



Appliance Administration Manual

v7.5

rev. 1.0.24



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

Copyright © 2002 – 2014 Loadbalancer.org, Inc.

Table of Contents

Chapter 1 – Introduction	8
About the Appliance.....	9
Version 7.5.x.....	9
Appliance Configuration.....	9
Deployment Guides.....	10
About this Manual.....	10
Additional Information.....	10
Chapter 2 – Load Balancing Concepts	11
Load Balancing – The Basics.....	12
Supported Protocols.....	12
OSI Layers.....	12
Load Balancing Algorithms.....	12
Weighted Round Robin.....	12
Weighted Least Connection.....	12
Destination Hashing.....	12
Real Server Agent.....	12
Layer 4 vs Layer 7.....	13
Our Recommendation.....	13
Loadbalancer.org Terminology.....	14
Chapter 3 – Load Balancing Methods	15
Supported Methods.....	16
One-Arm and Two-Arm Configurations.....	16
Direct Routing (DR).....	17
Network Address Translation (NAT).....	18
NAT Mode Packet re-Writing.....	19
Source Network Address Translation (SNAT)	20
Other Considerations.....	21
Does Your Application Cluster correctly Handle its own State?.....	21
Replication Solutions for Shared Data.....	21
Solutions for Session Data.....	21
Persistence (aka Affinity).....	21
What do You do if Your Application is not Stateless?.....	22
Loadbalancer.org Persistence Options.....	22
Which Load Balancing Method should I Use?.....	23
Our Recommendation.....	23
Chapter 4 – Appliance Fundamentals	24
The Hardware Appliance – Unpacking and Connecting.....	25
The Virtual Appliance – Hypervisor Deployment.....	26
Supported Hypervisors.....	26
Host Requirements.....	26
Downloading the Appliance.....	26
VMware Deployment.....	27
Hyper-V Deployment.....	27
Initial Network Interface Configuration.....	28
Using the Network Setup Wizard.....	28
Using Linux Commands.....	29
Appliance Access & Configuration Methods.....	30
Local Methods.....	30
Console Access.....	30
Appliance Configuration using Links.....	30
Keyboard Layout.....	30
Remote Methods.....	31
Accessing the WUI.....	31
Configuring the Appliance using the Wizard.....	32
Running the Wizard.....	32
Configuration the Appliance using the WUI.....	33
Main Menu Options.....	33
Full Root Access.....	34

Appliance Configuration Files & Locations.....	34
Chapter 5 – Appliance Management.....	35
Network Configuration.....	36
Physical Interfaces.....	36
Configuring IP Addresses.....	36
Configuring Bonding.....	37
Bonding Configuration Modes.....	38
Bonding for High-Availability (default mode).....	38
Bonding for Bandwidth.....	38
Bonding for High-Availability & Bandwidth.....	38
Configuring VLANs.....	39
Configuring Default Gateway & Static Routes.....	40
Configuring Hostname & DNS Configuration.....	41
System Date & Time and NTP Server Configuration.....	42
Auto Configuration using NTP Servers.....	42
Manual Configuration.....	42
Appliance Internet Access via Proxy.....	43
SMTP Relay Configuration.....	43
Syslog Server Configuration.....	44
Appliance Upgrade (Enterprise R16 to Enterprise).....	44
Running OS Level Commands.....	44
Restoring Manufacturer's Settings.....	45
Using the WUI.....	45
Using the Console / SSH Session.....	45
Restarting Services.....	45
Appliance Restart & Shutdown.....	47
Appliance Software Updates.....	47
Checking the Current Software Version & Revision.....	47
Online Update.....	47
Offline Update.....	49
Updating a Clustered Pair.....	50
Firewall Configuration.....	50
Manual Firewall Configuration.....	51
Firewall Lock-down Wizard.....	52
Contrack Table Size.....	53
Users & Passwords.....	53
Appliance Security Lockdown Script.....	55
Chapter 6 – Configuring Load Balanced Services.....	56
Layer 4 Services.....	57
The Basics.....	57
Creating Virtual Services (VIPs).....	57
Modifying a Virtual Service.....	58
Creating Real Servers (RIPs).....	61
Persistence Considerations.....	62
Persistence State Table Replication.....	62
DR Mode Considerations.....	62
What Is the ARP Problem?.....	62
Detecting the ARP Problem.....	63
Resolving ARP Issues for Linux.....	63
Method 1 (using iptables).....	63
Method 2 (using arp_ignore sysctl values).....	63
Resolving ARP issues for Solaris.....	64
Resolving ARP issues for Mac OS X / BSD.....	64
Resolving ARP issues for Windows Servers.....	65
Windows Server 2000.....	65
Windows Server 2003.....	68
Windows server 2008.....	71
Windows Server 2012.....	74
Verifying netsh Settings for Windows 2008 & 2012.....	77
Configuring IIS to Respond to both the RIP and VIP.....	78
Windows Firewall Settings.....	80
NAT Mode Considerations.....	82
NAT Mode Potential Issues.....	82
Enabling Real Server Internet access using Auto-NAT.....	82

Enabling Access to non Load-Balanced Services.....	82
One-Arm (Single Subnet) NAT Mode.....	83
Route Configuration for Windows Servers.....	83
Route Configuration for Linux Servers.....	84
Firewall Marks.....	84
Firewall Marks – Auto Configuration.....	84
Firewall Marks – Manual Configuration.....	85
Layer 4 – Advanced Configuration.....	90
Layer 7 Services.....	92
The Basics.....	92
Creating Virtual Services (VIPs).....	92
Modifying a Virtual Service.....	93
Creating Real Servers (RIPs).....	95
Persistence Considerations.....	96
Persistence State Table Replication.....	96
Layer 7 – Custom Configurations.....	96
Ex. 1 – Load Balancing based on URL match using ACL's in HAProxy.....	97
Ex. 2 – HTTP to HTTPS Redirect using HAProxy & Pound (SSL Termination on the Appliance).....	98
Ex. 3 – HTTP to HTTPS Redirect using HAProxy (SSL Termination on the Real Server).....	99
HAProxy Error Codes.....	100
Layer 7 – Advanced Configuration.....	101
SSL Termination.....	104
Concepts.....	104
SSL Termination on the Real Servers.....	105
SSL Termination on the Load Balancer.....	105
Creating an STunnel SSL Virtual Service (the Default SSL Terminator).....	105
STunnel Cipher Settings and the BEAST Attack.....	107
Creating a Pound SSL Virtual Service.....	108
Modifying a Pound SSL Virtual Service.....	110
Pound Cipher Settings and the BEAST Attack.....	110
Generating a CSR on the Load Balancer.....	110
Using an Existing Certificate.....	112
Creating a PEM file & Uploading to the Appliance.....	112
Adding an Intermediate Certificate.....	113
Converting between certificate formats.....	114
Converting .pfx certificates to PEM format.....	114
Converting .cer certificates to PEM format.....	114
Converting an Encrypted Private Key to an Unencrypted Key.....	115
SSL – Advanced Configuration.....	115
Floating IPs.....	116
Using Transparent Proxy (TProxy).....	117
TProxy & HAProxy.....	117
TProxy, HAProxy & Pound.....	118
TProxy, HAProxy & STunnel.....	119
Server Feedback Agent.....	120
Windows Agent.....	120
Linux / Unix Agent.....	122
Custom HTTP Agent.....	123
Configuring VIPs & RIPs via Command Line / Script.....	124
Layer 4.....	124
Layer 7.....	125
Chapter 7 – Real Server Health Monitoring.....	126
Introduction.....	127
Real Server health Monitoring – Using System Overview.....	127
Layer 4 Services.....	128
Layer 7 Services.....	132
Simulating Health-Check Failures.....	133
Fallback Server Settings.....	134
Chapter 8 – Appliance Clustering for HA.....	136
Introduction.....	137
Clustered Pair Considerations.....	137
Master / Slave Operation.....	137
Heartbeat.....	137
Master Slave Replication.....	137

Settings that are NOT Replicated.....	137
Configuring Heartbeat.....	138
Adding a Slave Unit after the Master has been Configured.....	140
Clustered Pair Diagnostics.....	141
Heartbeat State Diagnostics.....	141
Split Brain Scenarios.....	142
Forcing Master / Slave Failover & Failback.....	143
Testing & Verifying Master / Slave Replication & Failover.....	143
Chapter 9 – Application Specific Settings.....	146
FTP.....	147
Layer 4 Virtual Services for FTP.....	147
FTP Layer 4 Negotiate Health Check.....	147
FTP Recommended Persistence Settings.....	148
Layer 7 Virtual Services for FTP.....	148
Active Mode.....	148
Windows 2008 Example.....	149
Passive Mode.....	150
Windows 2008 Example.....	150
Limiting Passive FTP Ports.....	152
For Windows 2008.....	152
For Windows 2003.....	153
For Windows 2000.....	153
For Linux.....	153
Terminal Services / Remote Desktop Services & RDP.....	154
Layer 4 – IP Persistence.....	154
Layer 7 – Microsoft Connection Broker / Session Directory.....	154
Layer 7 – RDP Cookies.....	155
Other Applications.....	155
Chapter 10 – Configuration Examples.....	156
Introduction.....	157
Initial Network Settings.....	157
Example 1 – One-Arm DR Mode (Single Appliance).....	157
Configuration Overview.....	157
Network Settings.....	157
N.B. this step can be skipped if all network settings have already been configured.....	157
Virtual Service (VIP).....	158
Real Servers (RIPs).....	159
Real Server Changes – Solve the ARP Problem.....	159
Basic Testing & Verification.....	160
Example 2 – Two-Arm NAT Mode (Clustered Pair).....	161
Configuration Overview.....	161
Master Unit – Network Settings.....	161
Slave Unit – Network Settings.....	162
Master Unit – Heartbeat Settings.....	164
Checking the Status.....	165
Virtual Service (VIP).....	165
Real Servers (RIP).....	166
Real Server Changes – Set the Default Gateway.....	166
Verify the Slave Configuration.....	167
Basic Testing & Verification.....	167
Example 3 – One-Arm SNAT Mode & SSL Termination (Single Appliance).....	168
Configuration Overview.....	168
Network Settings.....	168
Virtual Service (VIP).....	170
Real Servers (RIP).....	170
SSL Termination.....	171
Basic Testing & Verification.....	172
Chapter 11 – Testing Load Balanced Services.....	173
Testing Load Balanced Services.....	174
Connection Error Diagnosis.....	174
System Overview.....	175
Using Log Files.....	176
Using Reports.....	176

Chapter 12 – Appliance Monitoring	177
Appliance Log Files.....	178
Load Balancer.....	178
Layer 4.....	178
Layer 7.....	178
SSL Termination (Pound).....	178
SSL Termination (STunnel).....	178
Heartbeat.....	178
Appliance Reports.....	179
Layer 4 Status.....	179
Layer 4 Traffic Rate.....	179
Layer 4 traffic Counters.....	180
Layer 4 Current Connections.....	180
Layer 4 Current Connections (resolve hostnames).....	180
Layer 7 Status.....	181
Layer 7 Stick Table.....	181
Graphing.....	182
Graphs – Load Balanced Services.....	182
Graphs – Appliance Specific.....	184
Graph Options.....	185
SNMP Reporting.....	187
SNMP for Layer 4 Based Services.....	187
Monitoring Layer 4 RIPs using SNMP.....	187
SNMP for Layer 7 Based Services.....	188
Monitoring Layer 7 RIPs using SNMP.....	188
Configuring Email Alerts.....	189
Global Alerts.....	189
VIP Level Alerts.....	190
Chapter 13 – Useful Tools & Utilities	191
Useful Diagnostics Tools.....	192
Netstat.....	192
Telnet.....	192
Tcpdump.....	193
Ethtool.....	193
Wireshark.....	194
Windows Specific Tools.....	194
WinSCP.....	194
PuTTY.....	194
Remote Support Tools.....	194
Chapter 14 – Backup & Restore and Disaster Recovery	195
Introduction.....	196
Backup & Restore.....	196
Restoring XML Files.....	197
Disaster Recovery.....	198
Being Prepared.....	198
Backing Up to a Remote Location.....	198
Using wget to Copy the Files.....	198
Backing up locally on the Load Balancer.....	199
Appliance Recovery using a USB Memory Stick.....	199
Disaster Recovery After Master Failure.....	202
Disaster Recovery After Slave Failure.....	204
Option 1 – Using the XML Backup.....	204
Option 2 – Synchronizing From the Master.....	205
Chapter 15 – Technical Support	206
Introduction.....	207
WUI Support Options.....	207
Contact Us.....	207
Technical Support Download.....	208
Appendix	209
Company Contact Information.....	210
Front & Rear Panel Layouts.....	211

Chapter 1 – Introduction

About the Appliance

The Loadbalancer.org appliance is an Intel based server running the GNU/Linux operating system with a custom kernel configured for load balancing.

The core software is based on customized versions of Centos 6 / RHEL 6, Linux 2.6, LVS, HA-Linux, HAProxy, Pound, STunnel & Ldirectord. Full root access is provided which enables complete control of all settings.

Appliances can be deployed as single units or in HA pairs, although Loadbalancer.org always recommend that HA pairs should be used for high availability and resilience.

Version 7.5.x

The latest version delivers an updated user interface, several new features as well as improvements to others.

A quick summary:

- Upgraded WUI with new menu options and improved grouping of sub-options
- Completely re-written graphing system
- Support for STunnel has been added
- Full NTP support has been added
- Improved master / slave role status display
- Enhanced master / slave synchronization and service control
- Simplified HA recovery process
- Enhanced data validation checks

Appliance Configuration

Initial network configuration can be carried out on the console by using the Network Setup Wizard or standard Linux network setup commands, or by connecting to the default IP address:port (192.168.2.21:9080) and making changes using the WUI.

Once the network is configured, the appliance can be configured manually or by using the Setup Wizard. The WUI is accessible using HTTP on port 9080 and HTTPS on port 9443. It's also possible to configure the load balancer at the console using the text based Links browser, although using the WUI is the recommended method.

For a clustered pair the slave device must be defined on the master. All configuration must be carried out on the master unit, the slave is then automatically kept in-sync as changes are made on the master.

Deployment Guides

Deployment guides have also been written that focus on load balancing specific applications. An up to date listing is available on the solutions page of our website: <http://www.loadbalancer.org/solutions.php>.

At the time of writing, the following deployment & quick-reference guides are available:

- [Load Balancing Microsoft IIS Web Servers](#)
- [Load Balancing Microsoft Terminal Services](#)
- [Load Balancing Microsoft Exchange 2010](#)
- [Load Balancing Microsoft Exchange 2013](#)
- [Load Balancing Microsoft Sharepoint 2010](#)
- [Load Balancing VMware View](#)
- [Load Balancing Microsoft OCS 2007 R2](#)
- [Load Balancing Microsoft Lync 2010](#)
- [Load Balancing Web Proxies / Filters](#)

About this Manual

This document covers all required administration information for v7.5.x Loadbalancer.org appliances.

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 our support team at: support@loadbalancer.org.

Chapter 2 – Load Balancing Concepts

Load Balancing – The Basics

Loadbalancer.org appliances enable two or more servers to be combined into a cluster. This enables inbound requests to be distributed across multiple servers which provides improved performance, reliability and resilience. Appliances can also be deployed as a clustered pair (our recommended solution) which creates a highly-available configuration.

Supported Protocols

Loadbalancer.org appliances support virtually any TCP or UDP based protocol including HTTP, HTTPS, FTP, SMTP, RDP, SIP, IMAP, POP, DNS etc.

OSI Layers

Load balancing at layer 4 and layer 7 is supported. LVS (*Linux Virtual Service*) is utilized at layer 4 whilst HAProxy is used at layer 7.

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.

Real Server Agent

To compliment the methods above, Loadbalancer.org appliances also support Real Server (i.e back-end server) agents. This permits the load balancing algorithm to be dynamically modified based on each Real Servers running characteristics. For example, one Real Server could have a run-away process that is consuming excessive CPU resources. Without the agent, the load balancer would have no way of knowing

this and would continue to send requests to the overloaded server based on the algorithm selected. With the agent installed on the Real Server, feedback is provided to the load balancer and the algorithm is then adjusted to reduce requests that are sent to that server. For more details on using the agent please refer to page 120.

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, HTTPS, DNS, RDP etc. Therefore the load balancer can only make load balancing decisions based on details available at layers 4 and below such as port numbers and IP addresses. At layer 7, the load balancer 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 (aka affinity or 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 on the web server rather than in a separate database. At Layer 4, Source IP persistence is the only option. At layer 7, additional methods are available such as HTTP cookie persistence where the load balancer sets a cookie to identify the session and Microsoft Connection Broker where the load balancer is able to utilize the redirection token for reconnecting users to existing sessions.

Real Server Changes

At Layer 4, either the ARP problem (please refer to pages 62-81 for more details) has to be solved (required when using Layer4 DR mode) or the default gateway on the Real Servers must be set to point at the load balancer (required when using Layer 4 NAT mode). 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 pages 117-119 and 95 respectively).

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.

Loadbalancer.org Terminology

Acronym	Definition
Load Balancer	An IP based traffic manager for server clusters
VIP	The Virtual IP address that a cluster is contactable on (Virtual Server/Service) <i>N.B. Prior to v7.5 a VIP is known as a 'Virtual Server', from v7.5 onwards it's known as a 'Virtual Service'</i>
RIP	The Real IP address of a back-end server in the cluster (Real Server)
GW	The Default Gateway for a back-end server in the cluster
WUI	Web User Interface
Floating IP	An IP address shared by the master & slave load balancer when in a high-availability configuration (shared IP)
Layer 4	Part of the seven layer OSI model, descriptive term for a network device that can route packets based on TCP/IP header information
Layer 7	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
DR	Direct Routing is a standard load balancing technique that distributes packets by altering only the destination MAC address of the packet
NAT	Network Address Translation – Standard load balancing technique that changes the destination of packets to and from the VIP (external subnet to internal cluster subnet)
SNAT (<i>HAProxy</i>)	Source Network Address Translation – Load balancer acts as a proxy for all incoming & outgoing traffic
SSL Termination (<i>Pound & STunnel</i>)	The SSL certificate is installed on the load balancer in order to decrypt HTTPS traffic on behalf of the cluster
MASQUERADE	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
One Arm	The load balancer has one physical network card connected to one subnet
Two Arm	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
Eth0	Usually the internal interface also known as Gb0
Eth1	Usually the external interface also known as Gb1

Chapter 3 – Load Balancing Methods

Supported Methods

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 (<i>Pound & STunnel</i>)	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 several Microsoft applications such as Exchange, Sharepoint, Terminal Services (Connection Broker & RDP Cookie persistence) that use SNAT mode
- Only required for Direct Routing implementation across routed networks (rarely used)

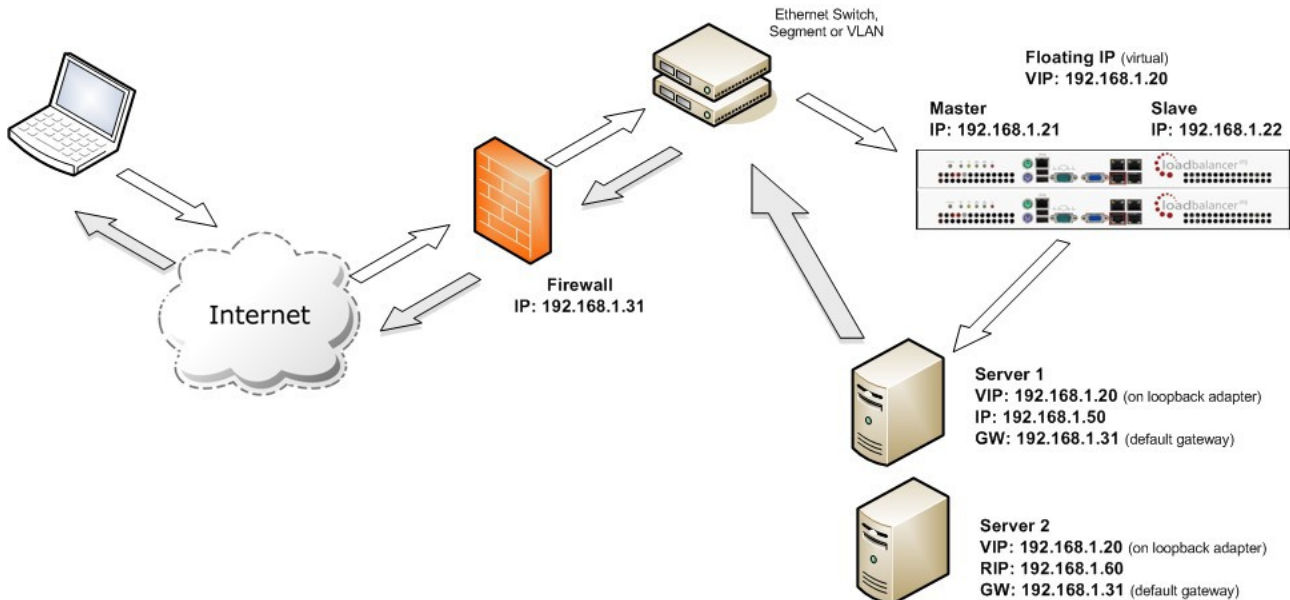
One-Arm and Two-Arm Configurations

The number of 'arms' is generally a descriptive term for how many physical connections (Ethernet interfaces) are used to connect a device to a network. It's 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.

