<input type="text" value="yes"/>	<a href="#">?</a>
Disable Write	<input type="text" value="off"/>	<a href="#">?</a>

**Check Interval** - Layer 4 (Ldirectord) health check interval in seconds. If this setting is too low, you may experience unexpected real server downtime.

**Check Timeout** - Layer 4 (Ldirectord) health check timeout in seconds. If this setting is too low, you may induce un-expected real server downtime.

**Negotiate Timeout** - Layer 4 (Ldirectord) negotiate health check timeout in seconds. The negotiate checks may take longer to process as they involve more server side processing than a simple TCP socket connect check. If this setting is too low, you may induce unexpected real server downtime.

**Failure Count** - Layer 4 (Ldirectord) number of times a check has to fail before taking server offline. The time to detect a failure and take down a server will be (check interval + check timeout) \* failure count.

**Quiescent** - When a real server fails a health check, do we kill all connections?

When Quiescent is *yes*, on a health check failure the real server is not removed from the load balancing table, but the weight is set to 0. Persistent connections will continue to be routed to the failed server, but no new connections will be accepted.

When Quiescent is *no*, the server is completely removed from the load balancing table on a health check failure. Persistent connections will be broken and sent to a different real server.

Quiescent only applies to health checks - it has no effect on taking real servers offline in System Overview. To manually force a real server to be removed from the table, set Quiescent to *no* and arrange for the server to fail its health check. This may be done, for example, by shutting down the daemon or service, changing the negotiate check value, or shutting down the server.

**Email Alerts** - Specify the global email alert address. The global email alert address is used to send notifications of real server health check failures. This can also be configured on a virtual server level.

**Auto NAT** - Automatically NAT outbound network connections from internal servers. By default servers behind the load balancer in a NAT configuration will not have access to the outside network. However clients on the outside will be able to access load balanced services. By enabling Auto NAT the internal servers will have their requests automatically mapped to the load balancers external IP address. The default configuration is to map all requests originating from internal network eth0 to the external IP on eth1. If you are using a different interface for external traffic you can select it here. Manual SNAT and DNAT configurations for individual servers can also be configured in the firewall script.

**Multi-threaded** - Perform health checks with multiple threads. Using multiple-threads for health checks will increase performance when you have a large number of virtual servers.

**Fallback Server** - Local Fallback server on / off switch . Configure whether the local fallback server is active or not, sometimes you may want the local fallback server switched off so that it doesn't change the SNMP results table when activated. You may also want to disable it for security purposes.

**Disable Write** - Disable writing to the layer 4 configuration file. When enabled (on) configuration changes via the WUI are not permitted, an on-screen message as shown below is displayed:



This is useful if you want to make manual changes to the configuration file. When disabled (off), changes are permitted via the WUI.



**NOTE:** If manual changes are made to configuration files, then Disable Write is switched off and changes are made via the WUI, the manual changes will be overwritten.

## Layer 7 - virtual servers

This menu option allows you to add, remove or modify Virtual Servers. Each Virtual Server can have an unlimited number of real servers (except the Enterprise R16 which is limited to 4 x VIPs each with up to 4 RIPs).

Layer 7 Virtual Servers support a variety of additional persistence modes including HTTP cookie, RDP cookie whilst still supporting IP address based persistence.

The Layer 7 Virtual Servers are configured separately from Layer 4 servers because they use HAProxy rather than the LVS (Linux Virtual Server) engine.

The screenshot shows the Loadbalancer.org web interface. The top navigation bar includes the logo, version 'ENTERPRISER16 v7.1', and a language dropdown set to 'English'. Below the navigation bar, there are tabs for 'Master', 'Slave', 'Active', 'Passive', and 'Link'. The main content area is titled 'EDIT CONFIGURATION > VIRTUAL SERVERS (HAPROXY)'. It features a button '[ Add a new Virtual Server ]' and a table listing two virtual servers:

Virtual Server	IP Address	Ports	Actions
cluster1	192.168.2.150	Ports 80	[ Modify ] [ Delete ]
cluster2	192.168.2.160	Ports 80	[ Modify ] [ Delete ]

The left sidebar contains a menu with options: System Overview, View Configuration, Edit Configuration, Layer 4 - Virtual Servers, Layer 4 - Real Servers, Layer 4 - Advanced Configuration, Layer 7 - Virtual Servers (highlighted), Layer 7 - Real Servers, and Layer 7 - Advanced Configuration. The footer indicates 'Copyright © Loadbalancer.org Limited 2002 – 2011'.

Layer 7 Virtual Servers are created in the same way as Layer 4 Virtual Servers, but by using a different option in the menu.

If Persistence Mode is set to *None*, the default Balance Mode is *Least Connection*. If Persistence Mode is set to *HTTP Cookie*, the default Balance Mode is *Round Robin*.

When HTTP Cookie persistence mode is used, the inserted cookie name is set to be the same as the Real Server Label (name).

With Layer 7, port re-direction is possible, i.e. VIP:80 → RIP:800 is possible

**NOTE:** Any changes to the Layer 7 configuration requires a restart of the HAProxy service.

## Adding a virtual server

### EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER (HAProxy)

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Server Ports	<input type="text" value="80"/>	?
Persistence mode	<input type="text" value="None"/>	?
Fallback Server	<input type="text" value="127.0.0.1:9081"/>	?
<input type="button" value="Update"/>		

**Label** - Designate a recognizable label for this Virtual Server. The Label is used as the cookie so make sure it is different for each server.

**Virtual Service IP Address** - Specify the virtual service's IP address.

**Virtual Service Ports** - Specify the ports on which the virtual service should accept connections. Individual port numbers should be separated by commas, and ranges may be specified using a dash. To specify all ports, use a single asterisk. For example: 81, 443 - 447, 103, 104, 105

**Persistence Mode** - Set the persistence mode as required:

*HTTP Cookie* - Use an HTTP cookie to ensure a client always hits the same server.

*MS Connection Broker* - The load balancer is able to interact with Session Directory / Connection Broker by enabling Routing Token Redirection mode. This mode allows the reconnection of disconnected sessions by utilizing a routing token to enable the load balancer to re-connect the client to the correct terminal server.

*RDP Cookie* - This method utilizes the cookie sent from the client in the Connection Request PDU to make sure a Terminal Server user always uses the same server. This cookie is created when the username is entered at the first client login prompt (mstsc.exe). Note that if the username is not entered here, the cookie is not created.

*Source IP* - Make sure the same source IP always hits the same server.

*Source Hash* – This is now deprecated, please use an alternative.

None - No persistence, users will use the servers in Round Robin mode.

**Fallback Server** - Set the Fallback server as required. This is where requests go if all servers in the cluster are down. The default port on the load balancer is 127.0.0.1:9081.

The Fallback server can be any server and you can also change the port if required (i.e. port re-direction)

Also refer the console commands **lb1port** & **lb2ports** on page 106 & 107.



For more details on configuring for Terminal Services, please refer to our RDP deployment guide available here :  
[http://loadbalancer.org/pdffiles/Microsoft\\_Terminal\\_Services\\_Deployment\\_Guide.pdf](http://loadbalancer.org/pdffiles/Microsoft_Terminal_Services_Deployment_Guide.pdf)

## **Modifying a virtual server**

When first adding a Virtual Server, only certain values can be configured, others are set at their default setting. These values can be changed after the Virtual Server has been created by clicking **[Modify]** next to the relevant Virtual Server. Settings that can be changed are:

**Layer 7 Protocol** - Select the Layer 7 protocol to be handled by this Virtual Service, either HTTP or any other TCP-based protocol. If this Virtual Service will handle only HTTP traffic, selecting that option here allows more flexibility in the processing of connections. The HTTP Cookie and HTTP application cookie modes, and the X-Forwarded-For header all require HTTP to be selected here. In addition, the HAProxy logs will show more information on the client requests and real server responses.

**Balance Mode** - The scheduler used to specify server rotation. Specify the scheduler to utilize when deciding the backend server to use for the next new connection.

**Timeout** - The time-out period before an idle connection is removed from the connection table. The source ip will be removed from memory when it has been idle for longer than the persistence timeout. The default units are minutes. Only applies when IP address persistence is selected.

**Table Size** - The size of the persistence connection table in KB. The size of the connection table (approx 50 bytes per entry) where connection information is stored to allow a session to return to the same server within the timeout period. The default units are in KB. Only applies when IP address persistence is selected

**Check Port** – Specify the port to use for health checking. If not specified here, the check port will be the same as the Virtual Server port. Useful when you are balancing multiple ports.

**Request to send** - Specify a specific file for the health check. Open the specified file and check for the response expected, useful for checking a server sided script to check the health of the back-end application.

**Response expected** - The content expected for a valid health check on the specified file. The response expected can be any valid regular expression statement.

**Maximum Connections** - Specifies the maximal number of concurrent connections that will be sent to this server. If the number of incoming concurrent requests goes higher than this value, they will be queued, waiting for a connection to be released

**Application Cookie Name** – Used to configure session stickiness on an existing application cookie. Set the name of the cookie here.

**Application Cookie Length** – Set the max number of characters that will be memorized and checked in each cookie value.

**Application Cookie Hold** - Set the time after which the cookie will be removed from memory if unused. If no unit is specified, this time is in milliseconds.



For more details on configuring health checks, please refer to page 99

## Layer 7 – real servers

This menu option allows you to add, remove or modify Real Servers. You can add an unlimited number of real servers to each Virtual Server (except the Enterprise R16 which is limited to 4 x VIPs each with up to 4 RIPs).

The screenshot shows the Loadbalancer.org Enterprise R16 v7.1 web interface. The top navigation bar includes the logo, version information, and a language dropdown set to 'English'. A secondary navigation bar shows 'Master | Slave' and 'Active | Passive' tabs, along with a 'Link' button. On the left, a sidebar menu lists various configuration options, with 'Layer 7 - Real Servers' highlighted. The main content area is titled 'EDIT CONFIGURATION > REAL SERVERS (HAPROXY)'. It contains a table with columns for 'HTTP\_Cluster', IP address, Port, and Weight. Two real servers are listed: 'Server1' (192.168.2.99) and 'Server2' (192.168.2.111), both on Port 80 with a Weight of 1. Each row has 'Modify' and 'Delete' buttons. An 'Add a new Real Server' link is also present. The footer indicates 'Copyright © Loadbalancer.org Limited 2002 – 2011'.

HTTP_Cluster	IP Address	Port	Weight	Actions
Server1	192.168.2.99	Port 80	Weight 1	[ Modify ] [ Delete ]
Server2	192.168.2.111	Port 80	Weight 1	[ Modify ] [ Delete ]

Adding / modifying a new real server is a simple case of specifying IP address, port number and weight. Other settings can be left at default values which are appropriate in many cases.

The Real Servers in a Layer 7 configuration can be on any subnet in any network as long as they are accessible from the load balancer.

### Adding / modifying a real server

The screenshot shows the 'EDIT CONFIGURATION > ADD A NEW REAL SERVER (HAPROXY)' form. It contains four input fields: 'Label' (with 'RIP Name' entered), 'Real Server IP Address', 'Real Server Port', and 'Weight' (with '1' entered). Each field has a help icon (question mark). Below the fields is an 'Update' button.

Label	RIP Name	?
Real Server IP Address		?
Real Server Port		?
Weight	1	?

Update

**Label** - Designate a recognizable label for this Real Server

**Real Server IP Address** - The IP address for the appropriate service on your Real Server.

**Real Server Port** - The port for the appropriate service on your Real Server.

**Weight** - Weight is an integer specifying the capacity of a server relative to the others in the pool. The valid values of weight are 0 through to 65535. The default is 1.



## Layer 7 - advanced configuration

This section allows you to configure the global timeouts and logging options for the load balancer.

### EDIT CONFIGURATION > ADVANCED CONFIGURATION (HAPROXY)

Layer 7 (HAProxy):		
Logging	off ▾	?
Redispatch	on ▾	?
Connection Timeout	4000	?
Client Timeout	42000	?
Srvtimeout	43000	?
Maximum Connections	40000	?
Ulimit		?
Abort on Close	on ▾	?
Transparent Proxy	off ▾	?
Interval	2000	?
Rise	2	?
Fall	3	?
Statistics Password		?
Disable HAProxy Config Write	off ▾	?

Update

**Logging** - Activate detailed logging of the Layer 7 HaProxy service. When activated the HAProxy log is written to /var/log/haproxy.

**Redispatch** - Allows HAProxy to break persistence and redistribute to working servers should failure occur. This setting should not require changing.

**Connection timeout** - HAProxy connection timeout in milliseconds. This setting should not require changing.

**Client Timeout** - HAProxy client timeout in milliseconds. This setting should not require changing.

**Srvtimeout** - HAProxy real server timeout in milliseconds. This setting should not require changing.

**Maximum Connections** - HAProxy maximum concurrent connections. This setting should not require changing, unless you are running a high volume site. See also Maximum Connections for a virtual Server (HAProxy).

**Ulimit** - The maximum number of file descriptors used for layer 7 load balancing. This value is optional. If no value is given then a default value will be used internally. For simple configurations where each virtual server only listens to one address/port a reasonable value is the sum of:

- \* 2 times the number of maximum connections (Global Settings Layer 7)
- \* Number of virtual servers on layer 7 (HAProxy)
- \* Number of real servers
- \* plus 1 for logging purpose

In a more sophisticated environment you should use the number of address/port/proxy tuples instead of the number of virtual servers.

**Abort on Close** - Abort connections when users close their connection. Recommended as the probability for a closed input channel to represent a user hitting the 'STOP' button is close to 100%

**Transparent Proxy** - Enable TPROXY support for Layer 7 Haproxy. TPROXY support is required in order for the real servers behind a layer 7 Haproxy configuration to see the client source IP address. The load balancer must be in a NAT configuration (internal and external sub nets) with the real servers using the load balancers internal Floating IP address as their default gateway.

*N.B. all Layer 4 methods are transparent by default*



For more details on using Tproxy, refer to page 97.

#### X-Forwarded-For Headers

Since the load balancer must be in a NAT configuration (i.e. VIPS & RIPS in different subnets) to utilize TPROXY, it is not always an appropriate solution. In situations such as this, it's possible to use the X-forwarded-for header that is included by default in all layer 7 Virtual Servers. Most web servers can then be configured to record the X-Forwarded-For IP in the log files. For example, with Apache it's simply a change to the log file configuration. For details on how to do this please refer to:

<http://blog.loadbalancer.org/apache-and-x-forwarded-for-headers/>

With Microsoft IIS a third party application is needed. For more details on this, please refer to:

<http://blog.loadbalancer.org/iis-and-x-forwarded-for-header/>

**Interval** - Interval between health checks. This is the time interval between real server health checks in milliseconds.