One-Arm	The load balancer has one physical network card connected to one subnet
Two-Arm	The load balancer has two network interfaces connected to two subnets – this can be achieved by using two physical network cards or by assigning two addresses to one physical network card

Direct Routing (DR)

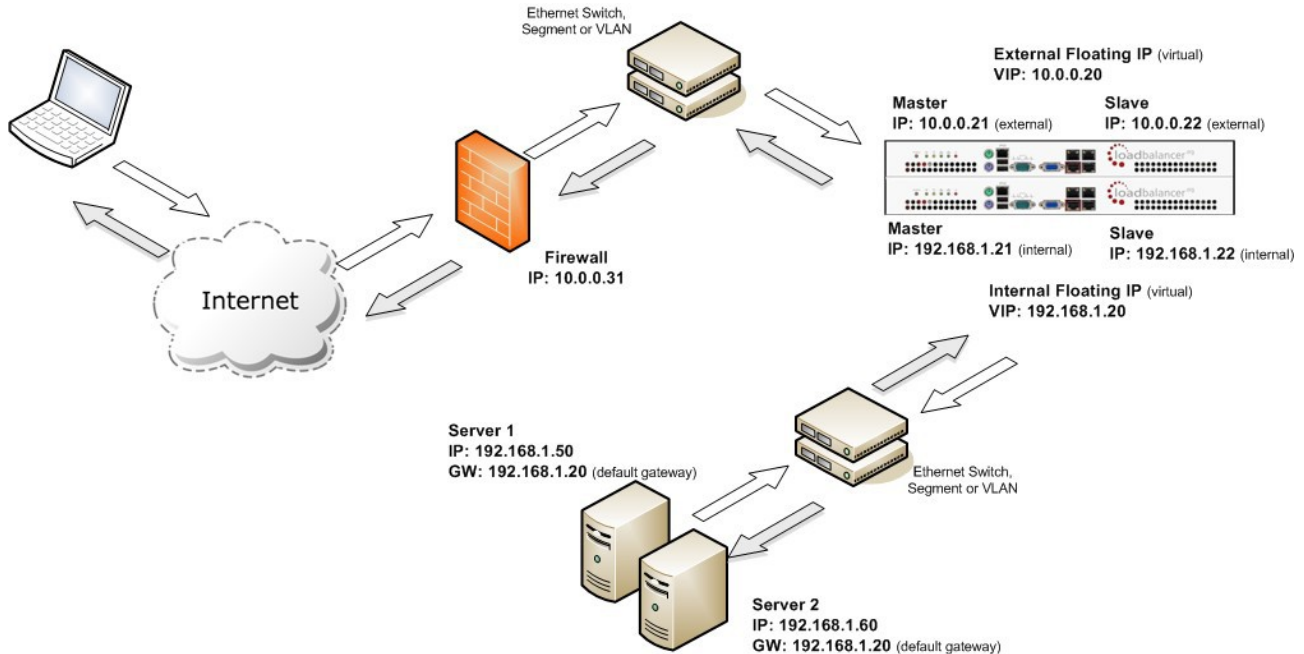
One-arm direct routing (DR) mode is a very high performance solution that requires little change to your existing infrastructure. *N.B. Brocade & A10 Networks call this Direct Server Return and F5 call it N-Path.*



- 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 62-81 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 Service 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'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. The second choice is Network Address Translation (NAT) mode. This is also a 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 Service to the internal Real Servers
- Normally eth0 is used for the *internal* network and eth1 is used for the *external* network although this is not mandatory. If the Real Servers require Internet access, Autonat should be enabled using the WUI option: *Cluster Configuration > Layer 4 – Advanced Configuration*, the external interface should be selected
- When the wizard is used, Real Servers are automatically given access to the Internet through the load balancer (via Auto-NAT)
- 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
- 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 Service 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. *N.B. It is possible to have Real and Virtual Services in the same subnet – please refer to the page 83. in the administration manual. N.B. 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 page 82 in the administration manual 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)
- Port translation is possible in NAT mode, i.e. VIP:80 → RIP8080 is allowed

NAT Mode Packet re-Writing

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

The following table shows an example NAT mode setup:

Protocol	VIP	Port	RIP	Port
TCP	10.0.0.20	80	192.168.1.50	80

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

Packet rewriting works as follows:

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

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

2) *The packet is rewritten and forwarded to the back-end server as:*

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

3) *Replies return to the load balancer as:*

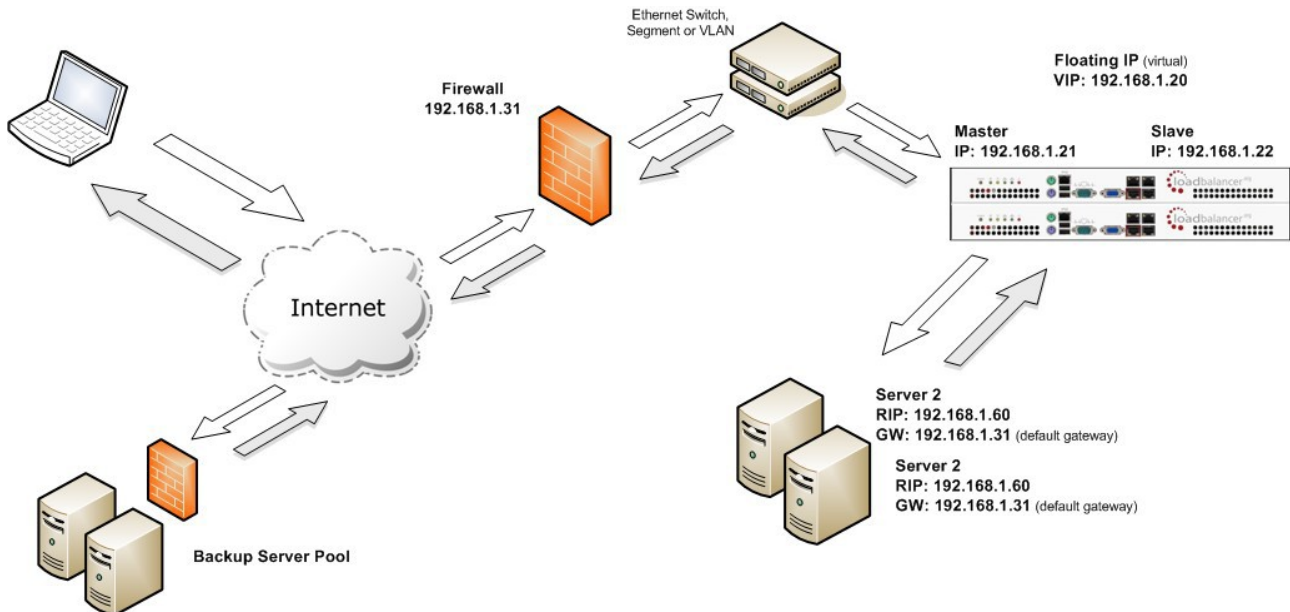
SOURCE 192.168.1.50:80 DEST x.x.x.x:3456

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

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

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 mode is also used with numerous Microsoft applications such as Exchange, Sharepoint, Lync etc.



This mode has the advantage of a one arm configuration and does not require any changes to the application servers. However, since the load balancer is acting as a full proxy it doesn't have the same raw throughput as the layer 4 methods.

The 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 load balancer, or for HTTP, using X-forwarded-For headers. Please refer to pages 117-119 and 95 respectively for more details.



For detailed configuration examples using various modes, please refer to chapter 10 starting on page 156.

Other Considerations

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 that's included by default in newer versions of Windows Server or in the resource kit for older versions. 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 (aka Affinity)

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 / Remote Desktop 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 Options

- Source IP (subnet)
- Cookie (Active or Passive)
- SSL session ID
- Microsoft Connection Broker / Session Broker Integration

The standard Layer 4 persistence method is source IP persistence, you can handle millions of persistent connections at Layer 4. Just modify your Virtual Service 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).

SSL session ID based persistence is useful in certain circumstances, although due to the way some browsers operate – notably Internet Explorer, the session ID can be renegotiated frequently which effectively breaks the persistence.

Which Load Balancing Method should I Use?

Layer 4 DR Mode offers the best performance and requires limited Real Server changes. The server application must be able to bind to the both the RIP & VIP at the same time.

Layer 4 NAT Mode is also a high performance solution but not as fast as DR mode. 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). Also each Real Server must use the load balancer as the default gateway.

Layer 7 SNAT Mode offers greater flexibility but at lower performance levels. It supports HTTP cookie insertion, RDP cookies, Session Broker integration and works very well with either Pound or STunnel when SSL termination is required. It does not require any changes to the application servers and can be deployed in one-arm or two-arm mode and. HAProxy is a high performance solution, but since it operates as a full proxy, it cannot perform as fast as the layer 4 solutions.

Our Recommendation

Where possible we recommend that Layer 4 Direct Routing (DR) mode is used. This offers the best possible performance since replies go directly from the Real Servers to the client, not via the load balancer. It's also relatively simple to implement.

Ultimately, the final choice does depend on your specific requirements and infrastructure.

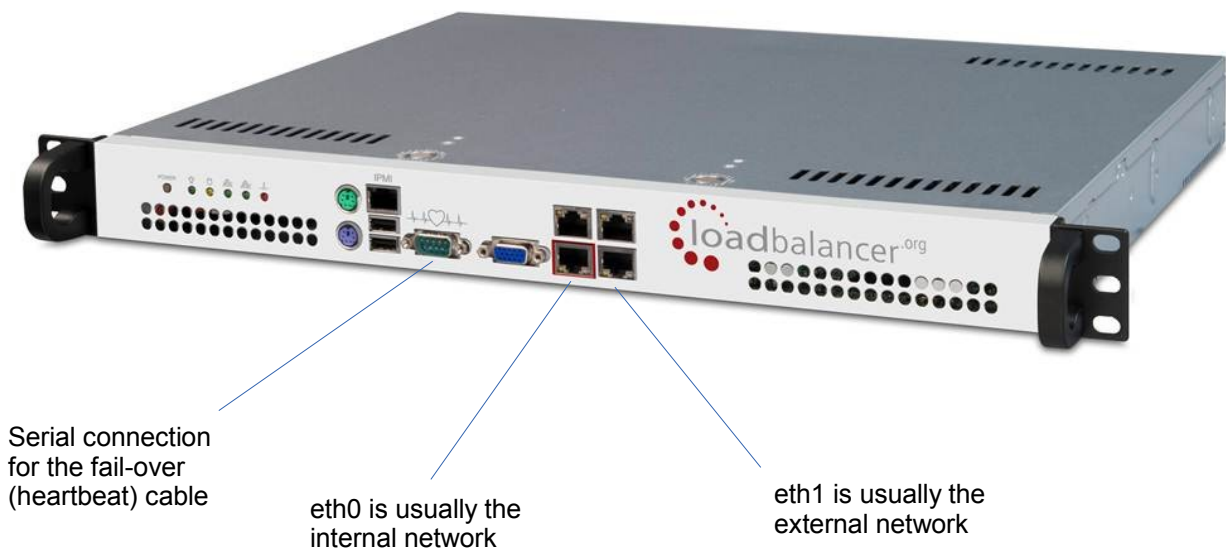


IMPORTANT NOTE – If you are using Microsoft Windows Real Servers (i.e. back-end servers) make sure that Windows NLB (Network Load Balancing) is completely disabled to ensure that this does not interfere with the operation of the load balancer.

Chapter 4 – Appliance Fundamentals

The Hardware Appliance – Unpacking and Connecting

- 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 a network cable from the switch to one of the Ethernet ports – typically *eth0* but this is not mandatory
- If using a two-armed configuration connect another cable to a second Ethernet port – typically *eth1* but this is not mandatory (N.B. the Enterprise and Enterprise R16 have 2 ports, the MAX and 10G have 4 ports)
- For a clustered hardware pair connect a serial cable (1 supplied with each appliance) between the two appliances – if this is not possible (e.g. different rack) heartbeat must be configured to use ucast over the network
- Attach a monitor to the VGA port and keyboard to the USB or PS/2 port
- Check mains power is on and press the power switch to start the appliance (the fans should start & front panel LED's should light)
- Allow a minute for booting



N.B. The above image shows the Enterprise MAX, for connecting other models please refer to the appendix.

The Virtual Appliance – Hypervisor Deployment

Supported Hypervisors

Currently, the Virtual Appliance is available for the following hypervisors:

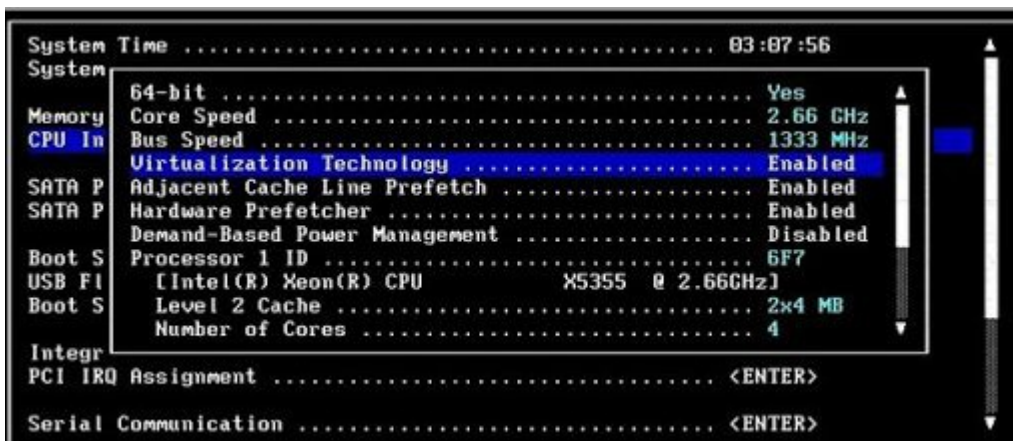
- VMware (Player/Workstation/Server & vSphere ESX/ESXi)
- Microsoft Hyper-V

Host Requirements

To run the Loadbalancer.org Enterprise VA (irrespective of which Hypervisor is being used) the following basic server specifications must be met:

- 64bit CPU
- Virtual Technology hardware support – either Intel-VT or AMD-V compliant CPU's

For an Intel based server, VT must be enabled in the BIOS as shown in the example below:



If your server is unable to support 64bit guests, an error message will be displayed when attempting to start the VA.

Downloading the Appliance

All downloads are accessible from the following location: <http://www.loadbalancer.org/downloads.php>

Once downloaded, extract the files from the .zip archive using your preferred utility. The download also includes a quickstart guide which covers the VMware and Hyper-V deployment process in more detail.

*N.B. To access the downloads you'll need to enter your name & email address, select the Hypervisor type (VMware or Hyper-V) and specify the application that you'll be load balancing. Once the required details are entered, click **Submit**, we'll then send you an email that includes the various links.*

Any information provided is 100% confidential. We may follow up with an email to see how you are getting on with the trial and offer assistance but under no circumstances will Loadbalancer.org send you other promotional material or share your information with a third party.

VMware Deployment

The exact steps depend on which VMware environment is in use. The following list provides a basic guideline:

- For vSphere Client use: **File > Deploy ovf Template**
- For Virtual Infrastructure Client use: **File > Virtual Appliance > Import**
- For VMware Server use: **Virtual Machine > Add VM to Inventory**

Hyper-V Deployment

Windows 2008 R2

1. Start Hyper-V Manager, then using the right-click menu or the Actions pane select *Import Virtual Machine* and then click **Next**
2. Browse to the location of the extracted download and select the folder LBVMHYPER-Vv7
3. Select the option "*Copy the virtual machine (create a new unique ID)*" and also select the "*Duplicate all files so the same virtual machine can be imported again*" check-box, click **Import**
4. The import will start, once complete the new appliance will appear in the Virtual Machine list
5. The appliance has 4 NIC cards, to connect these right-click the appliance and select *Settings* then for each Network Adapter select the required network
6. Right-click and select **Start** to power up the appliance, allow a minute to boot
7. If you're deploying a clustered pair, you'll first need to do one of the following steps before importing the second virtual machine. If this is not done, the second virtual machine cannot be deployed because the disk from the first import already exists, and there will therefore be a conflict:
 - i) Shutdown the first VM and modify the name of the disk
 - or
 - ii) Change the default file location using the Hyper-V *Settings* option in the *Actions* paneOnce one of the above is done, repeat steps 1-6 to create the second virtual machine.

Windows 2012

1. Start Hyper-V Manager, then using the right-click menu or the Actions pane select *Import Virtual Machine* then click **Next**
2. Browse to the location of the extracted download and select the folder LBVMHYPER-Vv7
3. Click **Next** until prompted for the Import Type, make sure that '*Copy the virtual machine (create a new unique ID)*' is selected and click **Next**
4. Tick the check-box '*Store the Virtual Machine in different location*', then define a suitable location for the virtual machines files and click **Next**
5. Define a location for the virtual hard disk files
6. Click **Next**, then click **Finish** to complete the import process. Once complete, the load balancer will appear in the Virtual Machines list
7. The appliance has 4 NIC cards, to connect these right-click the appliance and select *Settings* then for each Network Adapter select the required network
8. Highlight the new load balancer and start it either by using the right-click menu or the Actions pane

If you're deploying a clustered pair, repeat steps 2-8 for the slave unit, making sure that a different folder location is selected in steps 4 & 5.

Initial Network Interface Configuration

By default the load balancer is pre-configured with the following IP address & subnet mask:

192.168.2.21 / 255.255.255.0

This default address can be changed at the console in two ways:

- Using the built-in Network Setup Wizard
- Using traditional Linux commands

Using the Network Setup Wizard

To run the wizard, login to the console of the appliance as the 'setup' user. This is explained in the initial console start-up message as shown below:

```
Welcome to the Loadbalancer.org appliance.

To perform initial network configuration, log in to the console as
Username: setup
Password: setup

To access the web interface and wizard, point your browser at
http://192.168.2.21:9080/
or
https://192.168.2.21:9443/

lbmaster login: _
```

- login to the console:
Username: setup
Password: setup
- Once logged in, enter the IP address /mask, default gateway & DNS servers at the prompts as shown below:

```
Loadbalancer.org basic network set up

Static IP address (eg. 192.168.0.26) : 192.168.67.23/18

Default gateway (eg. 192.168.0.1) : 192.168.64.1

DNS Servers
Primary (eg. 192.168.0.250) : 192.168.64.1
Secondary (Leave blank to omit) :
```

After the required settings have been entered, a summary will be presented along with details of how to access the WUI as shown below:

```
Summary of settings
  Static IP address:      192.168.67.23/18
  Default gateway:       192.168.64.1
  DNS servers:           192.168.64.1

You may now connect the eth0 network interface to your switch, and
continue configuration through the web interface on:

      http://192.168.67.23:9080/lbadmin/

Press any key...
```

As mentioned in the text the IP address is now configured for interface eth0.

IP addresses for the other interfaces can now be configured using the WUI option: *Local Configuration > Network Interface Configuration* (to access the WUI please refer to pages 31 and 33) or by using Linux commands as explained in the following section.

Using Linux Commands

To set the IP address, login to the console or an SSH session as root:

Username: root
Password: loadbalancer

set the IP address using the following command:

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

e.g.

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

set the default gateway using the following command:

```
route add default gw <IP address> <interface>
```

e.g.

```
route add default gw 192.168.1.254 eth0
```

N.B. Setting the IP address in this way is temporary, the IP address MUST be set via the WUI to make this permanent otherwise settings will be lost after a reboot

Appliance Access & Configuration Methods

The appliance can be accessed & configured both locally and remotely.

Local Methods

Console Access

To access the console, simply connect a monitor and keyboard to the load balancer, power up and you'll be presented with a login prompt. The console can also be accessed via the serial port if heartbeat is configured to communicate over the network which is the default configuration for v7.5.

Log in to the console:

Username: root
Password: loadbalancer

Appliance Configuration using Links

Once logged into the console, the Links browser can be used to configure the appliance. To start Links and bring up the text based administration interface use the following command:

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

Log in to Links:

Username: loadbalancer
Password: loadbalancer

Use the *Up*, *Down* & *Enter* keys to move between and select the various menu options.

N.B. The preferred configuration method is the WUI which can be accessed via a browser as explained on page 31.

Keyboard Layout

To change the keyboard locale edit the file: `/etc/sysconfig/keyboard`, e.g. to change from a UK to a US layout:

1. edit `/etc/sysconfig/keyboard` using a browser such as 'vi' or 'vim' for Linux or WinSCP under Windows
2. replace `KEYTABLE="uk"` with `KEYTABLE="us"`
3. re-boot the appliance

Remote Methods

When configuring the appliance remotely, take care when changing network and firewall settings. If you do lock yourself out, you'll either need local console access or you can use remote management tools such as IPMI or iDRAC. All Supermicro based appliances include IPMI support, iDRAC is included on the Enterprise MAX & 10G and is optional on the Enterprise. For details on configuring IPMI please refer to the Appendix.

The appliance can be remotely accessed using the following tools:

- | | |
|---------------------------------------------------|--------------------------|
| • HTTP / HTTPS Web Browser | Web User Interface (WUI) |
| • OpenSSH (Linux hosts) or PuTTY (Windows hosts) | Secure Shell Access |
| • OpenSCP (Linux hosts) or WinSCP (Windows hosts) | Secure File Transfer |

Accessing the WUI

The WUI is accessed using a browser such as Firefox, Chrome etc. Appliance authentication is based on Apache .htaccess files. User admin tasks such as adding users and changing passwords can be performed using the WUI option: *Maintenance > Passwords*.

Accessing the WUI using HTTP:

http://192.168.2.21:9080/lbadmin/

(replace 192.168.2.21 with your IP address if it's been changed)

Accessing the WUI using HTTPS:

https://192.168.2.21:9443/lbadmin/

(replace 192.168.2.21 with your IP address if it's been changed)

Login to the WUI:

Username: loadbalancer
Password: loadbalancer



NOTE: A number of interoperability issues have been found with various versions of IE. The WUI has been tested and verified using both Firefox & Chrome.

Configuring the Appliance using the Wizard

Currently the wizard can only be used to setup a single layer 4 DR mode or NAT mode Virtual Service with a single Real Server. If additional Real Servers must be defined, or if you require a more complex configuration, please use the WUI. The wizard supports both single unit deployments and clustered pair deployments.

Outline steps – Single unit deployments:

- Set the IP address using the methods described earlier
- Now start the WUI and run the Wizard (*Cluster Configuration > Setup Wizard*)

Outline steps – Clustered pair deployments:

- Set the IP address on both units as described earlier
- For hardware appliances connect the serial cable
- Start the WUI on the slave unit and run the Wizard (*Cluster Configuration > Setup Wizard*)
- Now run the Wizard on the master unit to complete the process

Running the Wizard

The following prompt is displayed when first accessing the WUI:



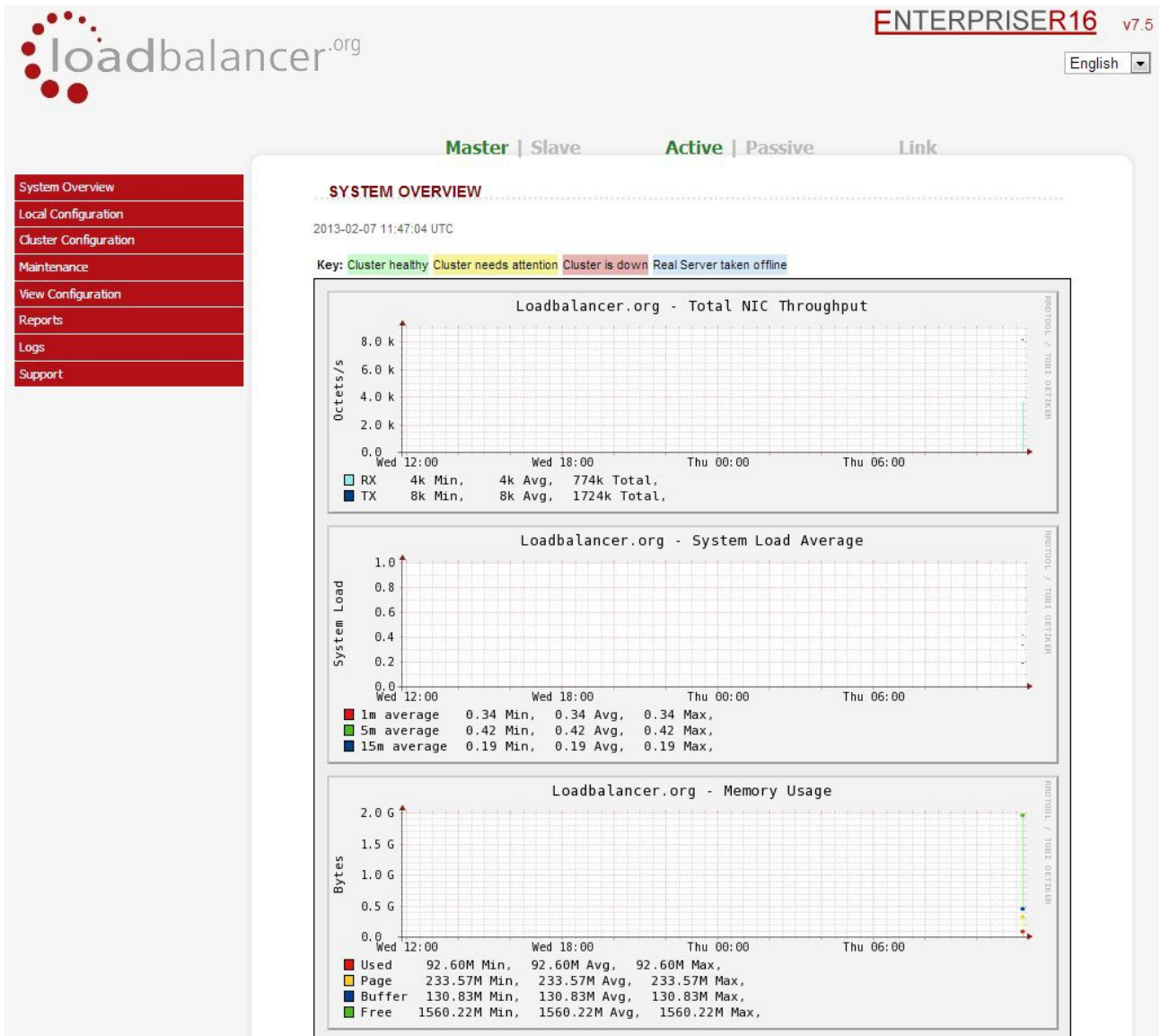
The wizard can be run at this point by selecting **yes** or at any other time using the WUI option: *Cluster Configuration > Setup Wizard*



NOTE: If you need to re-run the wizard at some later point, first restore the configuration to defaults before running the wizard using the WUI option: *Maintenance > Backup & Restore > Restore Manufacturers Defaults*

Configuration the Appliance using the WUI

Once logged in, the WUI is displayed as shown below:



Main Menu Options

System Overview – Displays a graphical summary of all VIPs, RIPS and key appliance statistics

Local Configuration – Configure local host settings such as IP address, DNS, system time etc.