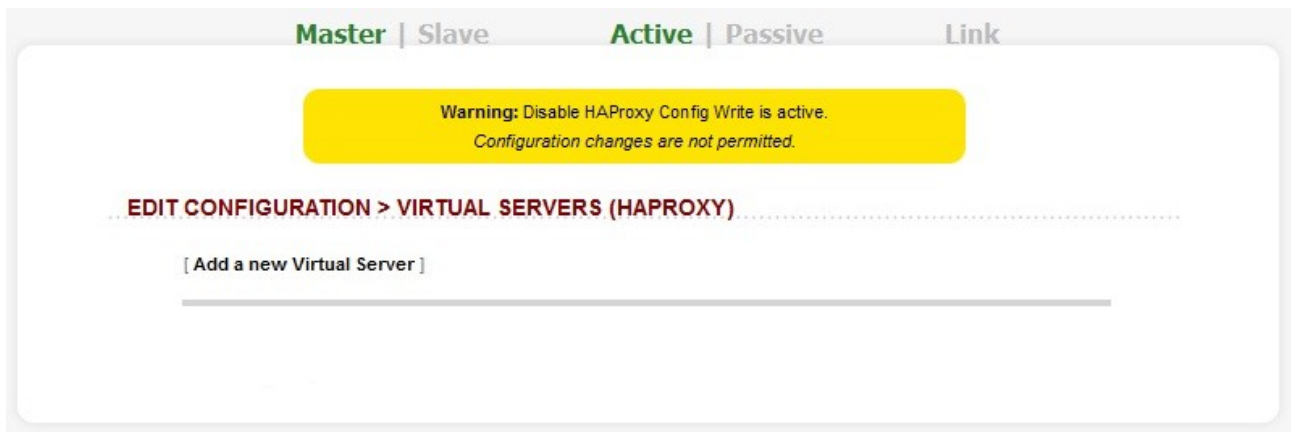
**Rise** - Number of health checks to Rise. The number of positive health checks required before re-activating a real server.

**Fall** - Number of health checks to Fall. The number of negative health checks required before de-activating a real server.

**Statistics Password** – Set the password used to access *Reports > Layer 7 Status*.



**Disable Haproxy Config Write** - Disable writing to the layer 7 configuration file. When enabled (on) configuration changes via the WUI are not permitted, an on screen message as shown below is displayed:



This is useful if you want to make manual changes to the configuration file. When disabled (off), changes are permitted via the WUI.



**NOTE:** If manual changes are made to configuration files, then Disable Write is switched off and changes are made via the WUI, the manual changes will be overwritten.

## SSL termination

If required, SSL can be offloaded to the load balancer. Pound is used to terminate SSL sessions and requires that the SSL certificate be deployed directly on the appliance. Http traffic will then be passed unencrypted to the real servers.

### Layer 7

In order to set up a proxy for the SSL traffic go to *Edit Configuration > SSL Termination*. SSL traffic can be terminated on port 443 and then re-directed to port 80 of the same VIP for HAProxy to pick it up, insert cookies and load balance.

### Layer 4

For layer 4, Pound must be configured in a similar way, but instead of forwarding requests to HAProxy, requests are forwarded to a Layer 4 virtual server configured to operate in NAT mode.

DR mode cannot be used since Pound acts as a proxy, and the real servers see requests with a source IP address of the virtual server. However since the real servers believe that they own the Virtual IP (due to the loopback adapter configured to handle to ARP problem) they are unable to reply to Pound.

The screenshot shows the loadbalancer.org EnterpriseR16 v7.1 web interface. The top navigation bar includes the logo, version information, and a language dropdown set to English. Below the navigation bar, there are tabs for Master, Slave, Active, Passive, and Link. The main content area is titled 'EDIT CONFIGURATION > SSL TERMINATION ADD A NEW VIRTUAL SERVER'. On the left, a sidebar menu lists various configuration options, with 'SSL Termination' currently selected. The main form contains several input fields for configuring a new virtual server:

Label	VIP Name	?
Virtual Server IP address	10.0.0.20	?
Virtual Server Port	443	?
Backend Virtual Server IP Address	10.0.0.20	?
Backend Virtual Server Port	80	?
Ciphers to use		?

An 'Update' button is located at the bottom right of the form.

By default a self generated SSL certificate is associated with the new Virtual Server. Its also possible to upload your current certificate provided that its in PEM format. Certificates can also be exported from Windows servers, converted to PEM format, then uploaded to the load balancer.

## Adding / modifying an SSL Virtual server

### EDIT CONFIGURATION > SSL TERMINATION ADD A NEW VIRTUAL SERVER

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Server Port	<input type="text" value="443"/>	?
Backend Virtual Server IP Address	<input type="text" value="10.0.0.20"/>	?
Backend Virtual Server Port	<input type="text" value="80"/>	?
Ciphers to use	<input type="text"/>	?

**Label** – Set the name for the Virtual Server

**Virtual Server IP address** – Set the IP address for the Virtual Server.

**Virtual Server Port** - Set the IP address for the Virtual Server. Normally will be port 443.

**Backend Virtual Server IP address** - Set the IP address for the Backend Virtual Server. This is normally the same IP address as the Virtual Server IP address but can be any valid IP. The IP selected must correspond to a Layer 4 NAT mode VIP or a Layer 7 Haproxy VIP which is where the unencrypted traffic will be sent for load balancing.

**Backend Virtual Server Port** - Set the port number for the Backend Virtual Server.

**Ciphers to use** - SSL Ciphers to use. List the SSL ciphers that Pound should accept. Leave blank for the default of any cipher. If you wish to restrict the ciphers that Pound should negotiate with the client, they may be specified here. If the field is left blank, Pound will use the default cipher list.

The ciphers should be specified in OpenSSL cipher list format, and may include individual ciphers or groups. Some examples of valid cipher lists:

- \* SSLv3
- \* TLSv1
- \* SSLv3:HIGH
- \* AES128-SHA:DES-CBC3-SHA:RC4-SHA:RC4-MD5:!SSLv2

## SSL - advanced configuration

### EDIT CONFIGURATION > SSL TERMINATION

SSL Termination		
Logging	off ▼	?
Client Timeout	30	?
Global Server Timeout	60	?
Transparent Proxy	off ▼	?
Disable Write	off ▼	?

Update

**Logging** - Activate detailed logging of the Pound SSL termination service. When activated the Pound log is written to `/var/log/poundssl`.

**Client Timeout** - Configure the global client response timeout in seconds. This setting should not require changing.

**Global Server Timeout** - Configure the global real server response timeout in seconds. This setting should not require changing.

**Transparent Proxy** - Enable TPROXY support in Pound SSL. The combination of Pound, TPROXY, and HAProxy allows SSL termination on the load balancer whilst passing the client's IP address to the real servers. This option only enables TPROXY in Pound - you will also need to enable TPROXY for HAProxy below, and add appropriate rules to the firewall.



One consequence of using transparent proxy with both Pound and HAProxy is that you can no longer access the HAProxy virtual service directly. With transparency turned on HAProxy will only accept traffic from Pound. The way around this is to create two HAProxy virtual services. One will listen on port 80, and be your standard HTTP service. The other will listen on a different port ; 81 for example - and will be the destination for traffic from Pound.



For more details on SSL & Pound configuration steps, please refer to page 91



For more details on using Tproxy, refer to page 97.