Cluster Configuration – configure load balanced services such as VIPs & RIPS

Maintenance – Perform maintenance tasks such as service restarts and taking backups

View Configuration – Display the saved appliance configuration settings

Reports – View various appliance reports & graphs

Logs – View various appliance logs

Support – Create a support download & contact the support team

For details of configuring the appliance and setting up load balanced services please refer to Chapter 5 (Appliance Management) and Chapter 6 (Configuring Load Balanced Services).

Full Root Access

One of the great advantages of the Loadbalancer.org appliance is that you have full root access. This unlocks the benefits of the underlying Linux OS. Other vendors tend to lock this down and only provide limited access to certain tools.

Appliance Configuration Files & Locations

Physical configuration:	/etc/sysconfig/network-scripts/ifcfg-eth*
Firewall configuration:	/etc/rc.d/rc.firewall
Firewall Lock down wizard:	/etc/rc.d/rc.lockdownwizard.conf
XML configuration file:	/etc/loadbalancer.org/lb_config.xml
Layer 4 configuration:	/etc/ha.d/conf/loadbalancer.cf
Layer 7 HAProxy configuration	/etc/haproxy/haproxy.cfg
Pound SSL configuration	/etc/pound/pound.cfg
STunnel configuration	/etc/stunnel/stunnel.conf
SSL Certificates	/etc/loadbalancer.org/certs
Fail-over (heartbeat) configuration:	/etc/ha.d/ha.cf



If you do require a custom configuration please contact our support team to discuss your requirements: support@loadbalancer.org

Chapter 5 – Appliance Management

Network Configuration

Physical Interfaces

The number of physical network interfaces depends on the model. The Enterprise and Enterprise R16 have 2 physical interfaces, the MAX and 10G have 4 physical interfaces and the various VA's all have 4 interfaces. If multiple logical interfaces are required, these can be added simply by specifying multiple IP addresses as shown below. If multiple cables must be connected, an external switch can be used.

Typically, the only reason for using all 4 interfaces is when bonding (e.g. 802.3ad) is required in a 2-arm SNAT mode (layer 7) or 2-arm NAT mode (layer 4) highly available configuration.

Configuring IP Addresses

IP addresses can be configured using the WUI option: *Local 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 subnet mask and the default gateway. Both IPv4 and IPv6 addresses can be configured.

To set IP address(es):

- In the WUI, open *Local Configuration > Network Interface Configuration*
- Assign the required IP address / mask, multiple addresses can be assigned to each adapter as shown below

The screenshot displays the Network Interface Configuration web user interface (WUI) with three main sections:

- Bonding:** A checkbox labeled "Bond eth0 & eth1 as bond0:" is currently unchecked. To its right is a help icon (question mark) and a button labeled "Bond Interfaces".
- VLAN:** An "Interface:" dropdown menu is set to "eth0". To its right is a help icon and a button labeled "Add VLAN". Below this, a "VLAN ID:" input field contains the number "1", with a help icon to its right.
- IP Address Assignment:** This section contains two text areas for configuring IP addresses on specific interfaces:
 - eth0:** The text area contains two lines of IPv4 addresses: "192.168.2.100/24" and "192.168.4.100/24".
 - eth1:** The text area contains two lines of addresses: "10.20.1.1/16" and "fde6:d14c:3089:1::360/64".

At the bottom of the configuration area is a button labeled "Configure Interfaces".

- Click **Configure Interfaces**



NOTE: If you already have Virtual Services defined when making changes to the network configuration, you should verify that your Virtual Services are still up and working correctly after making the changes.

Configuring Bonding

- In the WUI, open *Local Configuration > Network Interface 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



NOTE: At this point the interfaces will still have the same IP settings configured previously. Once an IP address is defined for the bond and **Configure Interfaces** is clicked these addresses will be removed. If bonding is later disabled these addresses will be re-applied to the interfaces.

- Enter the IP address for bond0 and click **Configure Interfaces**

The screenshot shows a web interface for network configuration. It is divided into three main sections: Bonding, VLAN, and IP Address Assignment. In the Bonding section, the checkbox 'Bond eth0 & eth1 as bond0:' is checked, and there is a 'Bond Interfaces' button. In the VLAN section, the 'Interface:' dropdown is set to 'bond0', and there is an 'Add VLAN' button. The 'VLAN ID:' field contains the number '1'. In the IP Address Assignment section, the interface 'bond0' is listed, and the IP address '192.168.2.100/24' is entered in the adjacent field. A 'Configure Interfaces' button is located at the bottom of the IP Address Assignment section.

By default, the bond is configured for high-availability, this can be changed by editing the file `/etc/modprobe.d/loadbalancer.conf`. This is covered in the following section.



NOTE: If you have a master and slave configured as an HA pair, make sure you configure bonding in the same way on both units. Failure to do this will result in heartbeat related issues.

Bonding Configuration Modes

Ideally all single points of failure should be eliminated from a network. To help achieve this a cross-wired switch environment can be used. Every server including the load balancers is cross wired into two switch stacks. 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 setup the appliance using a single network card and are happy with the configuration you can set up bonding using *Local Configuration > Network Interface Configuration*.

If required you can change the bonding mode in the file: /etc/modprobe.d/loadbalancer.conf. By default mode 1 is used which configures the bond for high availability. Simply edit the file and set the mode setting as needed.

Supported Modes:

Bonding for High-Availability (default mode)

mode 1

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

Bonding for Bandwidth

Change to mode 0

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

Bonding for High-Availability & Bandwidth

Change to mode 4

```
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.



If your Real Servers, ESX hosts etc. support network bonding using Broadcom's SLB (Smart Load Balancing), this can cause issues in Layer 4 DR mode if older drivers are used. We have successfully tested SLB (Auto Fallback Disable) with driver version 15.2.0.5. Therefore at least this version is recommended.

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 no VLAN configuration is required. If the load balancer is connected to a trunk port, then all the required VLANs will need to be configured on the load balancer.

To configure a VLAN:

- In the WUI, open *Local Configuration > Network Configuration*
- In the VLAN section 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 in this field

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

Buttons: 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 the same VLANs on the slave as these will not be replicated / created automatically.

Configuring Default Gateway & Static Routes

To set the Default Gateway for IPv4 and IPv6:

- In the WUI, open *Local Configuration > Routing*
- In the Default Gateway section define the IP addresses as shown in the example below:

Default Gateway			
IP v4	<input type="text" value="192.168.64.1"/>		
IP v6	<input type="text"/>		

Static Routes			
Subnet	<input type="text"/>	via gateway	<input type="text"/>
Subnet	<input type="text"/>	via gateway	<input type="text"/>
Subnet	<input type="text"/>	via gateway	<input type="text"/>

- Click **Configure Routing**

To configure Static Routes:

- In the WUI, open *Edit Configuration > Routing*
- In the Static Routes section configure the subnets & gateways as shown in the example below:

Default Gateway			
IP v4	<input type="text" value="192.168.64.1"/>		
IP v6	<input type="text"/>		

Static Routes			
Subnet	<input type="text" value="10.100.0.0/16"/>	via gateway	<input type="text" value="192.168.64.80"/>
Subnet	<input type="text"/>	via gateway	<input type="text"/>
Subnet	<input type="text"/>	via gateway	<input type="text"/>

- Click **Configure Routing**





N.B. Unlimited routes can be added, additional blank rows will be added to the WUI screen as they're used

Configuring Hostname & DNS Configuration

From v7.5 it's possible to define a custom hostname for each appliance. A new field named *Role* enables the purpose the appliance to be defined.

To set the Hostname, Role & DNS servers:

- In the WUI, open *Local Configuration > Hostname & DNS*

Hostname	<input type="text" value="lbmaster"/>	
Role	<input type="text" value="master"/>	
Domain Name Server – Primary	<input type="text" value="192.168.2.20"/>	
Domain Name Server – Secondary	<input type="text"/>	

- Specify the required *Hostname*, by default this is set to **lbmaster**
- Select the required *Role* – for a single unit leave this set to **master**, for an HA pair set the master appliance to **master** and the slave unit to **slave**

*N.B. If Hostname is left at its default value of **lbmaster**, when Role is changed to **slave**, Hostname will be automatically set to **lbslave**, if a custom hostname has been specified this will remain unchanged irrespective of which role is selected*

- Specify the DNS servers using the *Domain Name Server – Primary* and *Domain Name Server – Secondary* fields.
- Click **Update**

System Date & Time and NTP Server Configuration

From v7.5 full NTP time synchronization is supported. The WUI allows up to 3 NTP servers to be defined.

Auto Configuration using NTP Servers

To configure NTP:

- In the WUI, open *Local Configuration > System Date & Time*

Current system time 2013-02-07 15:24:16 UTC

System Timezone UTC

NTP Servers

Set Timezone & NTP

Date 2013 - 2 - 7

Time 15 : 24

Set Date & Time

- Select the required *System Timezone*
- Define your NTP servers using the *NTP Servers* fields
- Click **Set Timezone & NTP**

Manual Configuration

To manually set the date & time:

- Set the data & time using the *Date & time* fields
- Click **Set Date & Time**



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

Appliance Internet Access via Proxy

The appliance supports the ability to access the Internet via a proxy server.

To set the Proxy Server's IP address & Port:

- In the WUI, open *Local Configuration > Physical Advanced Configuration*



The screenshot shows a configuration section titled "Internet Access:". Below the title, there are two rows of configuration fields. The first row is labeled "Proxy IP Address" and has an empty text input field followed by a blue help icon. The second row is labeled "Proxy Port" and has an empty text input field followed by a blue help icon.

- Enter an appropriate IP address in the *Proxy IP Address* field
- Enter an appropriate port in the *Proxy Port* field
- Click **Update**

SMTP Relay Configuration

The appliance can be configured with an SMTP smart host to receive all mail messages generated by the load balancer. If this field is not configured the address will be auto-configured based on an MX lookup of the destination email address that's configured under *Cluster Configuration > Layer 4 – Advanced Configuration*.

To configure a smart host:

- In the WUI, open *Local Configuration > Physical Advanced Configuration*



The screenshot shows a configuration section titled "SMTP Relay:". Below the title, there is one row of configuration fields labeled "Smart Host" with an empty text input field followed by a blue help icon.

- Enter an appropriate IP address or hostname in the *Smart Host* field
- Click **Update**

Syslog Server Configuration

The appliance supports the ability to write all logs to an external Syslog Server. Once defined, all log messages will be sent to the remote server and logs will no longer be maintained locally. The server may be specified by IP address or hostname. Note: If you use a hostname, make sure DNS is correctly configured on the loadbalancer.

To configure a Syslog server:

- In the WUI, open *Local Configuration > Physical Advanced Configuration*

Syslog Server:

IP or Hostname 

- Enter an appropriate IP address in the *Proxy IP Address* field
- Enter an appropriate port in the *Proxy Port* field
- Click **Update**

Appliance Upgrade (Enterprise R16 to Enterprise)

The Enterprise R16 can be upgraded to the full (unrestricted) Enterprise model.

To upgrade the license:

- Contact sales@loadbalancer.org to purchase an upgrade license
- In the WUI, open *Local Configuration > Upgrade Appliance*
- Enter the license key provided
- Click **Install License Key**

Running OS Level Commands

The appliance supports the ability to run OS level commands directly from the WUI.

To run a command:

- In the WUI, open *Local Configuration > Execute Shell Command*
- Enter the relevant command in the field
- Click **Execute Shell Command**

The results of the command / any errors will be displayed at the top of the screen.

Restoring Manufacturer's Settings

The load balancers settings can be reset to factory default values in two ways. In both cases 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.

Using the WUI

- In the WUI, open *Maintenance > Backup & Restore*
- Click **Restore Manufacturer's Defaults**

Once restored, restart the appliance to complete the process.

Using the Console / SSH Session

```
lbrestore
```

Once restored, restart the appliance to complete the process.

Restarting Services

The various services running on the appliance can be manually reloaded or restarted if required. This is normally only required for HAProxy, Pound, STunnel and Heartbeat when configuration changes are made.



Restart Ldirectord

Restart Layer 4 Services. Restarting Ldirectord will result in a loss of layer 4 services during the restart. This causes the related process to be stopped and a new instance started. Generally only needed if Ldirectord has failed for some reason and needs to be started again from scratch.

Reload Ldirectord

Reload Layer 4 Services. The Ldirectord configuration is re-read and re-applied. Note that a reload occurs automatically whenever a layer 4 VIP or RIP is added, deleted or modified.

Restart HAProxy

Restart Layer 7 Services. Restarting HAProxy will result in a loss of layer 7 services during the restart. Restarting HAProxy will cause any persistence tables to be dropped and all connections to be closed, it's a complete restart and reload of the HAProxy configuration.

Reload HAProxy

Reload Layer 7 Services. HAProxy will start a new process (leaving the old one) with the new configuration. New connections will be passed onto this process, the old process will maintain existing connections and eventually terminate when there are no more connections accessing it. If you are using stick tables for persistence the entries will be copied between processes. *N.B. If you have long lasting tcp connections it can take quite some time for the old process to terminate, leaving those users running the old configuration. If this is taking too long – See Restart HAProxy.*

Clear HAProxy Stick Table

Clears All HAProxy persistence tables. If you are using a Layer 7 persistence mode that relies on stick-tables (IP persistence or RDP cookie persistence), this option will clear all entries from these tables. Once cleared, clients may be directed to a different server upon re-connection.

Restart Pound

Restart Pound SSL Termination Services. Restarting Pound will result in a loss of SSL termination services during the restart.

Restart STunnel

Restart STunnel SSL Termination Services. Restarting STunnel will result in a loss of SSL termination services during the restart.

Restart Heartbeat

Restart Heartbeat Services. Restarting Heartbeat will result in a loss of service during the restart. Restarting heartbeat will cause a temporary loss of all layer 4, layer 7 and SSL services.

Reload Heartbeat

Reload Heartbeat Services. If the configuration has not changed then nothing will happen. If the config has changed, a restart will occur. Restarting Heartbeat will result in a loss of service during the restart. Restarting heartbeat will cause a temporary loss of all layer 4, layer 7 and SSL services.

Restart IPTables

Restarts iptables. This will clear then re-read and re-apply the firewall rules.

Restart Syslogd

Restart the syslog services.

Reload Syslogd

Reload the syslog services.

Appliance Restart & Shutdown

The appliance can be restarted or shutdown using the WUI.

To restart or shutdown the appliance:

- In the WUI, open *Maintenance > System Control*
- Select the required option:
 - **Restart Server** – Shutdown and restart the appliance
 - **Halt Server** – Shutdown and halt the appliance

Appliance 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.

Checking the Current Software Version & Revision

The current software version and revision can be checked at the console, via an SSH session or via the WUI using the following command:

```
cat /etc/loadbalancer.org/version.txt
```

Online Update

to perform an online update:

- In the WUI, open *Maintenance > Software Update*
- Select **Online Update**

- Information similar to the following will be displayed:

Online Update

Online updates are only available if your organisation has a valid authorisation key. An authorisation key may be obtained from [Loadbalancer.org](https://loadbalancer.org) support.

Before starting the online update, we recommend that you backup the XML configuration file, firewall script, and any manual changes that have been made.

- Download XML Configuration File
- Download Firewall Script

Update from v7.4.2 to v7.4.3

Changes in this release:

- Fix data handling problem when restoring from an uploaded XML configuration file.

Warning: Updates should only be installed during a maintenance window.

Note: When the online update is started, some web browsers will remain in page loading state for an extended period. Once the update archive has been downloaded, the page display will update.

The screenshot shows a web interface for starting an online update. It features a text input field labeled 'Authorisation Key' with a dashed border. Below the input field is a button labeled 'Start Online Update'.

- Enter a valid Authorization Key – this can be found in your Technical Support Document under Online Update Code
- Click **Start Online Update**
- The online update process will start, and will display update messages similar to the following:
Starting online update...
Downloading update archive...
Archive downloaded.
Archive checksum verified.
Upgrading packages...
- Once complete (the update can take several minutes depending on download speed and upgrade version) the following message is displayed:

Information: Update completed successfully.

- Once the appliance is completely up to date and there are no additional updates available, a message similar to the following will be displayed:

Information: Version v7.4.3 is the current release. No updates are available.

NOTES:

- As indicated in the WUI, we recommend that you backup your XML configuration and firewall script using the links provided before running the update
- Make sure that the load balancer is able to access the Internet – if you have a proxy server, this can be defined using *Local Configuration > Physical Advanced Configuration*
- Make sure that the default gateway is set correctly (*Local Configuration > Routing*)
- Make sure that the DNS server are set correctly (*Local Configuration > Hostname & DNS*)

Offline Update

If the load balancer does not have access to the Internet, Offline Update can be used.

To perform an offline update:

- In the WUI, open *Maintenance > Software Update*
- Select **Offline Update**
- The following screen will be displayed:

Offline Update

The following steps will lead you through offline update.

1. Contact Loadbalancer.org support to obtain the offline update archive and checksum.
2. Save the archive and checksum to your local machine.
3. Select the archive and checksum files in the upload form below.
4. Click *Upload and Install* to begin the update process.

Archive:

Checksum:

- As explained in the on-screen text, contact the Loadbalancer.org support to obtain the archive & checksum files
- Browse to and select these files
- Click **Upload and Install**

Updating a Clustered Pair

To update a clustered pair:

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 143.

Firewall Configuration



Whilst the load balancer is capable of supporting complex firewall rules, we do not recommend using the load balancer as your main bastion host. We recommend that the load balancer is deployed behind your external firewall.

If you want to configure firewall rules, some points to consider are:

1. All Virtual Service connections are dealt with on the INPUT chain not the FORWARD chain
2. The WUI runs on HTTP port 9080 and HTTPS port 9443
3. SSH on the load balancer listens on the standard port (22)
4. SNAT & DNAT is handled automatically for all layer 4 NAT mode (LVS) and layer 7 (HAProxy) based Virtual/Real load balanced services
5. You can use the standard Linux filters against spoofing attacks and syn floods
6. LVS has built in DOS attack filters that can be implemented
7. Plenty of extra information is available on the Internet relating to Linux Netfilter and LVS (Linux Virtual Service), if you need any assistance email support: support@loadbalancer.org

Manual Firewall Configuration

The firewall can be configured manually using the WUI based script editor. This enables iptables rules and any other required commands to be easily defined. The form allows you to directly edit `/etc/rc.d/rc.firewall`.

Custom rules can be configured, or for belt & braces security your external firewall settings can be replicated on to the load balancer for multi-layer security.

If you're planning to use NAT mode you may want to use the load balancer as your main firewall but we recommend it is better and simpler to keep your firewall separate from the load balancer, especially if you want to set up VPNs etc.

You can also use the firewall script to group ports together using Firewall Marks (see page 84).

To configure custom firewall rules:

- In the WUI, open *Maintenance > Firewall Script*
- The following screen will be displayed:

```
#!/bin/sh
# $Id: rc.firewall 2766 2012-08-17 13:35:57Z nick $

#
# User firewall script for Loadbalancer.org appliance.
#


# Please note:
#
#       Most configurations will not require any changes to be made to
#       this script.
#
#       Administrators will only need to modify this script if their
#       needs are not met by the lock-down wizard, auto-NAT, and
#       automatic firewall mark functions of the web interface.

##### One-arm NAT Mode #####
# For one-arm NAT, ICMP re-directs will need to be disabled.
# (1 = on, 0 = off)
#echo "0" >/proc/sys/net/ipv4/conf/all/send_redirects
#echo "0" >/proc/sys/net/ipv4/conf/default/send_redirects

##### Manual Firewall Marks #####
# Example: Associate HTTP and HTTPS with Firewall Mark 1:
```

- Define additional rules anywhere in the script above the last two lines:

```
echo "Firewall Activated"
exit 0;
```

 **WARNING:** Be careful! - make a backup before changing this script so that you know you can roll everything back if you cause a problem. A backup can be created using the WUI option: *Maintenance > Backup & Restore > Make Local Firewall Script Backup*

Firewall Lock-down Wizard

The firewall lock down wizard can be used to automatically configure the load balancer to allow access to the various admin ports from one specific IP address or subnet. 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 255.255.255.0 which can be changed as required.

The firewall lockdown wizard is split into two files:

- *rc.lockdownwizard* contains the script that you can change.
- *rc.lockdownwizard.conf* contains a set of variable definitions that can be overwritten when you run the firewall lockdown wizard. So, please don't manually change this file.

When run, the script *rc.lockdownwizard* loads the settings from the definitions file *rc.lockdownwizard.conf* and uses them to generate the rules. The web interface only writes the definitions *rc.lockdownwizard.conf*, allowing the user to modify the script *rc.lockdownwizard*. You can modify *rc.lockdownwizard* via ssh or from the web interface via the link **[Modify the firewall lock down wizard script]**. Apart from this link there is no other influence from the web interface.

The default script does not depend on the configured Virtual Services or Real Servers, so the wizard does not need to be re-run when services are changed.

However, it does depend on the IP addresses of master and slave, and the ports used by the web interface, heartbeat, and HAProxy. If those settings are changed, the firewall lockdown wizard will need to be re-run in order to reflect the changes. The re-run of the firewall lockdown wizard will adapt the *rc.lockdownwizard.conf* definitions file automatically - all your changes in the script *rc.lockdownwizard* won't be touched when you re-run the firewall lockdown wizard.

To run the lock-down wizard:

- In the WUI, open *Maintenance > Firewall Lock Down Script*
- The following screen will be displayed:

Warning: Once the lock-down wizard is enabled, administration access to the load balancer will only be allowed from the Administration Subnet specified below.

Enable lock down script	<input checked="" type="checkbox"/>	?
Administration subnet	<input type="text" value="192.168.2.1/24"/>	?
<input type="button" value="Update firewall lock down"/>		

[[Modify the firewall lock down wizard script](#)]

- Define your administration subnet/host in the *Administration subnet* field
N.B. Make sure that the subnet mask is correct – by default a /24 mask is displayed.
- Click **Firewall Lock Down Wizard**

N.B. For a clustered pair, the wizard must be run on each appliance

Disabling the lock-down script

To disable the lock-down script un-check the *Enable lock down script* checkbox and click the **Update Firewall lock down** button.

N.B. If you accidentally block your own access to the appliance you will need to clear the current firewall rules and try again. to clear the firewall tables completely use the following command at the console:

```
/etc/rc.d/rc.flush-iptables
```

Conntrack Table Size

By default the connection tracking table size is set to 524288 and is fine in most cases. For high traffic deployment using NAT mode, or when using connection tracking in the firewall script, this value may need to be increased. If the connection tracking table fills up, the following error will be reported in the log:

```
ip_conntrack: table full, dropping packet.
```

To increase the setting:

- In the WUI, open *Local Configuration > Physical – Advanced Configuration*
- Use the following section:



The screenshot shows a web interface for Firewall configuration. Under the heading "Firewall:", there is a section for "Connection Tracking table size" with an empty input field and a help icon (question mark in a circle) to its right.

- Set the required value using the *Connection Tracking table size* field
- Click **Update**

Users & Passwords

By default the appliance includes three pre-defined user accounts. The default usernames, passwords, group membership and their primary use are:

Username	Default Password	Default Group	Permissions (see also the group 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'

N.B. These are Apache .htaccess style accounts and are not related to the local Linux OS level accounts.

The permissions for each group are shown below:

	Menu / Permissions							
Group	System Overview	Local configuration	Cluster Configuration	Maintenance	View Configuration	Reports	Logs	Support
config	Full	Full	Full	Full	View	Full	View	Full
report	View	None	none	None	View	Full	View	View
maint	Full	None	None	None	View	Full	View	Full

Modifying User Passwords

to modify a user's password:

- In the WUI, open *Maintenance > Passwords*
- In the following section, click the **[Modify]** link next to the relevant user

loadbalancer	[Modify]	
reportuser	[Modify]	[Delete]
maintuser	[Modify]	[Delete]

- Now change the password for the selected user:

Username *	<input type="text" value="helpdesk"/>
Password *	<input type="password"/>
Group	<input type="text" value="report"/> ▼
<input type="button" value="Edit User"/>	

Adding New Users

to add new users:

- In the WUI, open *Maintenance > Passwords*

- Use the following section:



The screenshot shows a web form for adding a new user. It consists of two input fields: 'Username *' and 'Password *', both with empty text boxes. Below these fields is a button labeled 'Add New User'.

- Enter 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**

Resetting forgotten Passwords

It's possible to reset passwords via the command line if required. To do this you'll need to login as root to the console / SSH session. The `htpasswd` command can then be used as shown below:

```
htpasswd -b /etc/loadbalancer.org/passwords loadbalancer <new password>
```

e.g. `htpasswd -b /etc/loadbalancer.org/passwords loadbalancer loadbalancer`

This example sets the password for user 'loadbalancer' to 'loadbalancer'

Appliance Security Lockdown Script

To ensure that the appliance is secure it's recommended that a number of steps should be carried out. These steps have been incorporated into a lockdown script which can be run at the console (recommended) or via a terminal session.

The script helps to lock down the following:

- the password for the 'loadbalancer' Web User Interface account
- the password for the Linux 'root' account
- which subnet / host is permitted access to the load balancer

It also regenerates the SSH keys that are used to secure communicating between the master and slave appliance.

To start the script, at the console or via an SSH terminal session run the following command:

```
lbsecure
```

Chapter 6 – Configuring Load Balanced Services

Layer 4 Services

The Basics

Layer 4 services are based on LVS (Linux Virtual Service). LVS implements transport-layer load balancing inside the Linux kernel. It is used to direct requests for TCP/UDP based services to the Real Servers, and makes services on the Real Servers appear as a Virtual Service on a single IP address.

Layer 4 services are transparent by default, i.e. the source IP address is maintained through the load balancer.

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.

When a VIP is added the load balancer automatically adds a corresponding floating IP address which is activated instantly. Check *View Configuration > Network Configuration* to ensure that the Floating IP address has been activated correctly. They will show up as secondary addresses / aliases.

Multiple ports can be defined per VIP, for example 80 & 443. In this case it may also be useful to enable persistence (aka affinity / stickiness) to ensure that clients hit the same back-end server for both HTTP & HTTPS traffic and also prevent the client having to renegotiate the SSL connection.

Creating Virtual Services (VIPs)

Each Virtual Service 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). Typically you'll need one Virtual Service for each distinct cluster. Multiple ports can also be specified.

to add a new layer 4 VIP:

- In the WUI, open *Cluster Configuration > Layer 4 – Virtual Services*
- Click **[Add a New Virtual Service]**

Label	<input type="text" value="VIP Name"/>	?
Virtual Service IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Service 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"/>	?

- Enter an appropriate *Label* (name) for the new Virtual Service
- Enter the required IP address in the *Virtual Service IP address* field
- Enter the required ports(s) in the *Virtual Service Ports* field, separate multiple ports with commas, specify a range with a hyphen and specify all ports using an asterisk (*)



NOTE: the following ports are used by the appliance and therefore cannot be used for Virtual Services: 22 (SSH), 9080 (WUI – HTTP), 9443 (WUI – HTTPS), 7777 (HAProxy statistics page), 7778 (HAProxy persistence table replication and 9081 (nginx fallback page).

- Select the required *Forwarding Method*

Direct Routing (DR) - This is the default 1-arm mode. Direct Routing is recommended as it's easy to understand and implement with two load balancers in failover mode (our recommended configuration). It only requires one external Floating IP address on the same subnet as your web server cluster and only one network card.

NAT – This is the default 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.

- Set *Persistent* as required

Enable persistence for this Virtual Service, 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.

N.B. If your Real Servers cannot keep session state persistence themselves, then you will obtain performance benefits from a load balancer, but may not obtain reliability benefits.

- Set the *Protocol* as required

TCP – Transmission Control Protocol is the default and most common option

UDP – User Datagram Protocol – used for DNS, SIP, etc.

One Packet Scheduling - used for UDP SIP connections

Firewall Marks – For use when traffic has been tagged in the firewall script using the MARK target

- Now proceed to define the RIPv (Real Servers) as detailed on page 61


Modifying a Virtual Service

When first adding a Virtual Service, only certain values can be configured, others are set at their default setting. These values can be changed after the Virtual Service has been created by clicking [**Modify**] next to the relevant Virtual Service. Additional settings that can be changed are:

Option	Sub-Option	Description
Balance Mode		<p>Weighted Least-Connection – assign more jobs to servers with fewer jobs, relative to the Real Servers' weight (the default).</p> <p>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.</p> <p>Destination Hash – assign jobs to servers through</p>

		looking up a statically assigned hash table by their destination IP addresses.
Feedback Method		<p>The method the load balancer uses to measure to performance of the Real Servers.</p> <p>Agent – A simple telnet to port 3333 on the Real Server.</p> <p>HTTP – A simple HTTP GET to port 3333 on the Real Server.</p> <p>none – No feedback (default setting).</p> <p>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.</p>
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.
Persistence Granularity		<p>Group IP addresses for the purposes of persistence. Some large ISPs use clustered proxies, where the clients' source IP address may change frequently. If you require persistence with HTTP, this may cause a problem. Setting a larger mask will associate a subnet with a single persistence record. For example, 255.255.255.0 specifies a whole class C subnet.</p> <p>The default is a single address, or 255.255.255.255.</p>
Fallback Server	IP Address	<p>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). When using HAProxy Layer 7 the nginx server port 80 is automatically disabled. You can also configure 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.</p>
	Port	Set the fallback server port, for DR mode leave this blank as it must be the same as the VIP.
	Email Alert Destination Address	Destination email address for server health-check notifications.
Health Checks	Check Type	<p>Specify the type of health check to be performed on the Real Servers. Note that External script is currently the only supported option for IPv6 Real Servers.</p> <p>Negotiate connection – Scan the page specified in Request to send, and check the returned data for the Response expected string.</p> <p>Connect to port – Attempt to make a connection to the specified port.</p> <p>External script – Use a custom file for the health check. Specify the script path in the Check Command field.</p>

		Off – No checks; all Real Servers are marked offline. On – No checks; all Real Servers are marked online. 5 – Repeating pattern of 5 Connect checks followed by 1 Negotiate check. 10 – Repeating pattern of 10 Connect checks followed by 1 Negotiate check.
	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.
	External Script 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.
Negotiate Check Options	Negotiate Check Service	Specify the protocol to use for negotiate health checks. For common protocols, this will match the Virtual Service port. Simple TCP may be used to send an arbitrary string to the server, and match against its response.
	Virtual Host	If the Negotiate check should be performed on a specific Virtual Host, specify the hostname here.
	Database Name	The database to use for the MySQL Negotiate check. This is a required option if MySQL is selected under Negotiate Check Service above. There is no default.
	Radius Secret	Configure the RADIUS secret string for the RADIUS negotiate check.
	Login	The login name to use with the Negotiate check where authentication is required.
	Password	The password to use with the Negotiate check where authentication is required.
	Request to send	With negotiate checks, 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 file names 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 (Object Class=*) search relative to this base. With Simple TCP, this should be a string to send verbatim to the server.
	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.

 For more details on configuring health-checks please refer to chapter 7.

Creating Real Servers (RIPs)

You can add an unlimited number of Real Servers to each Virtual Service (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 Service, whilst for a NAT mode Real Server, it's possible to configure the port to be the same or different to the Virtual Service's port.

to add a new layer 4 RIP:

- In the WUI, open *Cluster Configuration > Layer 4 – Real Servers*
- Click **[Add a New Real Server]** next to the relevant Virtual Service

Label	<input type="text" value="RIP Name"/>	
Real Server IP Address	<input type="text" value="IPAddress"/>	
Real Server Port	<input type="text"/>	
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 an appropriate *Label* (name) for the new Real Server
- Enter the required IP address in the *Real Server IP Address* field. This only applies to NAT mode, in DR mode port redirection is not supported so by default the port is the same as defined in the VIP
- Specify the required *Weight*, this 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
- Specify the *Minimum Connections*, this is 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

- Specify the *Maximum Connections*, this is 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

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 the master appliance using SSH or the console, then as root run the following commands:

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

Then login to the slave appliance using SSH or the console, then as root run the following commands:

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

N.B. To ensure that these sync daemons are started on each reboot put these commands in the rc.firewall. This can be done via the WUI using Maintenance > Firewall Script. Make sure that the full path is specified in the firewall script, i.e.

```
/usr/local/sbin/ipvsadm --start-daemon master  
/usr/local/sbin/ipvsadm --start-daemon backup
```

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

```
ipvsadm -Lc
```

N.B. This is the same command that the 'Layer 4 Current Connections' report is based on.

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



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

DR Mode Considerations

What Is the ARP Problem?

DR mode works by changing the MAC address of the inbound packets to match the Real Server selected by the load balancing algorithm. The destination IP address remains unchanged and therefore each Real Server must respond to both its own IP address and also the Virtual Service IP address (VIP). It's important that Real Servers do not 'fight' with the load balancer for control of the shared VIP. If they do then requests could be sent directly to the Real Servers rather than hitting the load balancer VIP as intended.

- You only need to resolve the ARP Problem on the Real Servers when you are using the default DR (Direct Routing) mode or TUN (Tunnel / IP-in-IP encapsulation) mode
- Real Servers must not respond to ARP requests for the VIP
- The application running on each real sever must respond to both the RIP address and the VIP address

Detecting the ARP Problem

Attempt to connect to the VIP and then use *Reports > Layer 4 Current Connections* to check whether the connection state is SYN_RECV as shown below. If it is, this is normally a good indication that the ARP problem has not been correctly 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
```

Resolving ARP Issues for Linux

Method 1 (using iptables)

You can use iptables (netfilter) on each Real Server to re-direct incoming packets destined for the Virtual Service 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
```

(Change the IP address to be the same as your Virtual Service)

This means redirect any incoming packets destined for 10.0.0.21 (the Virtual Service) locally, i.e. to the primary address of the incoming interface on the Real Server.



Method 1 may not always be appropriate if you're using IP-based virtual hosting on your web server. This is because the iptables rule above redirects incoming packets to the primary address of the incoming interface on the web server rather than any of the virtual hosts that are configured. Where this is an issue, use method 2 below instead.

Also, Method 1 does not work with IPv6 Virtual Services, use method 2 below instead.

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 Services 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 Services)

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 Services 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
```

N.B. Steps 1 & 2 can be replaced by writing directly to the required files using the following commands: (temporary until the next reboot)

```
echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
echo 1 > /proc/sys/net/ipv4/conf/eth0/arp_ignore
echo 1 > /proc/sys/net/ipv4/conf/eth1/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/eth0/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/eth1/arp_announce
```

Resolving ARP issues 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 to add this to the startup scripts for your server.

Resolving ARP issues for Mac OS X / 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 to add this to the startup 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.

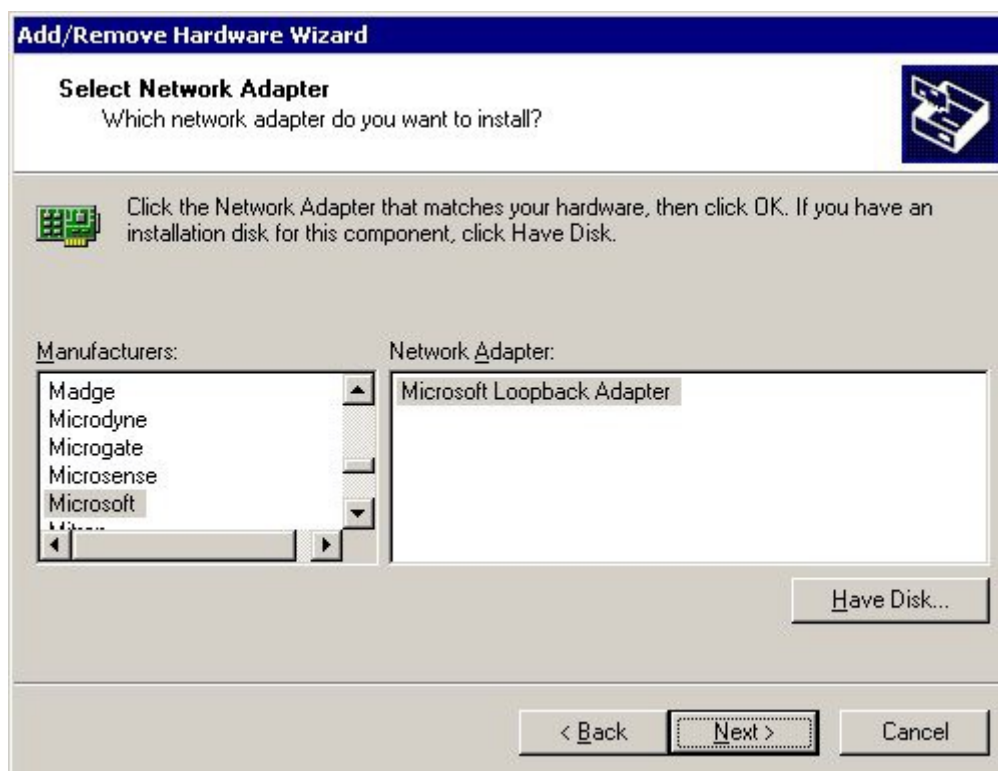
Resolving ARP issues for Windows Servers

Windows Server 2000

Windows Server 2000 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 Services 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.

Step 1: Install the Microsoft Loopback Adapter

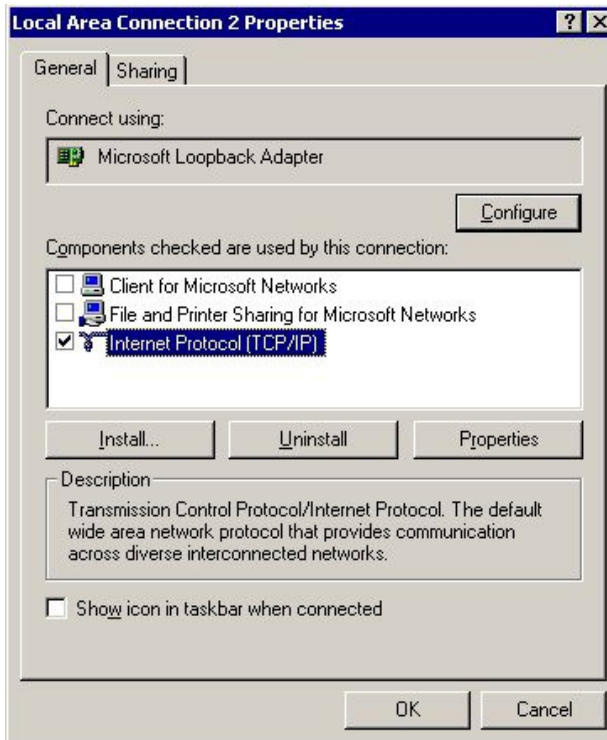
1. Open the Control Panel and double-click **Add/Remove Hardware**
2. Once the Hardware Wizard opens, click **Next**
3. Select **Add/Troubleshoot a device**, click **Next**
4. Once the device list appears, select **Add a new device** at the top of the list, click **Next**
5. Select **No, I want to select the hardware from a list**, click **Next**
6. Scroll down the list and select **Network Adapters**, click **Next**
7. Select **Microsoft & Microsoft Loopback Adapter**, click **Next** as shown below



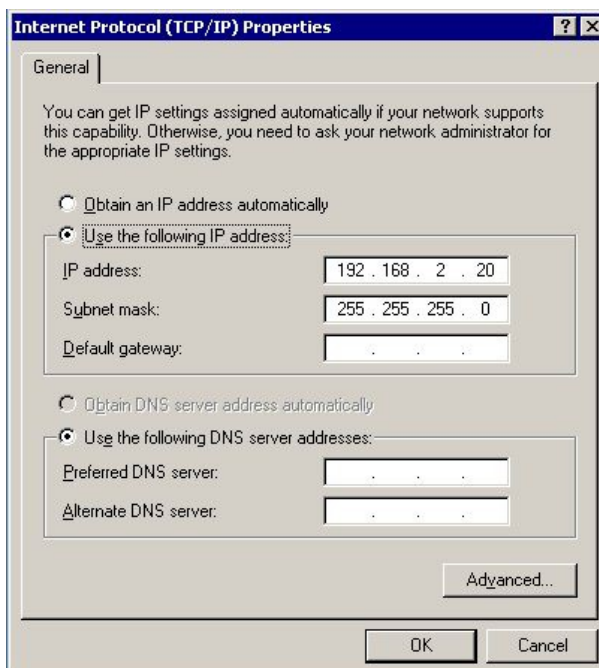
8. Click **Next** to start the installation, when complete click **Finish**

Step 2: Configure the Loopback Adapter

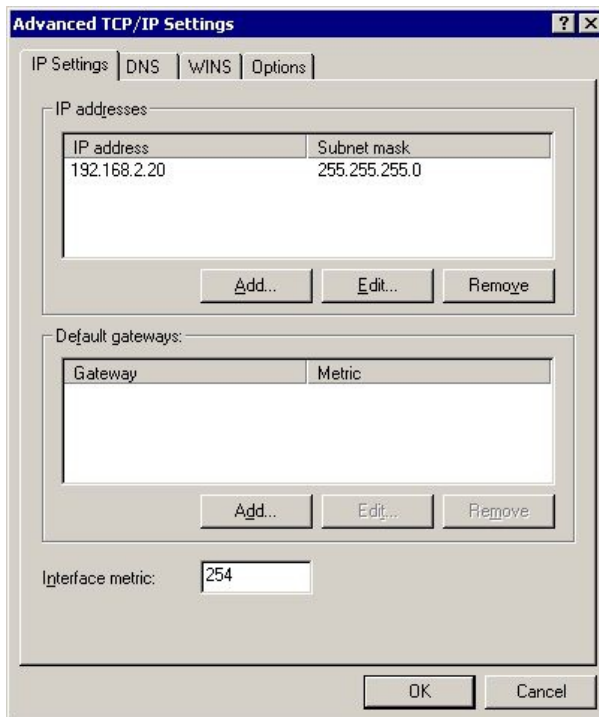
1. Open the Control Panel and double-click **Network and Dial-up Connections**
2. Right-click the new Loopback Adapter and select **Properties**
3. Un-check all items except **Internet Protocol (TCP/IP)** as shown below



4. Select **Internet Protocol (TCP/IP)**, click **Properties** and configure the IP address and mask to be the same as the Virtual Service IP address (VIP), e.g. 192.168.2.20/24 as shown below



5. Click **Advanced** and change the **Interface metric** to 254 as shown below, this prevents the adapter responding to ARP requests for the VIP address



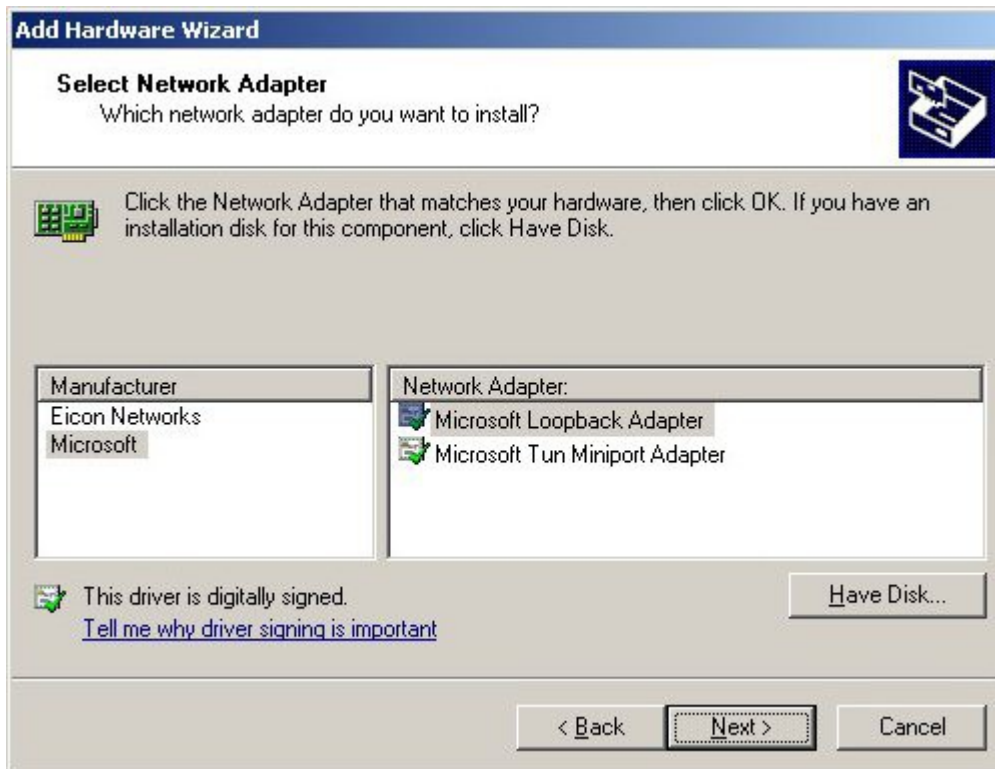
6. Click **OK** on Advanced Settings, TCP/IP Properties and Connection Properties to save and apply the new settings
7. Repeat the above steps for all other Windows 2000 Real Servers

Windows Server 2003

Windows server 2003 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 Services 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.

Step 1: Install the Microsoft Loopback Adapter

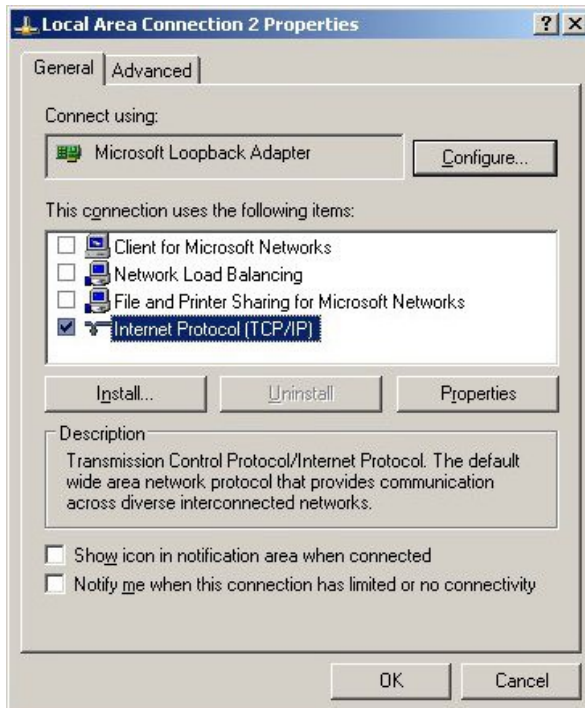
1. Open the Control Panel and double-click **Add Hardware**
2. Once the Hardware Wizard opens, click **Next**
3. Select **Yes, I have already connected the hardware**, click **Next**
4. Scroll to the bottom of the list, select **Add a new hardware device**, click **Next**
5. Select **Install the hardware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



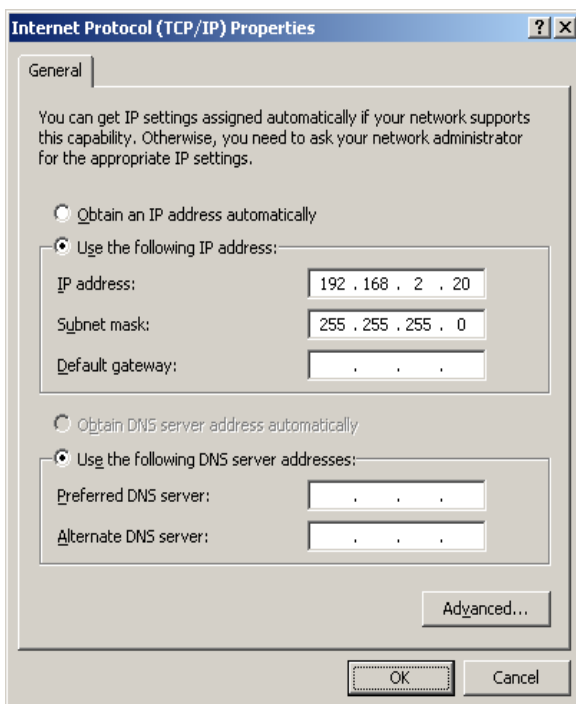
8. Click **Next** to start the installation, when complete click **Finish**

Step 2: Configure the Loopback Adapter

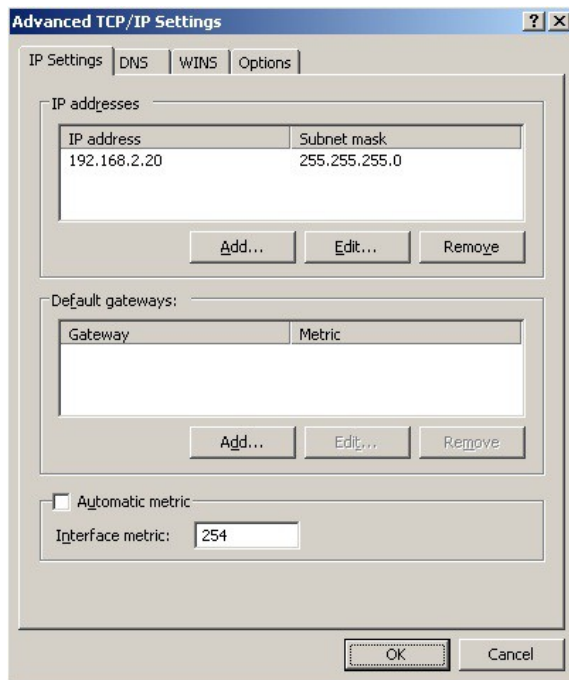
1. Open the Control Panel and double-click **Network Connections**
2. Right-click the new Loopback Adapter and select **Properties**
3. Un-check all items except **Internet Protocol (TCP/IP)** as shown below



4. Select **Internet Protocol (TCP/IP)**, click **Properties** and configure the IP address and mask to be the same as the Virtual Service (VIP), e.g. 192.168.2.20/24 as shown below



5. Click **Advanced**, un-check **Automatic metric** and change **Interface metric** to 254 as shown below, this prevents the adapter responding to ARP requests for the VIP address



6. Click **OK** on Advanced Settings & TCP/IP Properties, then click **Close** on Connection Properties to save and apply the new settings
7. Now repeat the above process for all other Windows 2003 Real Servers



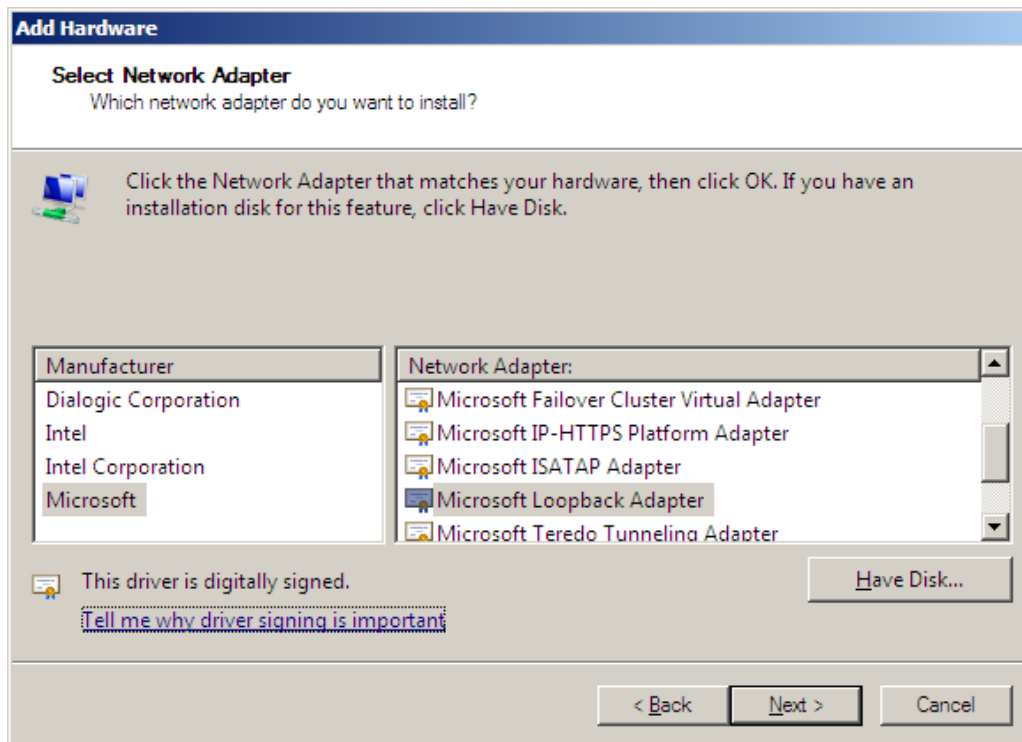
For Windows server 2003 SP1 & later, if you have enabled the built-in firewall, you will need to create an exception to enable access to the web server. This exception by default will allow traffic on both the network and Loopback Adapters.

Windows server 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. As with Windows 2000 / 2003, if the Real Server is included in multiple VIPs, you can add additional IP addresses to the Loopback Adapter that correspond to each VIP.

Step 1: Install the Microsoft Loopback Adapter

1. Click **Start**, then run **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**

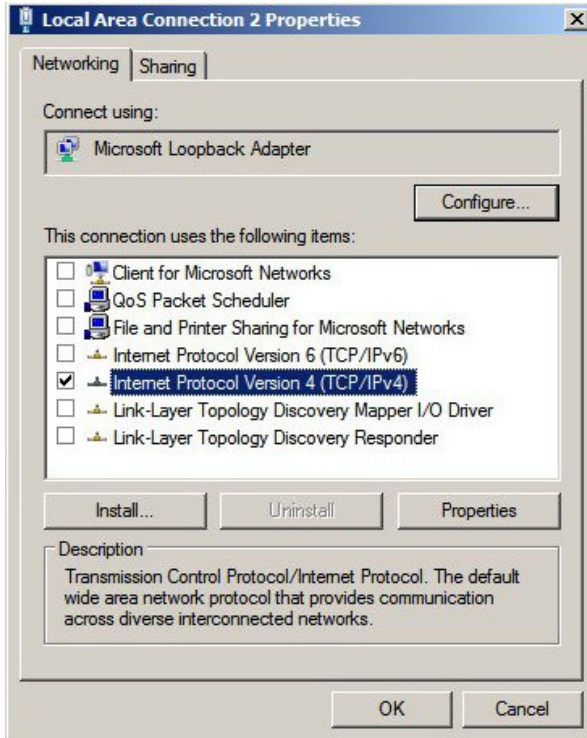


6. Click **Next** to start the installation, when complete click **Finish**

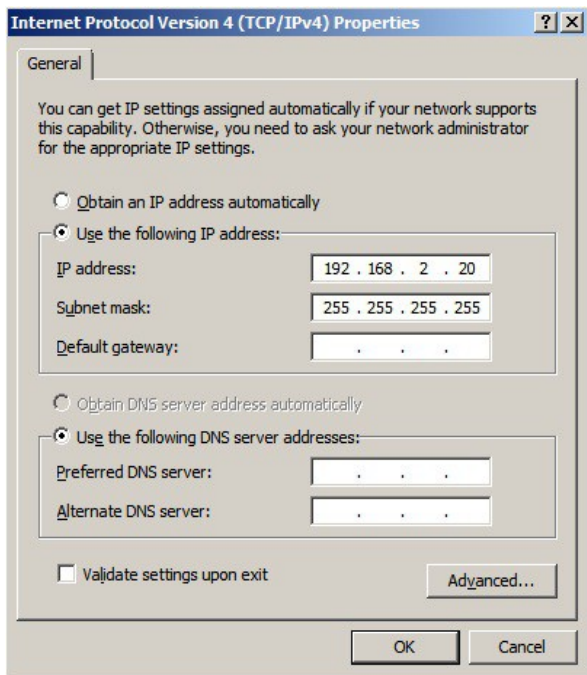
Step 2: Configure the Loopback Adapter

1. Open Control Panel and click **View Network status and tasks** under **Network and internet**
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)** as shown below



5. Select **Internet Protocol Version (TCP/IPv4)**, click **Properties** and configure the IP address to be the same as the Virtual Service (VIP) with a subnet mask of 255.255.255.255, e.g. 192.168.2.20 / 255.255.255.255 as shown below



6. Click **OK** on TCP/IP Properties, then click **Close** on Connection Properties to save and apply the new settings
7. Now repeat the above process on the other Windows 2008 Real Servers

N.B. For Windows 2008, it's not necessary to modify the interface metric on the advanced tab and should be left set to Automatic

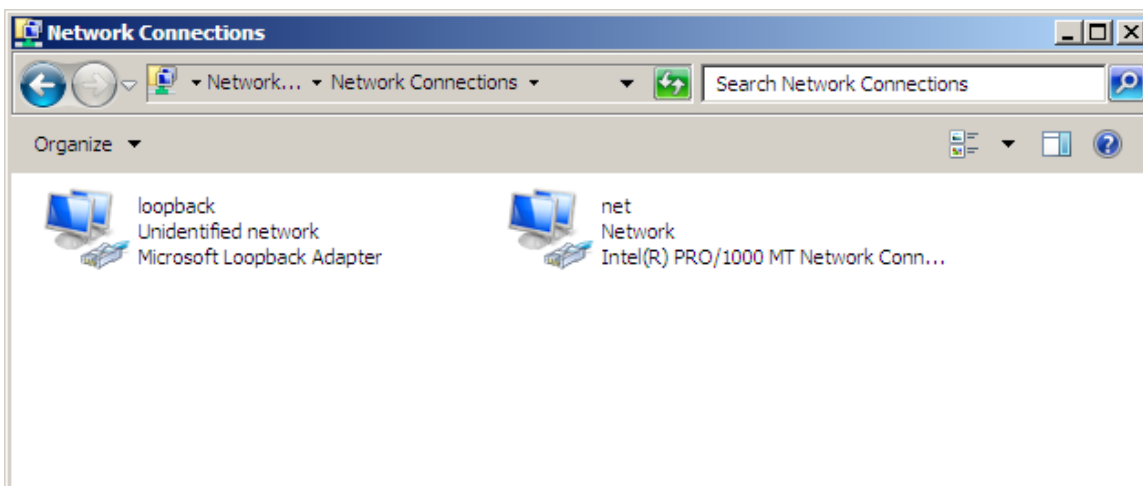
Step 3: Configure the strong / weak host behavior

Windows Server 2000 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 2008 and later 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 be able to respond to the VIP, the following commands must be run on each 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
```

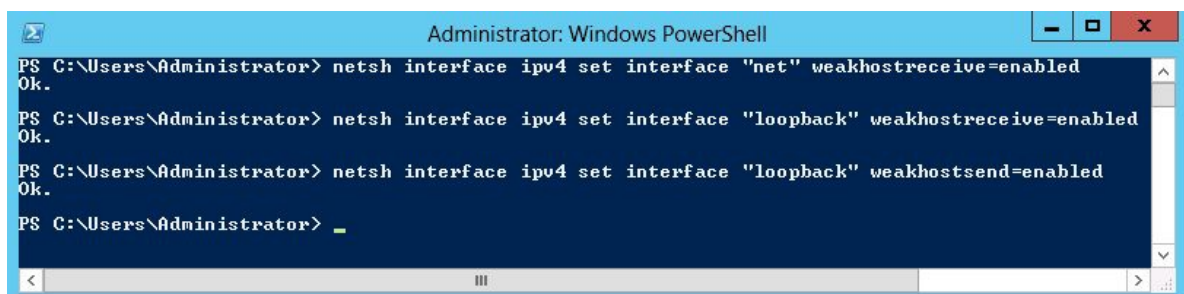
For these commands to work, the LAN connection NIC must be named "net" and the loopback NIC must be named "loopback" as shown below. If you prefer to leave your current NIC names, then the commands above must be modified accordingly. For example, if your network adapters are named "LAN" and "LOOPBACK", the commands required would be:

```
netsh interface ipv4 set interface "LAN" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostsend=enabled
```



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.

1. Start Powershell or use a command Window to run the 3 netsh commands as shown below



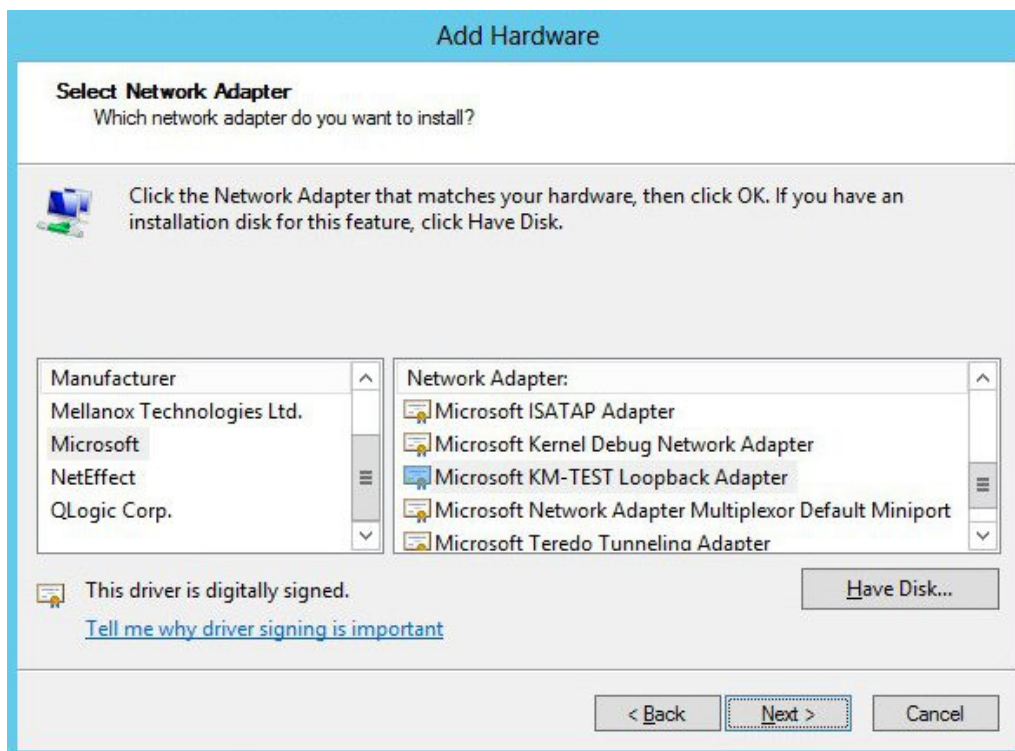
2. Now repeat these 3 commands on the other Windows 2008 Real Servers

Windows Server 2012

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. As with Windows 2000 / 2003 / 2008, if the Real Server is included in multiple VIPs, you can add additional IP addresses to the Loopback Adapter that correspond to each VIP.

Step 1: Install the Microsoft Loopback Adapter

1. Click **Start**, then run **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 KM-Test Loopback Adapter**, click **Next**

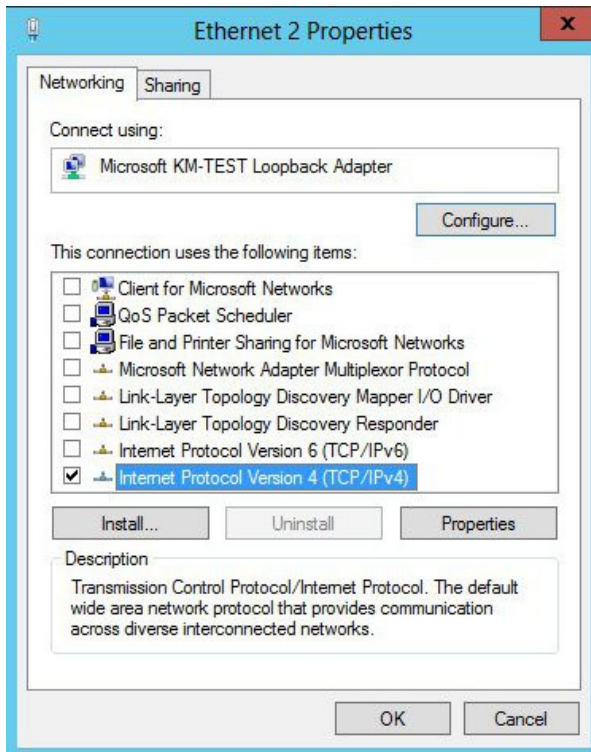


6. Click **Next** to start the installation, when complete click **Finish**

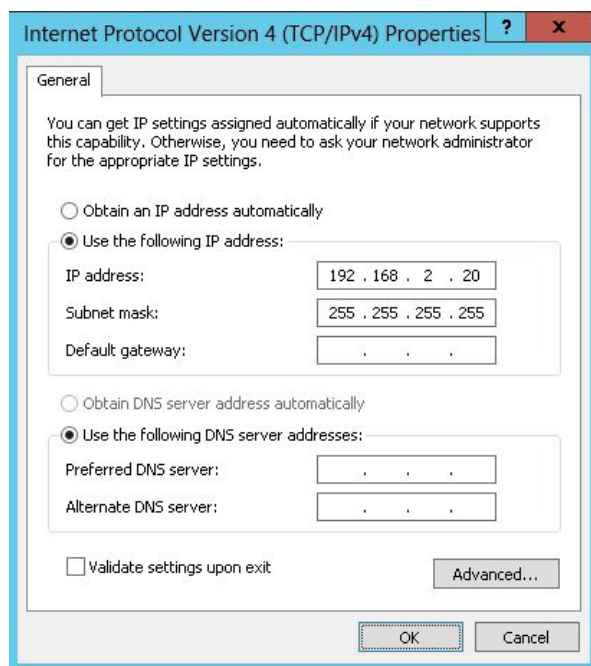
Step 2: Configure the Loopback Adapter

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

- Un-check all items except **Internet Protocol Version 4 (TCP/IPv4)** as shown below



- Select **Internet Protocol Version (TCP/IPv4)**, click **Properties** and configure the IP address to be the same as the Virtual Service (VIP) with a subnet mask of 255.255.255.255, e.g. 192.168.2.20 / 255.255.255.255 as shown below



- Click **OK** on TCP/IP Properties, then click **Close** on Ethernet Properties to save and apply the new settings
- Now repeat the above process on the other Windows 2012 Real Servers

N.B. For Windows 2012, it's not necessary to modify the interface metric on the advanced tab and should be left set to Automatic

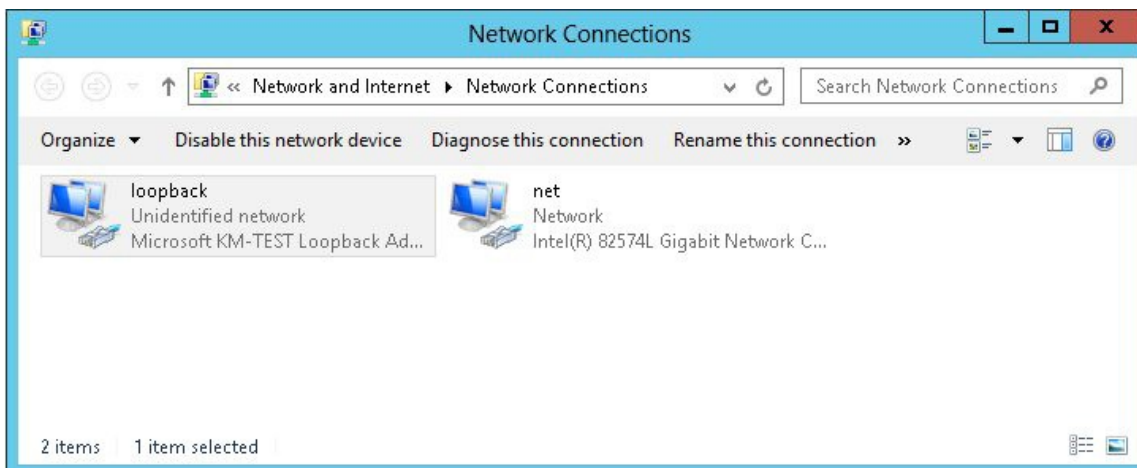
Step 3: Configure the strong / weak host behavior

Windows Server 2000 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 2008 and later 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 be able to respond to the VIP, the following commands must be run on each 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
```

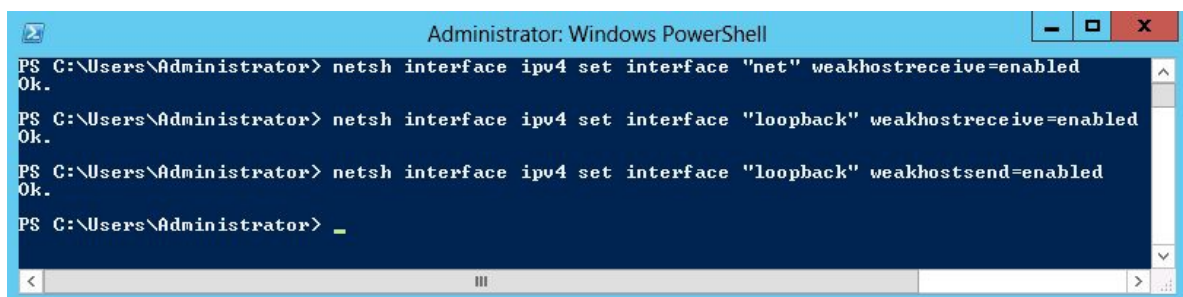
For these commands to work, the LAN connection NIC must be named "net" and the loopback NIC must be named "loopback" as shown below. If you prefer to leave your current NIC names, then the commands above must be modified accordingly. For example, if your network adapters are named "LAN" and "LOOPBACK", the commands required would be:

```
netsh interface ipv4 set interface "LAN" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostsend=enabled
```



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.

1. Start Powershell or use a command Window to run the 3 netsh commands as shown below



2. Now repeat these 3 commands on the other Windows 2012 Real Servers

Verifying netsh Settings for Windows 2008 & 2012

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`

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 / 2012, 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 problem in DR configurations.

Configuring IIS to Respond to both the RIP and VIP

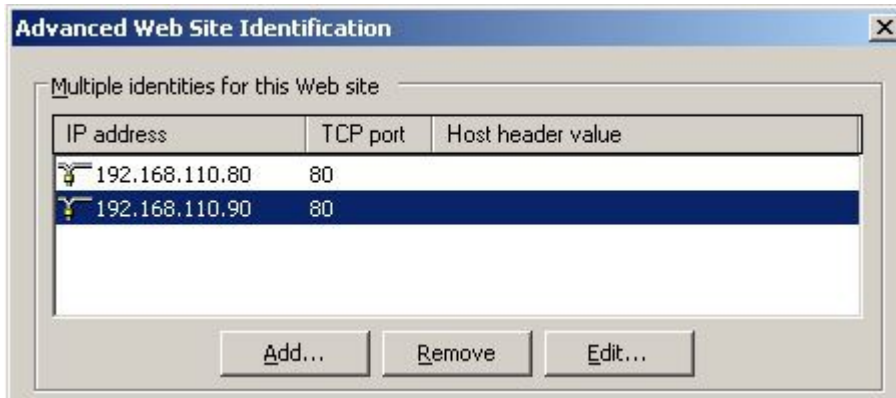
For DR & TUN modes, it's also important to make sure that IIS responds to both the VIP and RIP.

Windows 2000 / 2003

By default, IIS listens on all configured IP addresses, this is shown in the example below (shows Windows 2003 example). As can be seen the IP address field is set to 'All Unassigned'.