## Heartbeat configuration

### EDIT CONFIGURATION > MODIFY HEARTBEAT CONFIGURATION

Serial	<input checked="" type="checkbox"/>	<a href="#">?</a>
Unicast	<input type="checkbox"/>	<a href="#">?</a>
Broadcast	None ▾	<a href="#">?</a>
UDP Port for broadcast & unicast	6689	<a href="#">?</a>
Keepalive	3	<a href="#">?</a>
Deadtime	10	<a href="#">?</a>
Warntime	5	<a href="#">?</a>
Ping node		<a href="#">?</a>
Auto_failback	<input checked="" type="checkbox"/>	<a href="#">?</a>

Modify Heartbeat configuration

**Serial** - Enable or disable heartbeat master/slave communication over the serial port. Serial communication is the preferred method for load balancer pairs located in close proximity. Disabling serial communication will automatically activate console access via the serial port.

**Unicast** - Enable unicast heartbeat master/slave communication. This method of heartbeat communication uses unicast UDP between master and slave, with a destination port given by the UDP Port for broadcast & unicast parameter.

When unicast is enabled, the load balancer determines the correct interface and IP addresses to use based upon the configured slave IP address. Please ensure that the correct slave IP has been entered on the DNS & Hostname page before enabling unicast.

Unicast is the preferred communication method if serial cannot be used.

**Broadcast** - Enable broadcast heartbeat master/slave communication, and choose the interface. This method of heartbeat communication uses broadcast UDP between master and slave, with a destination port given by the UDP Port for broadcast & unicast parameter.

Care must be taken when using broadcast on multiple pairs of load balancers in the same network. Each high-availability pair must operate on a different UDP port if they are not to interfere with each other.

If heartbeat communication over the network is required, it is recommended that unicast be used in preference to broadcast.

**UDP Port for unicast & broadcast** - The UDP port number used by heartbeat for network communication over unicast or broadcast. By default, heartbeat uses port 694/udp for unicast or broadcast communication. If you have multiple load balancer pairs on the same subnet, and wish to use broadcast, you will need to set each pair to a different UDP port.

**Keepalive** - Specify the number of seconds between keepalive pings. The Keepalive setting must be less than the warntime and deadtime.

**Deadtime** - The number of seconds communication can fail before a fail over is performed. A very low setting of deadtime could cause un-expected fail overs.

**Warntime** - If communication fails for this length of time write a warning to the logs. This is useful for tuning your deadtime without causing failovers in production.

**Ping node** - Specify a mutually accessible IP address to test network availability. A good ping node to specify is the IP address of a router that both the master and slave node can access. If one node loses access to the ping node then a failover will occur. However if both nodes lose access nothing will change.

**Automatic Fail-back** - When the master returns to service after a failure, should it become active again? This option controls the cluster behavior when the master returns to service after a failure. With Automatic Fail-back enabled, the master will automatically return to active status, taking back the floating IP addresses from the slave. With Automatic Fail-back disabled, the slave will remain active and will retain the floating IP addresses. Fail-over back to the master may then be controlled manually.



For more details on heartbeat, please refer to page 99

## Floating IPs

In order for the load balancer to function, the unit must physically own the Virtual IP address that the clients are accessing before they get re-directed to a real server in the cluster. The Floating IP(s) are controlled by heartbeat to ensure that only one of the load balancers (normally the master) owns the Floating IP(s). The floating IP(s) are added automatically when new Virtual Servers are created. It's also possible to manually define the Floating IP(s) if required, this is normally only required when in layer 4 NAT mode where its recommended to use a floating IP address for the default gateway for the real servers.

### EDIT CONFIGURATION > EDIT FLOATING IP

192.168.2.122	[ Delete ]
192.168.2.123	[ Delete ]

### EDIT CONFIGURATION > ADD NEW FLOATING IP

Update

To add an IP address simple type the address into the field and click update. The IP address must be on a valid subnet for the load balancer.



NOTE: Floating IPs are not deleted automatically when Virtual Servers are removed, this must be done manually.

## Hostname & DNS

By default, all appliances are configured as master units. This is controlled via a drop down on the Hostname & DNS screen. The self explanatory options are *lbmaster* and *lbslave*.

When the wizard is used to configure a master/slave clustered pair, the slave unit is configured first and then the master unit. When configuring manually, its common to setup the master first, then add the slave later. If this is done the *Force full slave sync* option should be used to force all settings from the master to be replicated to the slave unit.

**EDIT CONFIGURATION > HOSTNAME & DNS**

Hostname:	lbmaster ▼	?
Slave Load Balancer:	192.168.2.121	?
Force full slave sync:	<input type="checkbox"/>	?
Domain Name Server:	192.168.2.1	?
Domain Name Server2:		?

Update

**Hostname** - Is this unit the master or slave? The hostname must be correct for heartbeat and replication to work as expected.

**Slave load balancer** - Specify the slave load balancers IP address. The slave load balancers IP address is required to activate replication of configuration data.

**Force full slave sync** - Force all current configuration files to the slave unit. If the slave has been disconnected from the network and changes have been made to the master you can force all changes across in one go using this option.

**Domain Name Server** - Specify the IP address of a Domain Name Server. This is required for the online feature and security updates to work, it also enables the reverse look up of IP Address information in reports.

Entering a DNS address will allow any reports that need to carry out a reverse lookup to work correctly and will also allow on-line updates via the Loadbalancer.org web site.

**Domain Name Server2** - Specify the IP address of a second Domain Name Server.



## Network interface configuration

Depending on the type of appliance you are using you may have either 2 or 4 network ports. For units with two interface cards *eth0* is normally used as the internal interface and *eth1* for the external interface. However, unlike other appliances on the market you can use any interface for any purpose giving flexibility to configure the unit as required.

In a standard one-arm configuration you would just need to configure *eth0*, the netmask and the default gateway.

### Typical configurations:

For layer 4 DR mode, only one interface is used – typically *eth0*

For layer 4 NAT mode, two interfaces are normally required, *eth0* for internal, *eth1* for external

For layer 7 (Haproxy), either one or two interfaces can be used depending on your requirements

### EDIT CONFIGURATION > NETWORK INTERFACE CONFIGURATION

The screenshot shows a web-based configuration interface for network settings. It is divided into three main sections: Bonding, VLAN, and IP Address Assignment.

- Bonding:** Contains a checkbox labeled "Bond eth0 & eth1 as bond0:" which is currently unchecked. To the right of the checkbox is a question mark icon and a button labeled "Bond Interfaces".
- VLAN:** Contains two input fields. The first is labeled "Interface:" and has a dropdown menu showing "eth0". To its right is a question mark icon and a button labeled "Add VLAN". The second field is labeled "VLAN ID:" and contains the number "1". To its right is a question mark icon.
- IP Address Assignment:** This section has two large text input areas. The first area, labeled "eth0" on the left, contains the IP address "192.168.2.21/24". The second area, labeled "eth1" on the left, is currently empty.

At the bottom of the interface is a button labeled "Configure Interfaces".

### Bonding

**Bond eth0 & eth1 as bond0** - Create a bonded interface by checking this box and clicking **Bond Interfaces**. This combines eth0 and eth1 as bond0. For units with 4 interfaces, an additional option to bond eth2 and eth3 is shown.

*NOTE: If you are using heartbeat over Ethernet you should modify the heartbeat configuration to use the new interface.*

## VLAN

802.1q VLANs can be defined here. This is typically required if your real servers are connected to specific VLANs. The exact requirements depend on your infrastructure. Native 802.1q VLAN support can be enabled to load balance clusters on multiple VLAN.

## IP address Assignment

**IP Address Assignment**

eth0	10.10.1.10/16
eth1	192.168.2.120/24 192.168.8.120/24

[Configure Interfaces](#)

Add single or multiple IP addresses to the interfaces.