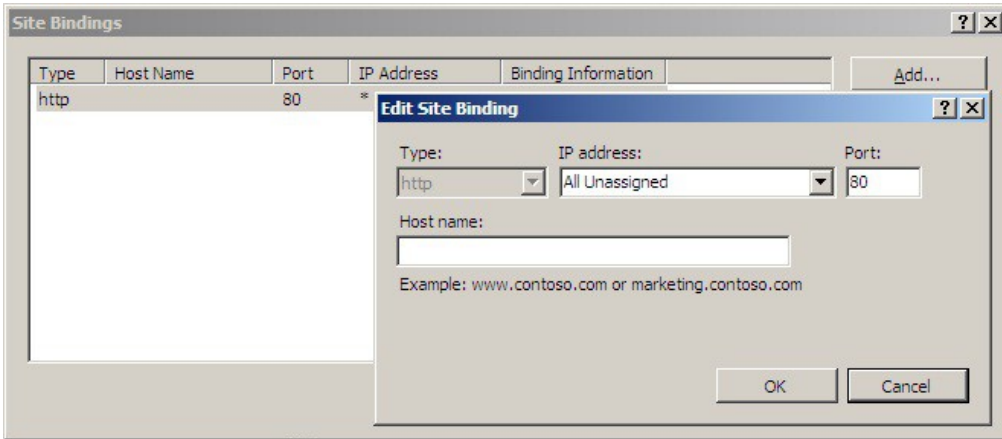


If the default configuration is left, no further IIS configuration is required. If you do change the IP address in the bindings from 'All Unassigned' to a specific IP address, then you need to make sure that you also add a binding for the Virtual Service IP address (VIP) as shown in the example below:

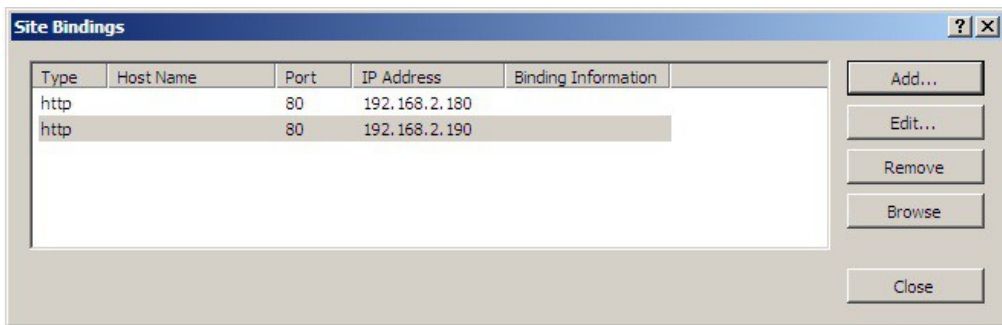


Windows 2008 / 2012

By default, IIS listens on all configured IP addresses, this is shown in the example below (shows Windows 2008 example). As can be seen the IP address field is set to “All Unassigned”.



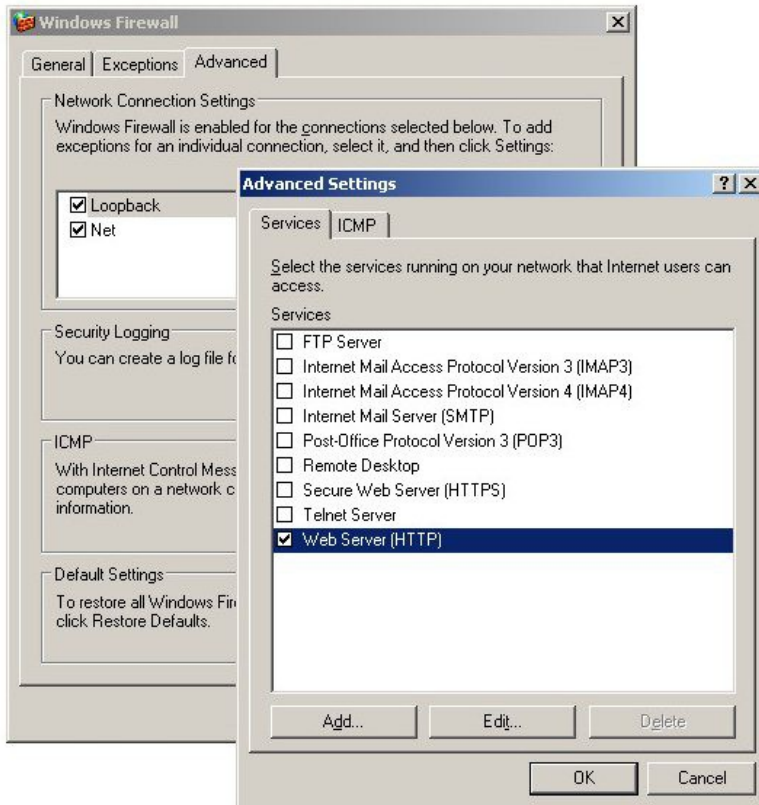
If the default configuration is left, no further IIS configuration is required. If you do change the IP address in the bindings from “All Unassigned” to a specific IP address, then you need to make sure that you also add a binding for the Virtual Service IP address (VIP) as shown in the example below:



Windows Firewall Settings

Windows 2003 SP1+

For Windows Server 2003 SP1 & later, if you have enabled the built-in firewall, you will need to enable the Web Server (HTTP) exception to permit access to the web server. This exception is created automatically when IIS is installed and when enabled allows traffic on both the network and Loopback Adapters.



Windows 2008 R1 Firewall Settings

For Windows 2008 R1 the firewall configuration is very similar to windows 2003 R2. Again, an exception is created automatically that must 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:



Windows 2008 R2 Firewall Settings

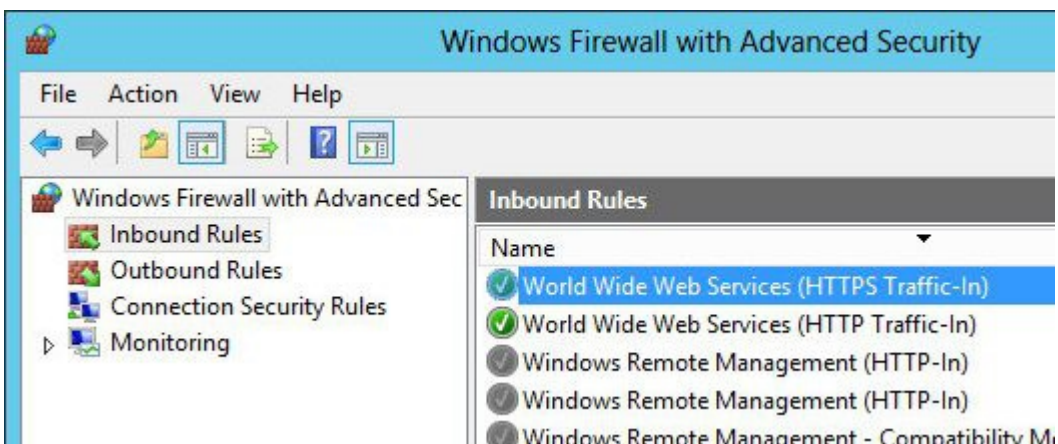
Windows 2008 automatically creates several default firewall rules for both inbound and outbound traffic. There are 3 firewall profiles and interfaces can be associated with one of these 3 profiles (domain, private and public) although the Loopback Adapter automatically gets associated with the public 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 2012 Firewall Settings

Windows 2012 is very similar to Windows 2008 R2 as shown below.



NAT Mode Considerations

NAT mode load balancing has the advantage that the only change required to the Real Servers is to modify the default gateway and possibly the IP address and subnet. Whilst NAT mode is fairly straight forward, a few points need to be considered.

NAT Mode Potential Issues

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

Enabling Real Server Internet access using Auto-NAT

When NAT mode is selected in the setup wizard, the Auto-NAT feature will be automatically enabled. If you have not used the wizard, you'll need to configure Auto-NAT manually.

To enable Auto-NAT manually:

- In the WUI, open *Cluster Configuration > Layer 4 – Advanced Configuration*
- Change Auto-NAT from **off** to the external interface being used – typically **eth1**
- Click **Update**

This activates the rc.nat script that forces external network traffic to be MASQUERADED to and from the external network. The iptables masquerade rule that's used for this is shown below:

```
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Enabling Access to non Load-Balanced Services

If you want specific services to be exposed on your Real Servers you have two choices:

- Setup a Virtual Service with a single Real Server for each service

or

- Setup a floating IP address and individual SNAT/DNAT rules for each service as shown in the example below. These lines can be added to the firewall script using the WUI option *Maintenance > Firewall Script*

```
INT_ADDR="10.50.110.238"  
EXT_ADDR="192.168.111.250"  
  
iptables -t nat -A POSTROUTING -p tcp -s $INT_ADDR -j SNAT --to-source $EXT_ADDR  
iptables -t nat -A PREROUTING -p tcp -d $EXT_ADDR -j DNAT --to-destination $INT_ADDR
```

Once the above SNAT/DNAT rules have been configured, the following firewall entries will be listed under *View Configuration > Firewall Rules*

```
Chain PREROUTING (policy ACCEPT 524 packets, 123K bytes)
pkts bytes target prot opt in out source destination
 2 104 DNAT tcp -- * * 0.0.0.0/0 192.168.111.250 to:10.50.110.238

Chain POSTROUTING (policy ACCEPT 80 packets, 4896 bytes)
pkts bytes target prot opt in out source destination
 0 0 SNAT tcp -- * * 10.50.110.238 0.0.0.0/0 to:192.168.111.250
```

N.B If Autonat is already enabled, only the DNAT rule will be required.



Please don't hesitate to contact support@loadbalancer.org to discuss any specific requirements you may have.

One-Arm (Single Subnet) NAT Mode

Normally the VIP and its associated RIPs are located on different subnets within the same logical network. However, it is possible to perform NAT mode load balancing on a single subnet by modifying the routing configuration of the Real Servers as detailed in the sections below.

Normally, when a client on the same subnet as the Real Server tries to access the Virtual Service on the load balancer the request will fail. This is because the Real Server will use the local network to get back to the client rather than going through the load balancer.

Route Configuration for Windows Servers

To rectify this issue for Windows servers, a route must be added to each server that takes priority over the default Windows routing rules.

This is a simple case of adding a permanent route as shown below:

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

N.B. Replace 192.168.1.0 with your local subnet address.

A new route will be added with a lower metric than the default route which forces all local traffic 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 Servers

To rectify this issue for Linux servers, we need to modify the local network route by changing 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
```

N.B. 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
```

N.B. Replace 192.168.1.21 with your load balancer gateway







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

Firewall Marks

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.

Firewall Marks – Auto Configuration

For example, to configure an HTTP/HTTPS NAT mode Virtual Service, simply specify port 80 & 443 separated by a comma in the 'Virtual Service Ports' field as shown below:

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"/>	
Forwarding Method	<input type="text" value="NAT"/>	
Persistent	<input type="text" value="yes"/>	
Protocol	<input type="text" value="TCP"/>	
<input type="button" value="Update"/>		

This will automatically configure the load balancer for firewall marks.

For NAT mode VIPs, leave the Real Server port blank as shown below:

Label	<input type="text" value="HTTP1"/>	
Real Server IP Address	<input type="text" value="192.168.50.2"/>	
Real Server Port	<input type="text"/>	
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"/>		

Packets will then be passed to the Real Servers on the same port as it was received at the VIP.

N.B. For Layer 4 DR mode VIPs, there is no Real Server Port field since port translation is not possible in this mode.

HEALTH CHECK PORT: For DR mode the check port is automatically set to be the first port in the list. For example, if ports 80 & 443 are defined for the VIP, the check port is automatically set to port 80. When using NAT mode, the check port must be set manually.

Firewall Marks – Manual Configuration

Firewall Marks can also be configured manually. This may be required for example when both TCP and UDP are needed for a particular VIP. The basic concept is to create a firewall rule that matches incoming packets to a particular IP address / port(s) and mark them with an arbitrary integer. A Virtual Service is also configured specifying this firewall mark integer instead of the IP address.

EXAMPLE 1 – Setup a new DR Mode Firewall Mark when no Initial VIP has been Created

Step 1: Create the New VIP

- Using the WUI, go to *Cluster Configuration > Layer 4 – Virtual Services*
- Click **[Add a new Virtual Service]**
- Instead of entering an IP address, enter a numeric value representing the 'mark' as shown below

Label	<input type="text" value="Server_Cluster"/>	?
Virtual Server IP address	<input type="text" value="1"/>	?
Virtual Server Ports	<input type="text"/>	?
Forwarding Method	<input type="text" value="Direct Routing"/>	?
Persistent	<input type="text" value="yes"/>	?
Protocol	<input type="text" value="Firewall Marks"/>	?
<input type="button" value="Update"/>		

- Leave the *Virtual Service Ports* field blank (the ports will be defined in the firewall script in step 5)
- Set the *Forwarding Method* to Direct Routing
- Set *Persistence* to Yes
- Set *Protocol* to Firewall Marks
- Click **Update**

Step 2: Define a Health-Check Port

- Using the WUI, go to *Cluster Configuration > Layer 4 – Virtual Services*
- Click **[Modify]** next to the new Virtual Service
- Enter the appropriate value in the *Check Port* field
- Click **Update**

Step 3: Add the Real Servers

- Using the WUI, go to *Cluster Configuration > Layer 4 – Real Servers*
- Click **[Add a new Real Server]**
- Enter the required details as shown below

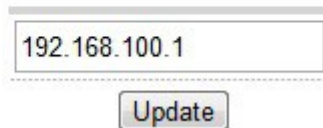
Label	<input type="text" value="Server1"/>	?
Real Server IP Address	<input type="text" value="192.168.100.10"/>	?
Weight	<input type="text" value="1"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?
<input type="button" value="Update"/>		

- Click **Update**

Step 4: Add the Associated Floating IP Address for the VIP

- Using the WUI, go to *Cluster Configuration > Floating IPs*
- Add a floating IP that corresponds to the required VIP, in this example 192.168.100.1

EDIT CONFIGURATION > ADD NEW FLOATING IP



192.168.100.1

Update

- Click **Update**

Step 5: Modify the Firewall Script

- Using the WUI, go to *Maintenance > Firewall Script*
- Uncomment / modify the example firewall marks section as shown below:

```
##### Manual Firewall Marks #####

# Example: Associate HTTP and HTTPS with Firewall Mark 1:
VIP1="192.168.100.1"
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

# A Virtual Service may then be created in the web interface, using 1 as the
# service address.

##### Packet Filtering #####

# You should always use a network perimeter firewall to lock down all
# external access to the load balancer except the required Virtual Services
# and the required services from your admin machine / network (SSH & HTTPS)

# Allow unlimited traffic on the loopback interface:
#iptables -A INPUT -i lo -j ACCEPT
#iptables -A OUTPUT -o lo -j ACCEPT

echo "Firewall Activated"
exit 0;
```

- Click **Update**
- If using a clustered pair, make the same changes to the firewall script (i.e. step 5) on the slave unit.




The VIP is now configured and will be accessible on ports 80 & 443.

EXAMPLE 2 – Setup a Firewall Mark by Modifying an Existing VIP

In this case, the VIP and the associated floating IP address will already exist so does not need to be created manually.

Step 1: Modify the Existing Virtual Service

- Using the WUI, go to *Cluster Configuration > Layer 4 – Virtual Services*
- Click **[Modify]** next to the relevant VIP
- Change the IP address to the chosen 'mark' value
- Clear the *Virtual Service Ports* field

Label	<input type="text" value="Server_Cluster"/>	
Virtual Server IP address	<input type="text" value="1"/>	
Virtual Server Ports	<input type="text"/>	

- Set the *Protocol* field to Firewall Marks
- Click **Update**

Step 2: Define a Health-Check Port

- Using the WUI, go to *Cluster Configuration > Layer 4 – Virtual Services*
- Click **[Modify]** next to the new Virtual Service
- Enter the appropriate value in the *Check Port* field
- Click **Update**

Step 3: Modify the Firewall Script

- Using the WUI, go to *Maintenance > Firewall Script*
- Uncomment / modify the example firewall marks section as shown in the following example. Additional ports can be added as required by adding additional iptables entries and specifying the appropriate port / protocol.


```

##### Manual Firewall Marks #####

# Example: Associate HTTP and HTTPS with Firewall Mark 1:
VIP1="192.168.100.1"
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

# A Virtual Service may then be created in the web interface, using 1 as the
# service address.

##### Packet Filtering #####

# You should always use a network perimeter firewall to lock down all
# external access to the load balancer except the required Virtual Services
# and the required services from your admin machine / network (SSH & HTTPS)

# Allow unlimited traffic on the loopback interface:
#iptables -A INPUT -i lo -j ACCEPT
#iptables -A OUTPUT -o lo -j ACCEPT

echo "Firewall Activated"
exit 0;

```

- Click **Update**

Firewall Mark Notes:

- When using firewall marks 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 if you are grouping port 80 and 443 traffic together you can only check one of these ports, typically this would be port 80.
- You can specify a range of ports rather than a single port as shown below:

```
iptables -t mangle -A PREROUTING -p tcp -d 10.141.12.34 -dport 1024:5000 -j MARK --set-mark 1
```

this specifies destination ports from 1024 to 5000

- You can leave the upper limit blank to use the default upper limit as shown below:

```
iptables -t mangle -A PREROUTING -p tcp -d 10.141.12.34 -dport 1024: -j MARK --set-mark 1
```

this specifies destination ports from 1024 to 65536

- You can specify a range of IP addresses as shown below:

```
iptables -t mangle -A PREROUTING -p tcp -m iprange -dst-range 10.141.12.34-10.141.12.40 --dport 80 -j MARK --set-mark 1
```

this specifies the destination IP address as a range from 10.141.12.34 to 10.141.12.40

Layer 4 – Advanced Configuration

This section allows you to configure the various layer 4 global settings.

Layer 4		
Lock Ldirectord Configuration	<input type="checkbox"/>	
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 Alert Source Address	<input type="text"/>	
Email Alert Destination Address	<input type="text"/>	
Auto-NAT	<input type="text" value="off"/> ▼	
Multi-threaded	<input type="text" value="yes"/> ▼	

Lock Ldirectord Configuration – Prevent the web interface from writing the Ldirectord configuration file, so that manual changes are retained. Manual changes to the Ldirectord configuration file may be overwritten if settings are edited in the web interface. Locking the configuration file will prevent the web interface from modifying the file, so that custom edits are preserved.

A warning message will be displayed on all Layer 4 configuration pages, and changes will be denied.

Warning: The Layer 4 configuration is set to read-only – changes made on this page will not be saved. Read-only mode may disabled on the [Advanced Configuration](#) page.

NOTE: If manual changes are made to configuration files, then *Lock Ldirectord Configuration* is unchecked any changes made via the WUI will overwrite the manual changes.

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 unexpected 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 set to **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 set to **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.

N.B. 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 Service 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 Services.

Layer 7 Services

The Basics

Layer 7 services are based on HAProxy which is a fast and reliable proxying and load balancing solution for TCP and HTTP-based applications.

Since HAProxy is a full proxy, Layer 7 services are not transparent by default, i.e. the client source IP address is lost as requests pass through the load balancer and instead are replaced by the load balancers IP own address.

Layer 7 supports a number of persistence methods including source IP address, HTTP cookie (both application based and inserted), RDP cookie and SSL session ID.

When a VIP is added the load balancer automatically adds a corresponding floating IP address which is activated instantly. 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.

Multiple ports can be defined per VIP, for example 80 & 443. In this case it may also be useful to enable persistence (aka affinity / stickiness) to ensure that clients hit the same back-end server for both HTTP & HTTPS traffic and also prevent the client having to renegotiate the SSL connection.









With Layer 7, port re-direction is possible, i.e. VIP:80 → RIP:800 is supported

Creating Virtual Services (VIPs)

Each Virtual Service 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). Typically you'll need one Virtual Service for each distinct cluster. Multiple ports can also be specified.


to add a new layer 7 VIP:

- In the WUI, open *Cluster Configuration > Layer 7 – Virtual Services*
- Click **[Add a New Virtual Service]**

Label	<input type="text" value="VIP Name"/>	
Virtual Service IP address	<input type="text" value="10.0.0.20"/>	
Virtual Service Ports	<input type="text" value="80"/>	
Persistence mode	<input type="text" value="None"/> 	
Fallback Server	<input type="text" value="127.0.0.1"/>	
Fallback Server Port	<input type="text" value="9081"/>	
Proxy Protocol	<input type="checkbox"/>	

- Enter an appropriate *Label* (name) for the new Virtual Service

- Enter the required IP address in the *Virtual Service IP address* field
- Enter the required ports(s) in the *Virtual Service Ports* field, separate multiple ports with commas, specify a range with a hyphen and specify all ports using an asterisk

 NOTE: the following ports are used by the appliance and therefore cannot be used for Virtual Services: 22 (SSH), 9080 (WUI – HTTP), 9443 (WUI – HTTPS), 7777 (HAProxy statistics page), 7778 (HAProxy persistence table replication and 9081 (nginx fallback page).

- Select the required *Persistence Mode*
 - HTTP Cookies* – Insert and use a cookie to ensure a client always hits the same server. In this mode, the inserted cookie name is set to be the same as the Real Server Label (name)
 - RDP Client Cookies* – Make sure a Terminal Server user always uses the same server. RDP Cookie relies on the mstshash cookie
 - MS Session Broker* – Make sure a Terminal Server user returns to the same server, even after the session has been disconnected. Reads the msts cookie
 - Source IP* – Make sure the same source IP always hits the same server
 - Source Hash* – Now deprecated
 - SSL Session ID* – Base persistence on SSL session ID. Note that this option is not always reliable because in some browsers (notably IE) the session ID can be renegotiated frequently which effectively breaks the persistence
 - None* – No persistence
- Set the Fallback server & port – This is where requests go if all servers in the cluster are down. The default is the local nginx instance running on the appliance, but this can also be any other appropriate host.
- Enable *Proxy Protocol* if you wish to use this VIP with STunnel for SSL off-load and pass the client's IP address to the Real Servers.

N.B. When using this option please also ensure that TProxy is enabled in the Layer7 Advanced options and that 'Set as Transparent Proxy' is enabled in your STunnel VIP.
- Click **Update**
- Now proceed to define the RIPs (Real Servers) as detailed on page 95


Modifying a Virtual Service

When first adding a Virtual Service, only certain values can be configured, others are set at their default setting. These values can be changed after the Virtual Service has been created by clicking [**Modify**] next to the relevant Virtual Service. Additional settings that can be changed are:

Option	Description
Layer 7 Protocol	<p>Select the Layer 7 protocol to be handled by this Virtual Service, either HTTP or Other TCP</p> <p>HTTP – Selected if the Virtual Service will handle only HTTP traffic. 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. In addition, HAProxy logs will show more information on the client requests and Real Server responses.</p> <p>Other TCP – Required for non HTTP traffic such as HTTPS, RDP etc.</p>

HTTP Close	<p>Enable or disable passive HTTP connection closing (Default: ON)</p> <p>When a client communicates with HAProxy it will only analyze, log and process the first request of each connection.</p> <p>This option is a requirement if your web server (and most do) has HTTP keepalive available.</p> <p>As HAProxy will only work with the first request within each keepalive connection this can lead to strange web page loading behavior. Only turn this off if you know what you are doing or have been advised to by the support engineers.</p>
Balance Mode	The scheduler used to specify server rotation. Specify the scheduler to utilize when deciding the back-end 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.
Table Size	The size of the table of connections in KB. The size of the table of connections (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.
Fallback Server Persistence	Configure the Fallback server to be persistent. During a health-check failure users can be forwarded to a fallback server. Setting this to on will make this server persistent so that when the Real Servers are put back in the pool, they will remain on the fallback server until their persistence times out. Setting this to off will move users to a Real Server as soon as one is available.
Check Port	Specify a different port for health checks. If specified this setting overrides the default check port, useful when you are balancing multiple ports.
Request to Send	<p>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.</p> <p>For example, if index.html was specified in this field, the following check directive would be automatically created in the HAProxy configuration file: <code>option httpchk GET /index.html HTTP/1.0</code></p> <p><i>(N.B. the back-slash character before 'index.html' is added automatically)</i></p>
Response Expected	<p>The content expected for a valid health check on the specified file. The response expected can be any valid regex statement.</p> <p>Continuing the example above, if the file index.html contained the word 'Copyright' response expected would be set to Copyright. The following check directive would then be automatically created in the HAProxy configuration file: <code>http-check expect rstring Copyright</code></p>
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	The name of a cookie used by the application running on the Real Servers. If set, this enables connection persistence based on an existing application cookie, ensuring that a client is always directed to the same Real Server. Note that this option requires the selection of HTTP Application Cookie persistence mode.
Application Cookie Length	The number of characters of the application cookie value to match. When storing and matching an Application Cookie value, the loadbalancer will use only the number of characters given here. If the cookie value is shorter than

	this maximum, only the actual length will be stored.
Application Cookie Hold Time	The time-out period before an idle application cookie is removed from memory. The application cookie will be removed from memory when it has been idle for longer than the Hold Time period. The default units are milliseconds.
Set X-Forwarded-For Header	Instruct HAProxy to add an X-Forwarded-For header to all requests, showing the client's IP Address. If HTTP is selected under Layer 7 Protocol, HAProxy is able to process the header of incoming requests. With this option enabled, it will append a new X-Forwarded-For header containing the client's IP Address. This information may be extracted by the Real Server for use in web applications or logging.
Proxy Protocol	Enable Proxy Protocol if using STunnel SSL Off-load. If you wish to use this VIP with STunnel for SSL off-load whilst passing the client's IP address to the real servers this option needs to be enabled (checked). Please ensure that TProxy is enabled in the Layer7 Advanced options and that the 'Set as Transparent Proxy' is enabled in your STunnel VIP.

 For more details on configuring health-checks please refer to Chapter 7.

Creating Real Servers (RIPs)

You can add an unlimited number of Real Servers to each Virtual Service (except the Enterprise R16 which is limited to 4 x VIPs each with up to 4 RIPs). For layer 7 VIPs port redirection is possible so the Real Server port field can be set to a different value to the VIP port. 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.

to add a new layer 7 RIP:

- In the WUI, open *Cluster Configuration > Layer 7 – Real Servers*
- Click **[Add a New Real Server]** next to the relevant Virtual Service



The screenshot shows a form with the following fields and values:

- Label:** RIP Name
- Real Server IP Address:** (empty)
- Real Server Port:** (empty)
- Weight:** 1

Each field has a question mark icon to its right. Below the fields is an 'Update' button.

- Enter an appropriate *Label* (name) for the new Real Server
- Enter the required IP address in the *Real Server IP Address* field
- Specify the required *Weight*, this 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

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 this can be enabled using the WUI.

to enable persistence state table replication:

- In the WUI, open *Cluster Configuration > Layer 7 – Advanced Configuration*
- Change *Persistence Table Replication* to **On**
- Click **Update**

N.B. This option is not available if you have IPv6 Virtual Services in your HAProxy configuration. Enabling this option will replicate persistence tables for all relevant layer 7 VIPs to the peer load balancer.

Layer 7 – Custom Configurations

It's possible to manually modify the HAProxy configuration file (/etc/haproxy/haproxy.cfg) to enable the appliance to support custom Layer 7 configurations. It's important to note that these manual changes can be overwritten under various circumstances:

- When changes are made to ANY Layer 7 VIP or RIP
- When a Layer 7 VIP is taken off/on line using System Overview

To prevent the configuration file being overwritten in these cases the Layer 7 advanced option *Lock HAProxy Configuration* can be used.

- In the WUI, open *Cluster Configuration > Layer 7 – Advanced Configuration*
- Enable *Lock HAProxy Configuration*
- Click **Update**
- The message shown below will be displayed:

Warning: The HAProxy configuration is set to read-only – changes made on this page will not be saved.
Read-only mode may disabled on the **Advanced Configuration** page.

Based on the above points, aim to configure Layer 7 services in the following order:

- Configure all standard Layer 7 services via the WUI
- Enable *Lock HAProxy Configuration*
- Now make the required custom changes to the HAProxy configuration file using your preferred editor



If you do manually add additional Virtual Services, don't forget to also add a Floating IP on the same address as the new VIP using *Cluster Configuration > Floating IPs*

Ex. 1 – Load Balancing based on URL match using ACL's in HAProxy

To support URL matched load balancing the structure of the HAProxy configuration file must be changed to use the front-end / back-end model as shown in the example below. This requires that the HAProxy file be modified manually which does have various implications as described at the start of this section.