**WARNING:** Obviously it's best to modify network settings whilst the unit is available locally!



For more details on configuring the network, please refer to page 64

## Routing

Used to configure the default gateway for the load balancer.

### EDIT CONFIGURATION > ROUTING

Routing		
Default Gateway	IP v4	<input type="text" value="192.168.2.1"/>
	IP v6	<input type="text"/>
<input type="button" value="Configure Routing"/>		

**Default Gateway IPv4** – set IPv4 default gateway.

**Default Gateway IPv6** – set IPv4 default gateway.

## System Date & Time

The load balancer's local clock is updated once a day using ntp, this requires that your default gateway and DNS are set correctly and that the load balancer has access to the ntp servers.

### EDIT CONFIGURATION > CHANGE THE LOCAL TIME ZONE

The current Date, Time & Time Zone is:

Wed May 11 12:01:57 GMT 2011    [change](#)    [?](#)

Please select a time zone ... ▼

NB. The internal clock is updated once a day using NTP.

**change** – displays update fields for date and time

**Select a time zone** - Timezone can be Coordinated Universal Time (UTC) or GMT based like GMT, GMT + 1 hour, GMT - 1 hour, and so on. Please consider that the GMT+/-X format as it is returned by the system differs from the GMT +/- X hours format. The GMT+/-X based statement follows the POSIX standard which means that GMT+X is X hours west of Greenwich. GMT-X means X hours east of Greenwich. So GMT+X means GMT - X hours and vice-verse.

## Physical – Advanced Configuration

For internet access via a proxy. You will need to configure this for online update to work if your load balancer is behind a proxy server. Leave both fields empty if you don't use a proxy.

### EDIT CONFIGURATION > PHYSICAL - ADVANCED CONFIGURATION

---

**Internet Access:**

Proxy IP Address  ?

Proxy Port  ?

**Firewall:**

Connection Tracking table size  ?

#### Internet Access

**Proxy IP Address** – Set the IP address of the Proxy server.

**Proxy Port** – Set the port of the Proxy server.

#### Firewall

**Connection tracking table size** - Set the size of the firewall connection tracking table, in number of connections. Each connection entry uses approximately 300 bytes of memory, and the default table size is approximately 30,000 connections. High traffic load balancers using NAT mode, or using connection tracking in the firewall script, may see the connection tracking table fill up. Systems experiencing this problem will report the following in the kernel log:

```
ip_conntrack: table full, dropping packet.
```

## Setup Wizard

Starts the setup wizard.

### EDIT CONFIGURATION > SETUP WIZARD

---

Is this unit part of an HA-pair? ☐ yes ☐ no

Allows layer 4 DR mode and layer 4 NAT mode Virtual Servers to be configured.

## Upgrade Appliance

This option allows a license key to be entered to unlock the R16 restrictions. The key is provided when a Enterprise license is purchased.

### EDIT CONFIGURATION > ENTER LICENCE KEY

This unit is currently limited to 4VIPs\*4RIPs. Please enter your License key to upgrade your unit (unlimited VIPs and RIPs).  
If you do not have a license key please contact [sales@loadbalancer.org](mailto:sales@loadbalancer.org):

Enter the license key provided and click **Enter License Key**. To purchase an upgrade key please email [sales@loadbalancer.org](mailto:sales@loadbalancer.org)

## Execute shell Command

Allows OS level commands to be run via the WUI.

### EDIT CONFIGURATION > EXECUTE SHELL COMMAND

This allows you to execute a shell command as user root. The output of the command will be displayed on screen.

**WARNING:** *You should really know what you are doing if you use this function.*

## Maintenance

### *Backup & Restore*

#### MAINTENANCE > BACKUP & RESTORE

##### Backup

- Download XML configuration file
- Download Firewall script
- Download SSL Certificates
- Make local XML backup

##### Restore

- Upload XML file & Restore:
- Restore from the last local XML backup
- Restore Manufacturer's defaults

#### *Backup*

**Download XML configuration file** – allows the load balancer's configuration file to be downloaded and saved where required.

**Download firewall script** - allows the load balancer's firewall script to be downloaded and saved where required.

**Download SSL Certificates** – allows the SSL Certificates to be backed up.

**Make Local XML Backup** – creates a backup of the current XML file in /etc/loadbalancer.org/userbkup

#### *Restore*

**Upload XML file & Restore** – allows an XML file to be uploaded and restored to the load balancer.

Once the restore completes the following message is displayed:

##### Restore

- Upload XML file & Restore:
- Restore from the last local XML backup
- Restore Manufacturer's defaults

**Information:** Configuration files restored.  
Please check the configuration, then restart heartbeat.

**Restore from the last local XML backup** – Restore the last local backup created with the 'Make local XML Backup' option.

**Restore Manufacturer's defaults** – Restore system settings to default values.

*NOTE: Currently the upload facility is not backward compatible with previous versions of the software, i.e. it is not possible to restore a V6.x XML file to a v7.x appliance*

## Restart Services

### Restart Ldirectord

It is unlikely that you will ever need to use this function. It just re-loads the health check configuration file. This does not usually result in any down time for the cluster.

### Restart HAProxy

If any changes are made to the Layer 7 (HAProxy) configuration, including server weights, a manual restart of HAProxy is required. If you get a failure to start please check your configuration to ensure you are not binding to ports that are already in use.

### Restart Pound

Any configuration changes to the SSL termination configuration or server certificates will require a restart of Pound. If you get a failure to start please check your configuration to ensure you are not binding to ports that are already in use.

### Restart Heartbeat

Heartbeat controls the startup of other services on the load balancer and also the fail over between the master and slave units. If you make any changes to the physical IP address then this will automatically restart heartbeat (*on a properly configured cluster this will also force a heartbeat restart on the slave*).

## System Control



**Restart Server** – Shutdown and restart the appliance.

**Halt Server** – Shutdown and halt the appliance.

## Software Update

### MAINTENANCE > SOFTWARE UPDATE

#### Online Update

#### Online Software & Security Updates v7.1

Version v7.1 is the current version (no updates available).

Auth Key	<input type="text" value="999"/>
----------	----------------------------------

If you have a current maintenance agreement for your appliance you can use this form to check for new online updates and install them. For the update to succeed, you will need:

- You will need a valid authorization key.
- You will need your default gateway & DNS correctly configured.
- You will need HTTP access to [www.loadbalancer.org](http://www.loadbalancer.org) enabled through your firewall.

Updates are also available as a complete downloadable ISO software image if preferred.

*NB. You will need to update both the Master & the Slave (normally the slave is updated first followed by the master). In some cases you may need to reboot or do a service httpd restart to get online update to recognize a DNS change.*

## Fallback Page

This section allows you to view and modify the local holding page on the load balancer. If you have a master and slave load balancer then you must change this on both servers. The fallback server on the load balancer is an implementation of NGINX.

### MAINTENANCE > FALLBACK PAGE

```
<html>
<head>
<title>The page is temporarily unavailable</title>
<style>
body { font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body bgcolor="white" text="black">
<table width="100%" height="100%">
<tr>
<td align="center" valign="middle">
The page you are looking for is temporarily unavailable.<br/>
Please try again later.<br/>
(port reminder 9080)
</td>
</tr>
</table>
```



#### Layer 4

The fallback page is displayed when all real servers fail. The fallback page is NOT displayed when servers are taken offline manually via the WUI.

At layer 4, to cause the fallback page to be displayed when real servers are taken offline, you would also need to force the real server to fail its health check by for example disabling the relevant service on the real server.