```
# HAProxy configuration file generated by loadbalancer.org appliance
global
    daemon
    stats socket /var/run/haproxy.stat mode 600 level admin
    pidfile /var/run/haproxy.pid
    maxconn 40000
    ulimit-n 81000
    tune.maxrewrite 1024

defaults
    mode http
    balance roundrobin
    timeout connect 4000
    timeout client 42000
    timeout server 43000

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
```

As shown in the above example, instead of the usual 'listen' directive (which groups the Virtual Service and it's real back-ends together), we now have separate frontend and backend sections.

In this example:

'test_acl1' ← this is the name / label of the ACL

'path_beg' ← this means match the beginning of the path to a certain value, in this case '/test1'

and similarly for test_acl2

There are numerous matching options available. For more details and examples, please refer to:

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

then search that page for "Matching at Layer 7"



Don't hesitate to contact support@loadbalancer.org to discuss any specific requirements you may have.

Ex. 2 – HTTP to HTTPS Redirect using HAProxy & Pound (SSL Termination on the Appliance)

In this example the redirect directive is used to redirect connections on port 80 to the Pound SSL VIP listening port 443. This can be achieved by manually adding the 2 lines shown below in bold to the HAProxy configuration file:

```
listen VIP1
  bind 192.168.110.142:80
  mode http
  balance leastconn
  acl ACL-A src 192.168.110.142 ← see note 1
  redirect prefix https://192.168.110.142 if !ACL-A ← see note 2
  cookie SERVERID insert nocache indirect
  server backup 127.0.0.1:9081 backup non-stick
  option httpclose
  option forwardfor
  option redispatch
  option abortonclose
  maxconn 40000
  server rip1 192.168.70.195:80 weight 1 cookie rip1 check inter 2000 rise 2 fall 3
minconn 0 maxconn 0 on-marked-down shutdown-sessions
```

Steps:

1. Create a standard Pound / HAProxy VIP for SSL termination (for an example see page 168)
2. Using an editor such as vi or vim, add the two lines as shown in bold above (substituting the correct IP address)

N.B. Using this method, a floating IP address is automatically added when the VIP is created using the WUI. If you modify the HAProxy config file directly without using the WUI, make sure you also add a corresponding floating IP using: Edit Configuration > Floating IP's

Note 1

This line configures an acl named 'ACL-A', where the criteria for a match is that the source IP address must be 192.168.110.142.

Note 2

This line causes the redirect to https://192.168.110.142 to occur when the acl is not matched, i.e. for all traffic that is NOT coming from the Pound VIP.

This can also be a domain name entry such as:

```
redirect prefix https://www.loadbalancer.org if !ACL-A
```



Don't forget that any manual changes can be overwritten in various circumstances as explained in the start of this section.

Ex. 3 – HTTP to HTTPS Redirect using HAProxy (SSL Termination on the Real Server)

In this example a simple VIP is added which redirects inbound requests to another VIP that is listening on port 443.

```
listen VIP-80 192.168.110.178:80          ← see note 1
    redirect location https://192.168.110.178:443 ← see note 2
listen VIP-443
    bind 192.168.110.178:443
    mode tcp
    balance leastconn
    server backup 127.0.0.1:9081 backup non-stick
    option redispatch
    option abortonclose
    maxconn 40000
    server rip1 192.168.101.2:443 weight 1 check inter 2000 rise 2 fall 3 minconn 0 maxconn
0 on-marked-down shutdown-sessions
```

Steps:

1. Create a standard VIP with associated RIPv that listens on port 443 (VIP-443 in the above example)
2. Using an editor such as vi, add the two lines as shown in bold above (substituting the correct IP)

N.B. Using this method, a floating IP address is automatically added when the VIP is created using the WUI. If you modify the HAProxy config file directly without using the WUI, make sure you also add a corresponding floating IP using: Edit Configuration > Floating IP's

Note 1

An additional VIP (VIP-80) is added that listens on port 80.

Note 2

A redirect is implemented to redirect to the second VIP (VIP-443) on port 443. HTTPS traffic is then passed on to the Real Server.

This can also be a domain name entry such as:

```
redirect location https://www.loadbalancer.org
```



Don't forget that any manual changes can be overwritten in various circumstances as explained in the start of this section.

HAProxy Error Codes

For reference, HAProxy's own error codes are as follows:


























Code	When / Reason
200	access to stats, 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

For a complete HAProxy reference please refer to the following link:

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

Layer 7 – Advanced Configuration

This section allows you to configure the various layer 7 global settings.

Layer 7 (HAProxy):		
Lock HAProxy Configuration	<input type="checkbox"/>	
Logging	off 	
Log Only Errors	off 	
Redispatch	on 	
Connection Timeout	4000	
Client Timeout	42000	
Real Server Timeout	43000	
Maximum Connections	40000	
Ulimit		
Abort on Close	on 	
Transparent Proxy	off 	
Interval	2000	
Rise	2	
Fall	3	
Statistics Password		
Statistics Port		
Request buffer length		bytes 
Header buffer length		bytes 
Persistence Table Replication	off 	
Persistence Table Replication port		

Lock HAProxy Configuration – Prevent the WUI writing to the HAProxy configuration file. Manual changes to the HAProxy configuration file may be overwritten if settings are edited in the web interface. Locking the configuration file will prevent the web interface from modifying the file, so that custom edits are preserved. A warning message will be displayed on all Layer 7 configuration pages, and changes will be denied.

Warning: The HAProxy configuration is set to read-only – changes made on this page will not be saved.
Read-only mode may disabled on the [Advanced Configuration](#) page.



NOTE: If manual changes are made to configuration files, then *Lock HAProxy Configuration* is unchecked any changes made via the WUI will overwrite the manual changes.

Logging – Activate detailed logging of the Layer 7 HAProxy service. When activated the HAProxy log is written to `/var/log/haproxy.log`.

Log Only Errors – Do not log operational connection details, only log errors.

Redispatch – Allows HAProxy to break persistence and redistribute to working servers should failure occur. Normally this setting should not require changing.

Connection Timeout – HAProxy connection timeout in milliseconds. This setting should normally not require changing.

Client Timeout – HAProxy client timeout in milliseconds. This setting should normally not require changing.

Real Server Timeout – 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 Service (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 Service 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 Services 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 Services.

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 subnets) with the Real Servers using an IP address on the load balancer (preferably a floating IP) as their default gateway.

N.B. all Layer 4 methods are transparent by default



For more details on using TProxy, refer to pages 117-119.

N.B. 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 also possible to use the X-Forwarded-For header with layer 7 Virtual Services. Most web servers can then be configured to record the X-Forwarded-For IP address in the log files.

For details on how to enable X-Forwarded-For support, please refer to page 95.

For details on how to enable X-Forwarded-For support with Apache and IIS, please refer to the following Loadbalancer.org blog links:

Apache: <http://blog.loadbalancer.org/apache-and-x-forwarded-for-headers/>

IIS: <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*.

Statistics Port – Change the listening port for the HAProxy web based statistics report from the default 7777.

Request Buffer Length – Set the health check buffer length in bytes.

N.B. Changing this value will effect the performance of HAProxy. Do not make changes unless you know exactly what you are doing.

Lower values allow more sessions to coexist in the same amount of RAM, and higher values allow some applications with very large cookies to work. The default value is 16384 bytes. It is strongly recommended not to change this from the default value, as very low values will break some services such as statistics, and values larger than the default size will increase memory usage, possibly causing the system to run out of memory. Administrators should consider reducing the Maximum Connections parameter if the request buffer is increased.

Header Buffer Length – Set HAProxy header buffer length

Set the header buffer length, in bytes The header buffer is a section of the request buffer, reserved for the addition and rewriting of request headers. The default value is 1024 bytes. Most applications will only require a small header buffer, as few headers are added or rewritten.

Persistence Table Replication – When enabled, HAProxy's persistence tables are replicated to the slave device.

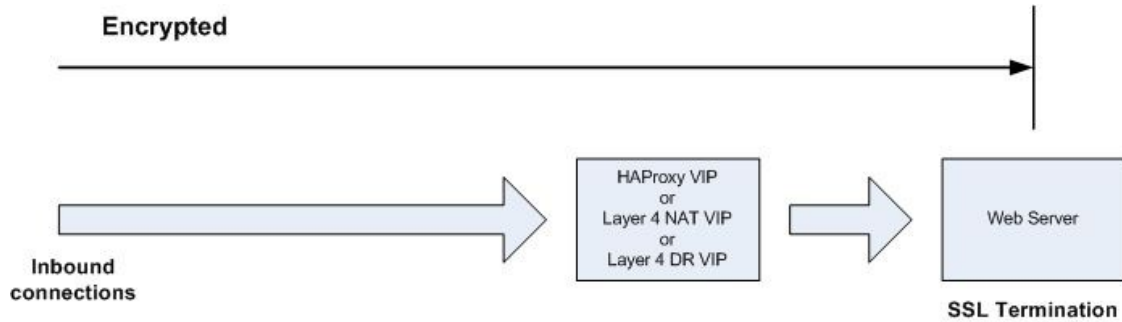
Persistence Table Replication Port – Set the TCP port to use for persistence table replication.

SSL Termination

Concepts

SSL termination can be performed on the Real Servers or on the load balancer (aka SSL offloading).

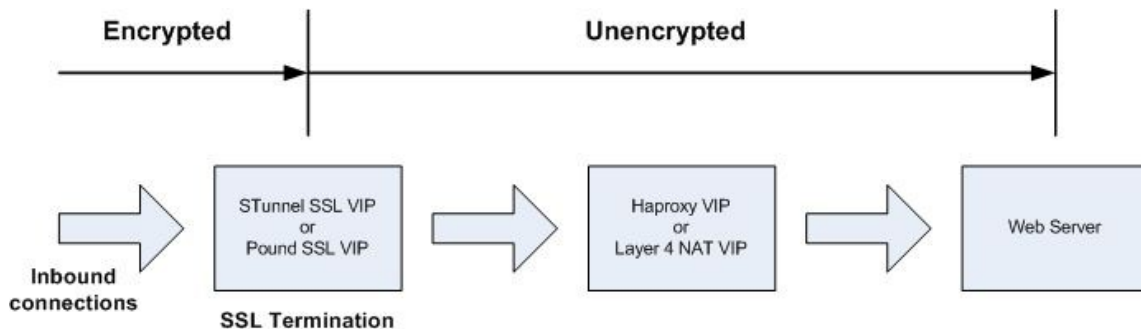
SSL Termination on the Real Servers:



NOTES:

- 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 – in this case the only option is source IP persistence

SSL Termination on the Load balancer:



NOTES:

- Since SSL is terminated on the load balancer, data from the load balancer to the web 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 web servers and your security requirements
- It's possible to use HTTP cookie based persistence
- A Pound or STunnel SSL Virtual Service is used to terminate SSL. The backend for the Virtual Service can be either a Layer 4 NAT mode Virtual Service or a Layer 7 HAProxy Virtual Service



NOTE: SSL termination on the load balancer can be very CPU intensive. In most cases, for a scalable solution, terminating SSL on the Real Servers is the best option.

SSL Termination on the Real Servers

In this case SSL certificates are installed on each Real Server in the normal way. The load balancer is then configured with a VIP that listens on HTTPS port 443 and distributes inbound requests to the Real Servers again on port 443 as shown in the layer 4 DR mode example below:

SSL	192.168.110.50	Port 443/tcp	Direct Routing	[Add a new Real Server]
SSL1	192.168.110.51		Weight 1	[Modify] [Delete]
SSL2	192.168.110.52		Weight 1	[Modify] [Delete]

A fairly common configuration is to include port 80 in the VIPs definition and also enable persistence. This ensures that both HTTP and HTTPS requests from a particular client are always sent to the same Real Server as shown below:

SSL	192.168.110.50	Ports 80,443/tcp	Direct Routing	[Add a new Real Server]
SSL1	192.168.110.51		Weight 1	[Modify] [Delete]
SSL2	192.168.110.52		Weight 1	[Modify] [Delete]

SSL Termination on the Load Balancer

In this case SSL certificate(s) must be installed on the load balancer. The appliance supports the use of both STunnel (default) and Pound for SSL termination.

To configure SSL termination on the appliance a Virtual Service must be defined that specifies an IP address and port to listen for inbound HTTPS connections and a back-end IP address / port where to forward the corresponding unencrypted HTTP connection.

By default a self-signed certificate is used for the new VIP which is ideal for testing but needs to be replaced for production deployments.

Creating an STunnel SSL Virtual Service (the Default SSL Terminator)

to add an STunnel SSL VIP:

- In the WUI, open *Cluster Configuration > SSL Termination*
- Click **[Add a New Virtual Service]**

Label	<input type="text" value="VIP Name"/>	?
Virtual Service IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Service Port	<input type="text" value="443"/>	?
Backend Virtual Service IP Address	<input type="text" value="10.0.0.20"/>	?
Backend Virtual Service Port	<input type="text" value="80"/>	?
Ciphers to use	<input type="text"/>	?
Do not insert empty fragments	<input type="checkbox"/>	?
SSL Terminator	<input type="radio"/> Pound <input checked="" type="radio"/> STunnel	?
Delay DNS Lookups	<input type="checkbox"/>	?
Disable SSLv2 Ciphers	<input checked="" type="checkbox"/>	?
Allow Client Renegotiation	<input checked="" type="checkbox"/>	?
Disable Renegotiation	<input type="checkbox"/>	?
Time To Close	<input type="text" value="0"/>	?
Set as Transparent Proxy	<input type="checkbox"/>	?

- Enter an appropriate *Label* (name) for the new Virtual Service
- Enter the required IP address in the *Virtual Service IP address* field
- Enter the required port in the *Virtual Service Port* field – typically 443
- Enter the required IP address in the Back-end *Virtual Service IP address* field

This is normally the same IP address as the Virtual Service IP address but can be any valid IP. The IP address specified must correspond to a Layer 7 HAProxy VIP or a Layer 4 NAT mode VIP. Unencrypted traffic will be sent here for load balancing.

N.B. DR mode cannot be used since STunnel acts as a proxy, and the Real Servers see requests with a source IP address of the Virtual Service. 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 STunnel.

- Enter the required port in the Back-end *Virtual Service Port* field
- Define the list of accepted ciphers using the *Ciphers to use* field

Leave blank for the default of any cipher. If you wish to restrict the ciphers that STunnel should negotiate with the client, they may be specified here. If the field is left blank, STunnel 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 are shown below:

- * SSLv3
- * TLSv1
- * SSLv3:HIGH
- * AES128-SHA:DES-CBC3-SHA:RC4-SHA:RC4-MD5:!SSLv2

- **Configure *Do not Insert Empty Fragments***

Disables a countermeasure against a SSL 3.0/TLS 1.0 protocol vulnerability affecting CBC ciphers. This option needs to be enabled (checked) to ensure mitigation of both the BEAST and CRIME MITM attacks. It is also required for PCI Testing.

- **Ensure *SSL Terminator* is set to **STunnel****

- **Configure *Delay DNS Lookup***

Delay DNS lookup for 'connect' option. This option is useful for dynamic DNS, or when DNS is not available during STunnel startup (road warrior VPN, dial-up configurations).

- **Configure *Disable SSLv2 Ciphers***

When ticked this option disables all SSLv2 Ciphers by using the OpenSSL 'SSL_OP_NO_SSLv2' option.

- **Configure *Allow Client Renegotiation***

Sets whether the client is allowed to renegotiate the cipher order. This option should be enabled (checked) to mitigate the BEAST attack.

- **Configure *Disable SSL Renegotiation***

Applications of the SSL renegotiation include some authentication scenarios, or re-keying long lasting connections. On the other hand this feature can facilitate a trivial CPU-exhaustion DoS attack. This option should be enabled (checked) to mitigate the BEAST Attack.

- **Configure *Time to Close***

Configure the global client response timeout in seconds.

- **Configure *Set as Transparent Proxy***

If you wish to use HAProxy and TProxy this option needs to be enabled (checked) to allow SSL termination on the load balancer whilst passing the client's IP address to the Real Servers. This option only enables TProxy on a Single STunnel VIP – if you're using HAProxy with this VIP you will also need to enable TProxy for your HAProxy VIP (please refer to the examples on pages 117-119)

STunnel Cipher Settings and the BEAST Attack

The following STunnel options should be set to mitigate the BEAST attack:

Ciphers to use – a minimum cipher list of 'RC4:HIGH:!MD5:!aNULL' is required

Allow Client Renegotiation – this option should be disabled (un-checked)

Do Not Insert Empty Fragments – this option should be enabled (checked)

Disable SSL Renegotiation – this option should be enabled (checked)

If these options are set, this should prevent the BEAST attack, and should also help to mitigate DoS attacks and MITM Attacks.

Creating a Pound SSL Virtual Service

to add a Pound SSL VIP:

- In the WUI, open *Cluster Configuration > SSL Termination*
- Click **[Add a New Virtual Service]**

Label	<input type="text" value="VIP Name"/>	?
Virtual Service IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Service Port	<input type="text" value="443"/>	?
Backend Virtual Service IP Address	<input type="text" value="10.0.0.20"/>	?
Backend Virtual Service Port	<input type="text" value="80"/>	?
Ciphers to use	<input type="text"/>	?
Do not insert empty fragments	<input type="checkbox"/>	?
SSL Terminator	<input checked="" type="radio"/> Pound <input type="radio"/> STunnel	?
Enable WebDAV Verbs	<input type="checkbox"/>	?
Rewrite HTTP Redirects	<input checked="" type="checkbox"/>	?
Honor Cipher Order	<input type="checkbox"/>	?
Allow Client Renegotiation	<input type="text" value="No Client Renegotiation"/>	?
Disable SSLv2 Ciphers	<input checked="" type="checkbox"/>	?
Disable SSL Compression	<input checked="" type="checkbox"/>	?

- Enter an appropriate *Label* (name) for the new Virtual Service
- Enter the required IP address in the *Virtual Service IP address* field
- Enter the required port in the *Virtual Service Port* field – typically 443
- Enter the required IP address in the Back-end *Virtual Service IP address* field

This is normally the same IP address as the Virtual Service IP address but can be any valid IP. The IP address specified must correspond to a Layer 7 HAProxy VIP or a Layer 4 NAT mode VIP. Unencrypted traffic will be sent here for load balancing.

N.B. 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 Service. 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.

- Enter the required port in the Back-end *Virtual Service Port* field
- Define the list of accepted ciphers using the *Ciphers to use* field

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 are shown below:

- * SSLv3
- * TLSv1
- * SSLv3:HIGH

Enter the required IP address in the Back-end Virtual Service IP address field

- Configure *Do Not Insert Empty Fragments*

Disables a countermeasure against a SSL 3.0/TLS 1.0 protocol vulnerability affecting CBC ciphers. This option needs to be enabled (checked) to ensure mitigation of both the BEAST and CRIME MITM attacks. It is also required for PCI Testing.

- Ensure *SSL Terminator* is set to **Pound**

- Configure *Enable WebDAV Verbs*

When enabled extends which HTTP / WebDAV verbs are accepted.

- Configure *Rewrite HTTP Redirects*

Pound to change the Location: and Content-location: headers in responses If they point to the back-end itself or to the listener (but with the wrong protocol) the response will be changed to show the virtual host in the request. NOTE: If you do not know what this means leave this as the default (enabled).

- Configure *Honor Cipher Order*

When choosing a cipher during a handshake, normally the client's preference is used. If this directive is enabled, the server's preference will be used instead. When choosing a cipher during a SSLv3 or TLSv1 handshake, normally the client's preference is used. If this directive is enabled, the server's preference will be used instead.

This option should be enabled to mitigate the BEAST attack.

- Configure *Allow Client Renegotiation*

Sets whether the client is allowed to renegotiate the cipher order. In Pound when set to either:

- No Client Renegotiation, no client renegotiation will be honored
- Secure Renegotiation, secure renegotiation will be honored
- Insecure Renegotiation, insecure renegotiation will be honored

This option should be set to 'No Client Renegotiation' to mitigate the BEAST attack.

- Configure *Disable SSLv2 Ciphers*

Allow the option to Disable all SSLv2 Ciphers. When ticked this option disables all SSLv2 Ciphers by using the OpenSSL 'SSL_OP_NO_SSLv2' option.

- Configure *Disable SSL Compression*

Disable DEFLATE compression even if both server and client supports it. If this option is enabled (checked), the server will disable DEFLATE compression even if both server and client supports it. In case compression is enabled an attacker with access to encrypted network traffic can conduct a "CRIME" attack by making client issue requests with specific character sequences and observing whether they got compressed or not, indicating their presence in part of the request that is not under his control (e.g. cookie headers).

Modifying a Pound SSL Virtual Service

When first adding a Pound SSL Virtual Service, only certain values can be configured, others are set at their default setting. These values can be changed after the Virtual Service has been created by clicking [**Modify**] next to the relevant Virtual Service. Additional settings that can be changed are:

Option	Sub-Option	Description
Headers	Header Field Name	Add your own header to be passed on by Pound. Set Field Name allows the name part of the header to be specified: [field-name]: [field-value]
	Header Field Value	Add your own header to be passed on by Pound. Set Field Value allows the value part of the header to be specified: [field-name]: [field-value]

Pound Cipher Settings and the BEAST Attack

The following Pound options should be set to mitigate the BEAST attack:

Ciphers to use – a minimum cipher list of 'RC4:HIGH:!MD5:!aNULL' is required

Honor Cipher Order – this option should be enabled (checked)

Allow Client Renegotiation – this option should be set to 'No Client Renegotiation'

Do not Insert Empty Fragments – this option should be enabled (checked)

If these options are set, this should prevent the BEAST attack, and should also help to mitigate DoS attacks and MITM Attacks.

Generating a CSR on the Load Balancer

By default, when creating an SSL Virtual Service a self-signed certificate is used. This is ideal for testing but needs to be replaced for production 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 generate a CSR:

- In the WUI, open *Cluster Configuration > SSL Termination*
- Click [**Certificate**] next to the relevant Virtual Service
- Complete the fields as shown in the example below:

Country code (C)	Great Britain (UK)	?
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	?
CSR Key Length	1024 bits	?

- Click **Generate SSL Certificate Request**

Certificate Signing Request	<pre>-----BEGIN CERTIFICATE REQUEST----- MIIB7DCCAUCQAQAwgasxCzAJBgNVBAYTAkdCMRIwEAYDVQQIDAlIYW1wc2hpcmUx EzARBgNVBAcMC1BvcnRzbW91dGgxGTAXBgNVBAoMEEXvYWRiYWxhbmNlci5vcmcx EDA0BgNVBAcMB1N1cHBvcnQxHTAbBgNVBAMMFHd3dy5sb2FkYmFsYW5jZlIub3Jn MScwJQYJKoZIhvcNAQkBFhhzdXBwb3J0QGxvYWRiYWxhbmNlci5vcmcwZ8wDQYJ KoZIhvcNAQEBBQADgY0AMIGJAoGBAOTF5bB381q3ru0gPjH9tLGSYq+sqRXyYftW WNLh2GFSsGWVKdzju6B7N4KvTuPdKAmsischQ1t/fYQ7T5SAEeBH8e37AGW3sBKl hqjeHIeNvOwyfSwaxE0/7InuAz23UNLSIhxwtLOArN1+AM1eegXc3A/R9/o8uym/</pre>
Signed Certificate from CA	<p>Paste your signed certificate here.</p>


- Copy the resulting CSR from the top pane and send this to your chosen Certificate Authority

N.B. Select Apache as the platform type during the certificate generation process.

- Once you receive your signed certificate from the CA, copy/paste this into the lower pane

Certificate	-----BEGIN CERTIFICATE REQUEST----- MIIB7TCCAVYCAQAwwaxCzAJBgNVBAYTAkdCMRIwEAYDVQQIDAlIYW1wc2hpcmUx EzARBgNVBAcMClBvcnRzbW91dGgxGTAXBgNVBAoMEExvYWRiYWxhbmN1ci5vcmcx ETAPBgNVBAcMCFN1cHBvcnQxMR0wGwYDVQQDDBR3d3cubG9hZGJhbGFuY2VyLm9y
Signing Request	ZzEnMCUGCSqGSIB3DQEJARYYc3VwcG9ydEBSb2FkYmFsYW5jZXIub3JnMIGfMA0G CSqGSIB3DQEBAQUAA4GNADCBiQKBgQDCOJ/oLS99DWNnyWwAfzP1YTuswqOz4IkV 0oYSgI8oSKzuwauo9QAb18crGYcPf4Pp9+0f0S18DhseafI58QePxdIAOTHoNxz sJRwbCXGym6CBLSCg4e7rMxScMByN5b+HRXIXJ/pKfWVuszvnS/roEyzYNHC8153 IaSBMnK2LwIDAQABAAAwDQYJKoZIhvcNAQEFBQADgYEAe0Jkxrm2SrHI29WM+Iva
Signed Certificate from CA	-----BEGIN CERTIFICATE----- MIIFGzCCBAOgAwIBAgIQHKIm6FlrJ3Qk20Cfiv7v/zANBgkqhkiG9w0BAQUFADCB yzELMAKGA1UEBhMCVVMxZmFzAVBgNVBAoTDlZlcm1TaWduLCBjbmuMTAwLgYDVQQQL EydGb3IqVGZzdCBQdXJwb3N1cyBFBm5LiAgTm8gYXNzdXJhbmN1cy4xQjBAbgNV BAAsTOVRlcm1zIG9mIHVzZSBhdCBodHRwczovL3d3dy52ZXJpc2lubi5jb20vY3Bz L3Rlc3RjYSAoYykwOTEtMCsGA1UEAxMkVmVyaVNpZ224gVHJpYWwgU2VjdXJlIFN1 cnZlciBDQSAtIEcyMB4XDTEzMDIyODAwMDAwMFoXDTEzMDMzMDIyNTk1OVowgb8x CzAJBgNVBAYTAkdCMRIwEAYDVQQIEw1IYW1wc2hpcmUxEzARBgNVBAcUC1BvcnRz bW91dGgxGTAXBgNVBAoUEExvYWRiYWxhbmN1ci5vcmcxETAPBgNVBAcUCFN1cHBv

- Then click **Upload Signed Certificate**

 If you need to add intermediate certificates to the chain, this can be done by appending these certificates at the end 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 & Uploading to the Appliance

- 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 (*truncated versions are shown*):