#### Layer 7

For layer 7 VIPs the fallback page is displayed when all real servers are unavailable AND when all are taken offline via the WUI. The page can be hosted on the load balancer or on an external server. Set the Fallback Server option of the VIP accordingly.



For more details on Fallback server settings , please refer to page 106

## Firewall Script

This form allows you to directly edit /etc/rc.d/rc.firewall.

### MAINTENANCE > FIREWALL SCRIPT

```
#!/bin/sh

# $Id: rc.firewall 1341 2011-04-14 11:29:17Z nick $
# Firewall script for Load Balancer

##### SETUP #####
# Allow unlimited traffic on the loopback interface
#iptables -A INPUT -i lo -j ACCEPT
#iptables -A OUTPUT -o lo -j ACCEPT

##### NAT MODE #####
```

Update



**WARNING: BE CAREFUL!** Make a backup before changing this script so that you know you can roll everything back if you cause a problem.

If you wish to clear the firewall tables completely use the following command from the console:

```
/etc/rc.d/rc.flush-iptables
```

This can either be used for belt & braces security; for example to replicate your normal firewall settings onto the load balancer as well for double security. What kind of settings? Well normally you don't want any customers to be able to access the administration IP address on the load balancers, you only want them to have access to, say port 80 & 443 on the VIP interface.

You can also use the firewall script to group ports together using Firewall Marks (see Section E).

If you are planning to use NAT mode you may also want to use the load balancer as your main firewall which is fine, but we think it is a lot simpler to keep your firewall separate from your load balancer. Especially if you want to set up VPNs etc.

## Firewall Lock Down Wizard

The firewall lock down wizard automatically configures the load balancer to allow access to the various admin ports from one specific IP address. The wizard automatically detects the IP of the client running the WUI and inserts this into the Admin IP field. The default mask is set to 255.255.255.255. If you need to specify an administration network, change the mask as required.

The lock down wizard will allow full access to all the defined VIPs and reply traffic from the defined real servers.

The generated script is stored here: `/etc/rc.d/rc.lockdownwizard`

This script is activated at the end of the `/etc/rc.d/rc.firewall` script.

Any changes that you have already made to the `/etc/rc.d/rc.firewall` script are kept in place.

**MAINTENANCE > FIREWALL LOCK DOWN WIZARD**

**Warning:**  
This will block all access to the load balancer unless it matches the Admin IP or a Virtual Service IP

Admin IP: 192.168.2.7

Admin Network: 255.255.255.255

Firewall Lock Down Wizard

[ Modify the firewall lockdown wizard script ]

**Admin IP** – define the IP address of the management computer. This is auto-configured to be the IP address of the computer used to run the WUI. This can also be defined as a network address.

**Admin network** – define the netmask. Can be changed if a management range is required.

**Firewall Lock Down Wizard** – click this button to auto-generate the script.

**Modify the firewall lockdown wizard script** – use this option to edit the auto-generated script.

*NB. If you accidentally block your own access to the appliance you will need to clear the current firewall rules and try again.*

If you wish to clear the firewall tables completely use the following command from the console:

```
/etc/rc.d/rc.flush-iptables
```



**IMPORTANT!** The firewall lockdown wizard should only be run after the load balancer is fully configured and tested. Also, if changes are made later to the load balanced services, the wizard should be re-run to ensure these changes are reflected in the lockdown script.

## Initialize Graphs

This option will construct a series of RRDTool databases and relevant cron jobs to update those databases using the output from LVSGSP. More cron jobs are then used to generate the daily, weekly, monthly and yearly charts accessible from the reports section.

This option should be run after you have configured your Virtual & Real servers. If you later add additional VIPs or RIPv6s you will need to re-run this again.



**WARNING:** All current statistics will be lost when this function is used

## Users & Passwords

This section is used to manage user accounts that have access to the web based administration system.



When using a clustered pair, security changes are not automatically replicated so changes must be made on both master and slave

The default usernames and passwords, their default group membership and their primary use are as follows:

Username	Default Password	Default Group	Use (for full permission details see the table below)
loadbalancer	loadbalancer	config *	appliance administration account
reportuser	reportuser	report	viewing the appliance configuration, reports & logs
maintuser	maintuser	maint	same as reportuser, can also take servers on/off line & create the support download archive file

\* It's not possible to change the default group for user 'loadbalancer'

**NOTE:** These are simple apache .htaccess style accounts and are not related to the local Linux accounts.

The permissions for each group are as follows:

	Menu / Access						
Group	System Overview	View configuration	Edit Configuration	Maintenance	Reports	Logs	Support
config	Full	View	Full	Full	Full	View	Full
report	View	View	None	None	Full	View	View
maint	Full	View	None	None	Full	View	Full

## Resetting Passwords

To do this you'll need to run root access to the console or terminal session. e.g. to change the password for user 'loadbalancer' use the following command :

```
htpasswd -b /etc/loadbalancer.org/passwords loadbalancer <new password>
```



**SECURITY:** Don't forget to change your root password from the console using the passwd command!

## Adding New Users

New users can be added using the **Add New User** option:

### MAINTENANCE > ADD NEW USER

Username *	<input type="text"/>
Password *	<input type="password"/>
<input type="button" value="Add New User"/>	

- Simply type in the required Username & Password and click **Add New User**
- By default, new users will be added to the report group (least privilege). To change this, click **[Modify]** next to the user, select the required group and click **Edit User**

## Reports

Used to display a variety of reports for monitoring the system.

### Layer 4 Status

This live report shows the current weight and number of active & inactive connections for each real server. If a real server has failed a health check, it will not be listed.

Use the *Logs > Layer 4* option to view the *ldirectord* log file if expected servers are not listed.

### Layer 7 Status

This report is provided by the stats instance of HAProxy. This web page contains the current live status of all of the configured layer 7 HAProxy virtual and real servers.

Log in using: **Username:** loadbalancer  
**Password:** loadbalancer

NOTE: This password can be changed using *Edit Configuration > Layer 7 – Advanced Configuration*

## HAProxy version 1.5-dev6-lb1, released 2011/04/08

### Statistics Report for pid 7619

#### > General process information

pid = 7619 (process #1, nbproc = 1)  
uptime = 0d 0h07m04s  
system limits: memmax = unlimited; ulimit-n = 81000  
maxsock = 80014; maxconn = 40000; maxpipes = 0  
current conns = 1; current pipes = 0/0  
Running tasks: 1/2

active UP  
active UP, going down  
active DOWN, going up  
active or backup DOWN  
active or backup DOWN for maintenance (MAINT)  
backup UP  
backup UP, going down  
backup DOWN, going up  
not checked

Display option:

- [Hide 'DOWN' servers](#)
- [Refresh now](#)
- [CSV export](#)

External resources:

- [Primary site](#)
- [Updates \(v1.5\)](#)
- [Online manual](#)

Note: UP with load-balancing disabled is reported as "NOLB".

cluster1	Queue		Session rate		Sessions				Bytes		Denied	Errors		Warnings	Server															
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend				0	1	-	0	1	40 000	70		0	13 090	0	0	70					OPEN									
backup	0	0	-	0	0		0	0	-	0	0	0	0			0	0	0	0	0			1	-	Y					-
real_server1	0	0	-	0	0		0	0	-	0	0	0	0	0	0	0	0	0	0	5m15s UP	L4OK in 0ms	1	Y	-	0	1	1m47s		-	
Backend	0	0		0	0		0	0	40 000	0	0	0	13 090	0	0		0	0	0	7m4s UP		1	1	1		0	0s			

stats																													
	Queue			Session rate			Sessions					Bytes		Denied		Errors		Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend			1	3	-	1	1	2 000	7		3 334	63 710	0	0	0						OPEN								
Backend	0	0	0	0	0	0	0	2 000	0	0	3 334	63 710	0	0	0	0	0	0	0	0	7m4s UP		0	0	0		0		

## Layer 4 Traffic Rate

This report shows the current connections per second and bytes per second to each real server. If a real server has failed a health check, it will not be listed.

## Layer 4 traffic Counters

This report shows the volume of traffic to each real server since the counters were last re-set. If a real server has failed a health check, it will not be listed.

## Layer 4 Current Connections

The current connections report is very useful for diagnosing issues with routing or ARP related problems. In the example below, the state is shown as SYN\_RECV, this is normally a good indication that the ARP problem has not been solved.

### REPORTS > LAYER 4 CURRENT CONNECTIONS

```
IPVS connection entries
pro expire state      source          virtual         destination
TCP 00:51 SYN_RECV    192.168.2.7:64763 192.168.2.109:80 192.168.2.99:80
```

## Layer 4 Current Connections (resolve hostnames)

This is the same as the current connections report but is slower as it looks up the DNS name of each IP address.

*NOTE: These reports are generated in real time. Direct Routing is the default load balancing method and you will not see any stats for return packets (as they do not pass through the load balancer). You would however see them if you were in NAT mode.*

## Graphing

When first run, you'll be prompted to Initialize Graphs. This creates the required database files that are used to store the data that is used to produce the graphs. Once initialized, you are presented with the following options:

**GRAPHING**

**Interface Throughput Reports**  
Show Throughput ☒ Hide Throughput

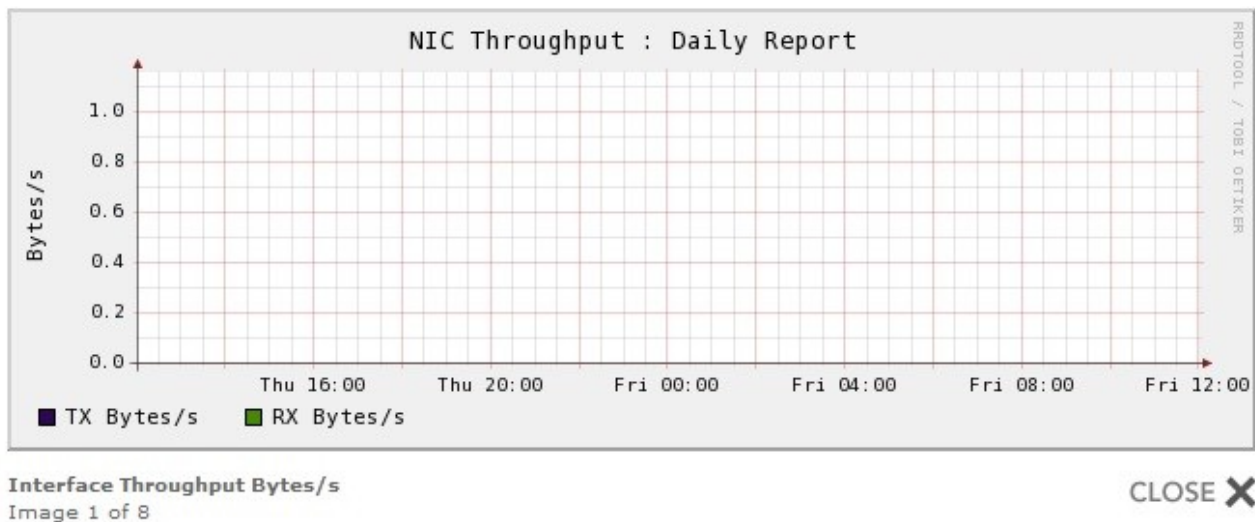
**Daily Reports**  
Show Daily ☒ Hide Daily

**Weekly Reports**  
Show Weekly ☐ Hide Weekly

**Monthly Reports**  
Show Monthly ☐ Hide Monthly

**Yearly Reports**  
Show Yearly ☐ Hide Yearly

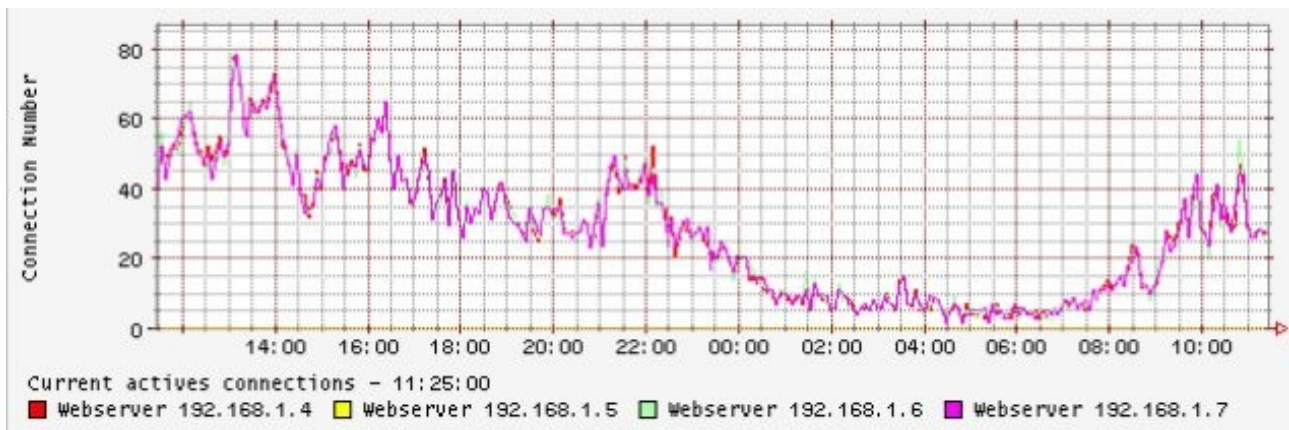
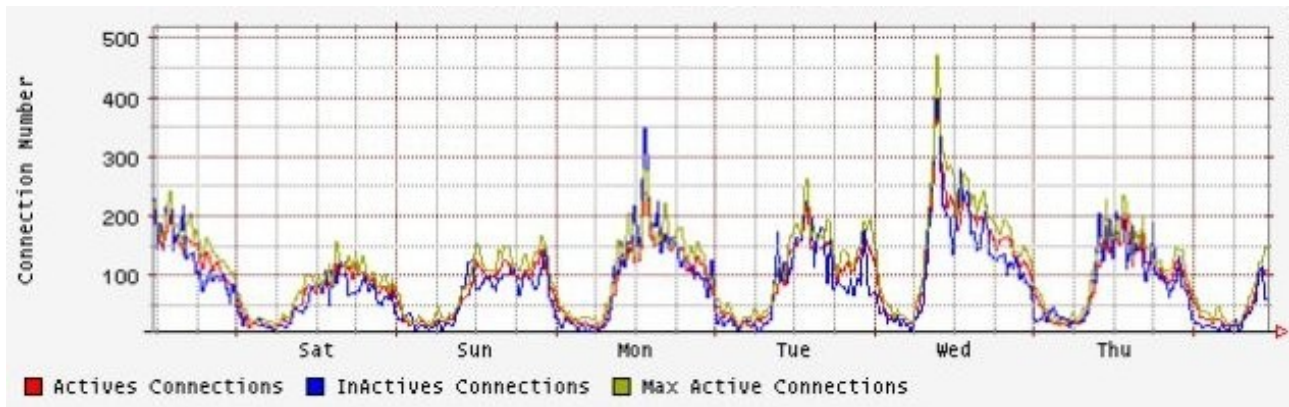
Once graph data starts to accumulate, the graph links will change from a Loadbalancer.org logo icon to a graph icon. This can then be clicked to open that particular graph.



*Why does the average activity get lower over time?*

There is a good mathematical reason for this, but the graphs now also show max connections as well as average connections.





### *Reset Packet Counters*

Resets the packet counters to zero for the load balancer reports.

## Logs

### *Load balancer*

The Lbadmin log shows all changes made via the admin system. Very useful to track all changes to the configuration.

### *Layer 4*

The ldirectord log shows the output from the health checking daemon. This is useful for checking how healthy your real servers are or pinning down any configuration errors. The logging here can be quite verbose but it clearly shows what the health checking daemon is doing.

### *Layer 7*

If activated via *Edit Configuration > Layer 7 – Advanced Configuration*, this will show the contents of `/var/log/haproxy`. This is a very detailed log of all HAProxy transactions.

### *SSL Termination*

If activated via *Edit Configuration > SSL – Advanced Configuration*, this will show the contents of `/var/log/poundssl`. This is a very detailed log of all Pound SSL transactions.

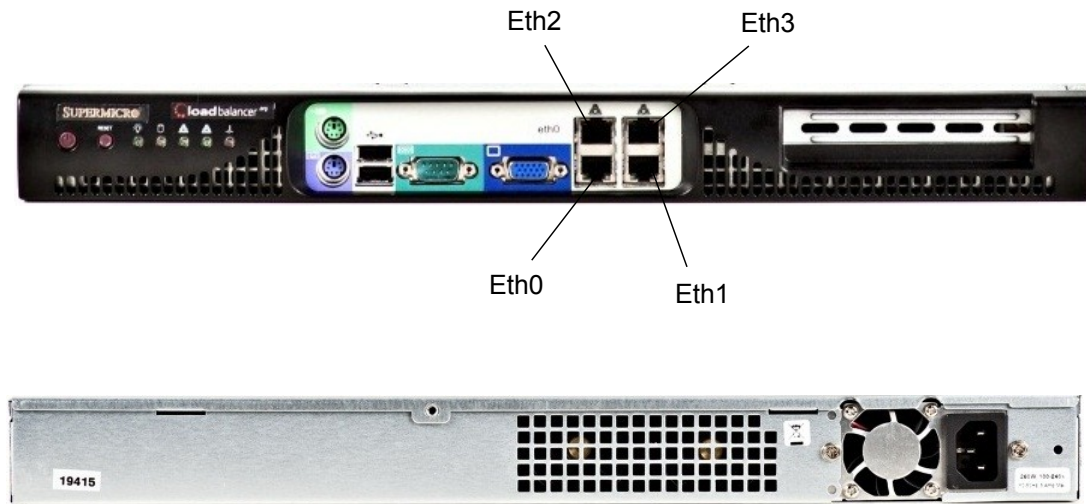
### *Heartbeat*

The heartbeat log shows the status of the heartbeat daemons. Heartbeat is used whether configured as a single device or as a clustered pair. The log provides a detailed real-time status of heartbeat.

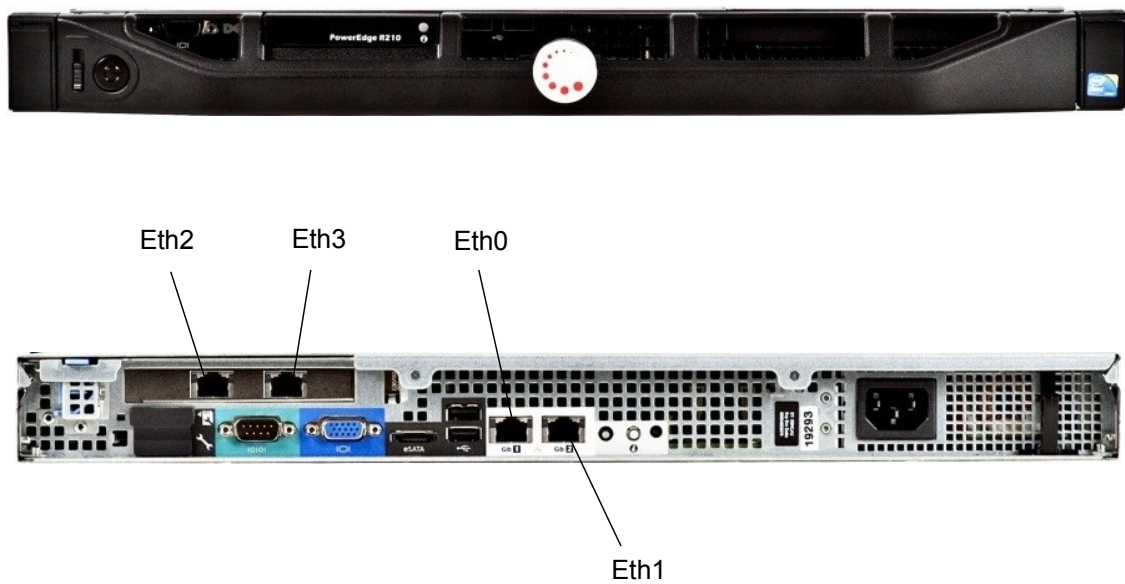
## **Section H – Appendix**

## Front & Rear Panel Layouts

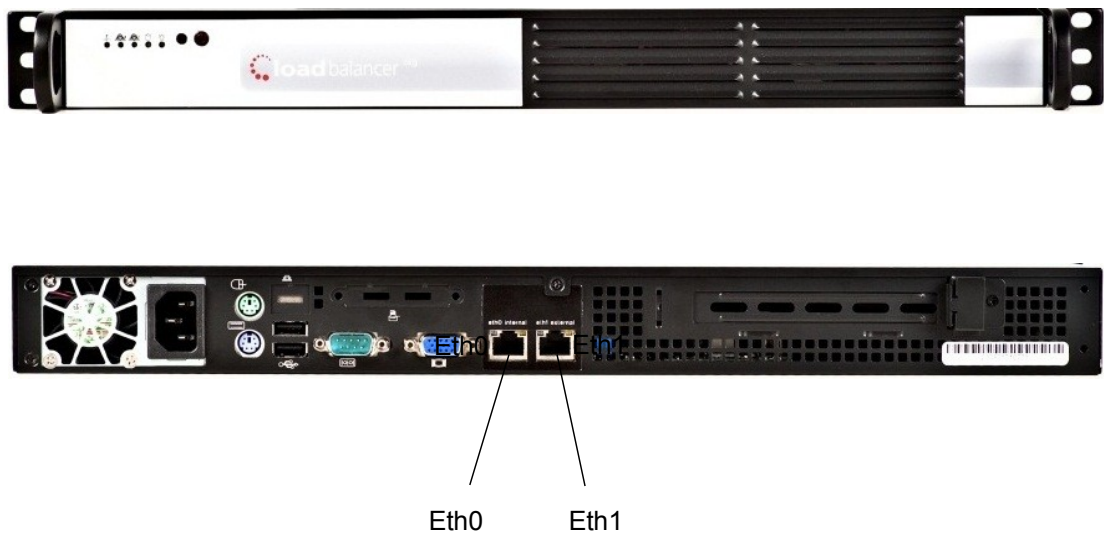
### Enterprise Max – Supermicro



### Enterprise Max / 10G – Dell



Enterprise / Enterprise R16 – Supermicro



Enterprise / Enterprise R16 – Dell