```
-----BEGIN CERTIFICATE-----
MIICDCCAhmGAWIBAgIJAL98jHEiUm3iMA0GCSqGSIB3DQEBBQUAMEUxCzAJBgN
E89UJCG2nMW5JBnkyHYbQTVU8MeR3ilhe2fw+qVE2pgxWYyWaGm8QwTsxQKgbx
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajilSfE
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwlm3HyovxjrymusOeIFgZiWyuablrreCplo+iydRf
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajilSf
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HUIrTqwES4Lunff
-----END RSA PRIVATE KEY-----
```


- Save the file
- Using the WUI open *Cluster Configuration > SSL Termination*
- Click [**Certificate**] next to the relevant Virtual Service
- Navigate to the bottom of the screen, then using the browse option select the file just created

Upload prepared PEM file

C:\certs\pem.txt

- Click **Upload PEM file**
- Now restart Pound using the restart link at the top of the page or via the WUI option: *Maintenance > Restart Services* and clicking **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*.

Adding an Intermediate Certificate

Certificate authorities may require that an intermediate certificate is installed. This can be in a number of ways:

1. When the certificate received from the CA is copied into the lower pane, also copy in the intermediate as mentioned in the previous section.
2. After the certificate is configured an intermediate certificate can be added using the WUI option: *Cluster Configuration > SSL Termination > [Certificate]* and pasting the intermediate certificate at the end of the lower pane, then clicking **Upload Signed Certificate**
3. If uploading a PEM file, the intermediate certificate can be copied to the end of the file as shown below prior to upload:

```
-----BEGIN CERTIFICATE-----
MIICsDCCAhmGAWlBAglJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxOzAJBGN
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajlSfE
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAABKQCCcPYkYHm8gYwlm3HyoVxjrymusOeIFgZiWyuablrreCplo+iydRf
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlGCTVYd3eo/Dv/KZ16p4HUIrTqwES4Lunff
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIICsDCCAhmGAWlBAglJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxOzAJBGN
E89UJCG2nMW5JvBNkyHYbQTvU8MeR3ilhe2fw+qVE2pgxWYyWaGm8QwTsxQKgbx
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajlSfE
-----END CERTIFICATE-----
```

Converting between certificate formats

OpenSSL can be used to convert between various certificate formats. This is usually included by default in Linux distributions. For Windows, it can be freely downloaded from the following location:

<http://slproweb.com/products/Win32OpenSSL.html>

At this URL you'll need to download and install the Visual C++ 2008 Redistributable, then download either the light or full version of OpenSSL. Once installed, you'll have an OpenSSL directory located on your filesystem (default location c:\OpenSSL)

To use the program, open a command window, navigate to the location where it was installed (by default c:\OpenSSL\bin) then run the required command.

Converting .pfx certificates to PEM format

Using Windows:

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

e.g.

```
openssl pkcs12 -in c:\cert.pfx -nodes -out c:\cert.pem
```

Using the Appliance / Linux:

```
openssl pkcs12 -in /path/filename.pfx -nodes -out /path/filename.pem
```

e.g

```
openssl pkcs12 -in /root/cert.pfx -nodes -out /root/cert.pem
```

Converting .cer certificates to PEM format

Using Windows:

```
openssl x509 -in drive:\path\filename.cer -inform DER -out drive:\path\filename.pem -outform PEM
```

e.g

```
openssl x509 -in c:\cert.cer -inform DER -out c:\cert.pem -outform PEM
```

Using the Appliance / Linux:

```
openssl x509 -in /path/filename.cer -inform DER -out /path/filename.pem -outform PEM
```

e.g

```
openssl x509 -in /root/cert.cer -inform DER -out /root/cert.pem -outform PEM
```

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 OpenSSL command either on the load balancer or another machine with openssl installed:

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

SSL – Advanced Configuration

Pound Global Settings

Pound Global Settings		
Lock Pound Configuration	<input type="checkbox"/>	?
Logging	Off ▾	?
Client Timeout	30	?
Global Server Timeout	60	?
Ulimit	<input type="text"/>	?
Transparent Proxy	Off ▾	?

Lock Pound Configuration – When enabled it will stop the user interface overwriting the configuration files so manual changes can be made.

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.

Ulimit – Set Ulimit value for Pound process. This setting will change the maximum number of file descriptors available to the Pound process. The default is 81000.

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.



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. One way to get around this is to configure the HAProxy VIP to listen on 2 ports. 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. This is covered on page 118.

STunnel Global Settings

STunnel Global Settings

Debug Level 

Debug Level – Option to set the debugging level for all STunnel Services. The Debug Level is a one of the syslog level names or numbers emerg (0), alert (1), crit (2), err (3), warning (4), notice (5), info (6), or debug (7). The higher the number the more detail will be contained in the STunnel Logs.

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 added automatically when new Virtual Services are created.

It's also possible to manually define the Floating IP(s) if required, this is normally only required when using layer 4 NAT mode or when using TProxy where in both cases the load balancer must be the default gateway for the Real Servers.

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) at any time.

FLOATING IPS

192.168.111.110	[Delete]
192.168.111.111	[Delete]
192.168.111.112	[Delete]
192.168.111.115	[Delete]
192.168.111.116	[Delete]
192.168.111.118	[Delete]

New Floating IP

Add Floating IP

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.



IMPORTANT: When using a clustered pair, ensure that the slave also has a static IP address assigned that's in the same subnet as the floating IP being added. Failure to do so will result in heartbeat issues during a failover.



NOTE: Floating IPs are not deleted automatically when Virtual Services are removed or modified, this must be done manually.

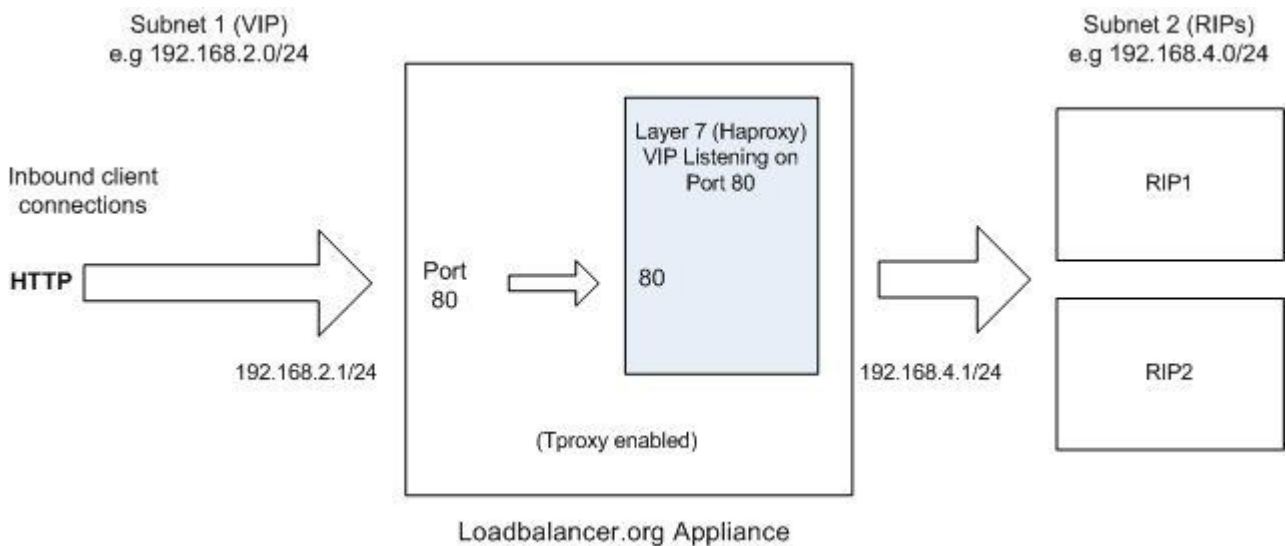
Using Transparent Proxy (TProxy)

HAProxy, Pound and STunnel are proxies which means that a new connection is established from the proxy out to the back-end server in response to an inbound client connection to the proxy. This means that the source IP address of the packet reaching the server will be the proxies address, or more specifically the IP address assigned to the load balancers Ethernet interface.

TProxy can be used with HAProxy, Pound and STunnel to maintain the actual source IP address of the client. When enabling TProxy, it's important to be aware of the topology requirements for TProxy to work correctly. This is covered in the examples below.

TProxy & HAProxy

In this example, TProxy is enabled with a layer 7 Virtual Service. This setup is illustrated in the following diagram.

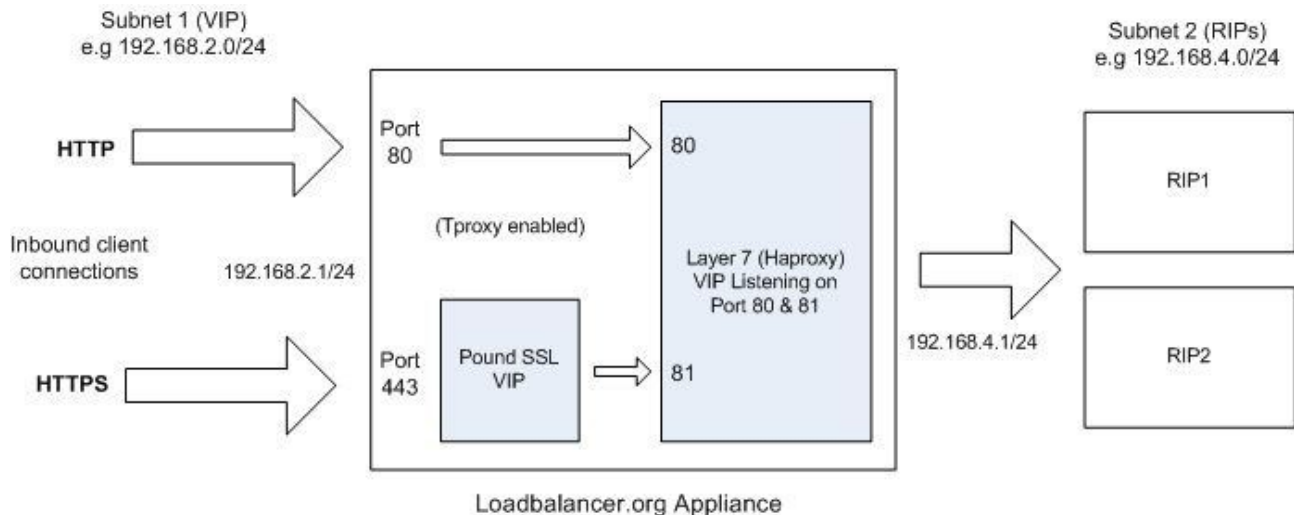


Topology Requirements / Notes

- 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 (in the above example, eth1 = 192.168.2.1 and eth0 = 192.168.4.1)
- TProxy must be enabled using the WUI option: *Cluster Configuration > Layer 7 – Advanced Configuration* and setting **Transparent Proxy** to 'On'
- On the Real Servers, the default gateway must be configured to be an IP address on the load balancer. When using a clustered pair, this should be a floating IP to allow failover to the slave

TProxy, HAProxy & Pound

In this example, Pound is used to terminate SSL. Pound passes the decrypted traffic to a layer 7 back-end VIP where the Real Servers are configured. This setup is illustrated in the following diagram.



N.B. Using STunnel rather than Pound in this scenario is not supported. For STunnel, 2 separate HAProxy VIPs must be used as explained on the following page.

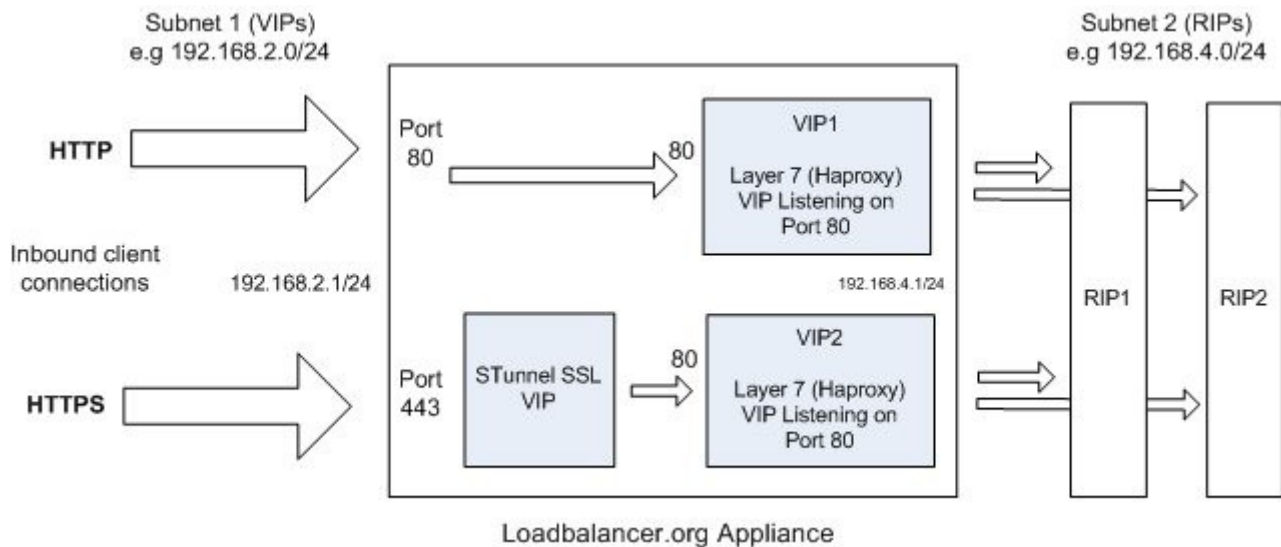
Topology Requirements / Notes

- 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 (in the above example, eth1 = 192.168.2.1 and eth0 = 192.168.4.1)
- Configure the Layer 7 VIP to listen on 2 ports – e.g. 80 & 81, then use port 81 for the Pound back-end and port 80 for client connections. Configure the Pound VIP to listen on the same IP address / port 443 and set its back-end to be port 81 of the HAProxy VIP.
This way, clients connect to a single IP address listening on port 80 & 443.
- TProxy for HAProxy must be enabled using the WUI option: *Cluster Configuration > Layer 7 – Advanced Configuration* and setting **Transparent Proxy** to 'On'
- TProxy for Pound must be enabled using the WUI option: *Cluster Configuration > SSL – Advanced Configuration* and setting **Transparent Proxy** to 'On'
- On the Real Servers, the default gateway must be configured to be an IP address on the load balancer. When using a clustered pair, this should be a floating IP to allow failover to the slave

TProxy, HAProxy & STunnel

In this example, STunnel is used to terminate SSL. STunnel passes the decrypted traffic to a layer 7 back-end VIP where the Real Servers are configured. As mentioned in the previous section, when STunnel is used, 2 separate HAProxy VIPs are required. This setup is illustrated in the following diagram.

N.B. If you require a single IP address with persistence across both ports 80 and 443, use the TProxy/HAProxy/Pound configuration described on the previous page.



Topology Requirements / 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 (in the above example, eth1 = 192.168.2.1 and eth0 = 192.168.4.1)
- Configure each Layer 7 VIP to listen on 1 port – e.g. port 80. Then configure the same Real Servers for both VIPs
- TProxy for HAProxy must be enabled using the WUI option: *Cluster Configuration > Layer 7 – Advanced Configuration* and setting **Transparent Proxy** to 'On'
- For VIP2, TProxy for STunnel must be enabled by checking the **Proxy Protocol** option when creating or modifying the VIP
- For the STunnel VIP, TProxy must be enabled by checking the **Set as Transparent Proxy** option when creating or modifying the VIP
- On the Real Servers, the default gateway must be configured to be an IP address on the load balancer. When using a clustered pair, this should be a floating IP to allow failover to the slave

Server Feedback Agent

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 Services 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 value in the range 0 – 100.

The load balancer typically expects a 0-99 integer response from the agent which relates to the CPU idle state, 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.

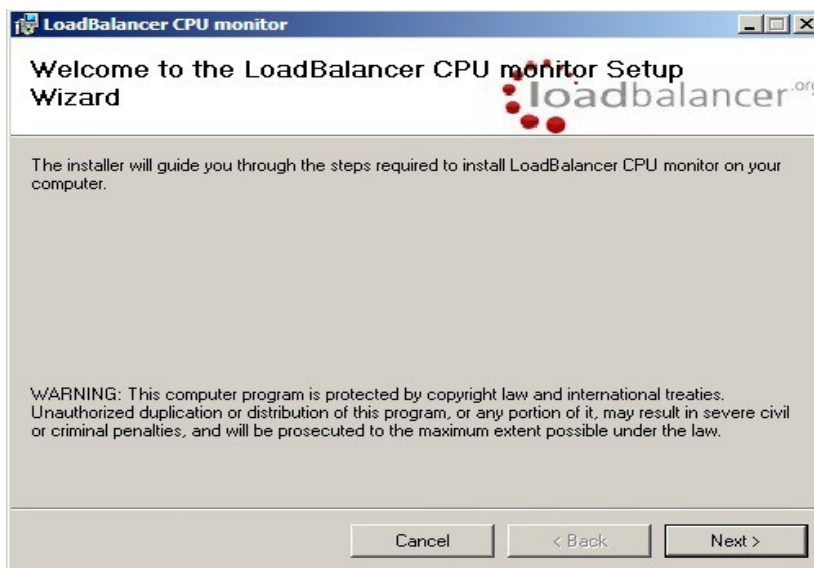
N.B. The feedback agent will never take a server offline, only the standard health check can do this.

Windows Agent

The Windows feedback agent can be downloaded from:

<http://downloads.loadbalancer.org/agent/windows/LBCPUMonInstallation.msi>

To install the agent, run LBCPUMonInstallation.msi on each server

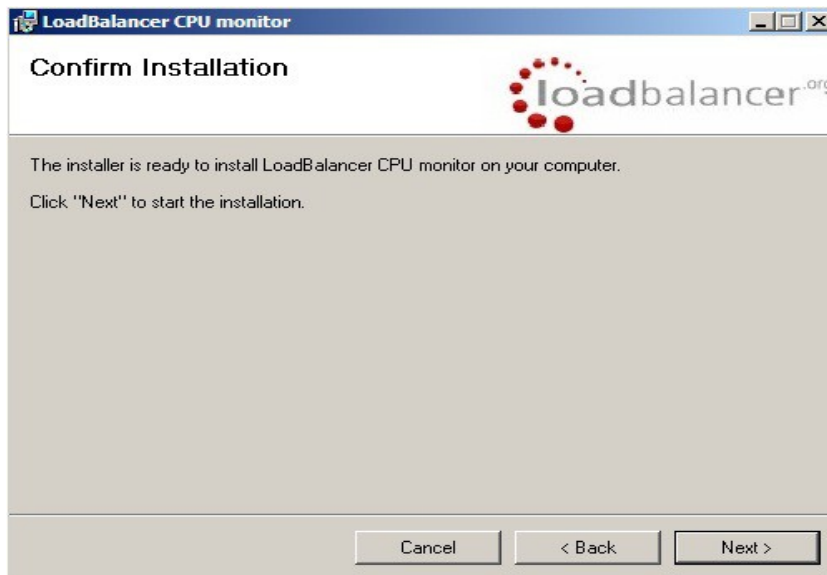


Click **Next**

N.B. The agent should be installed on all Real Serves in the cluster



Select the installation folder and click **Next**

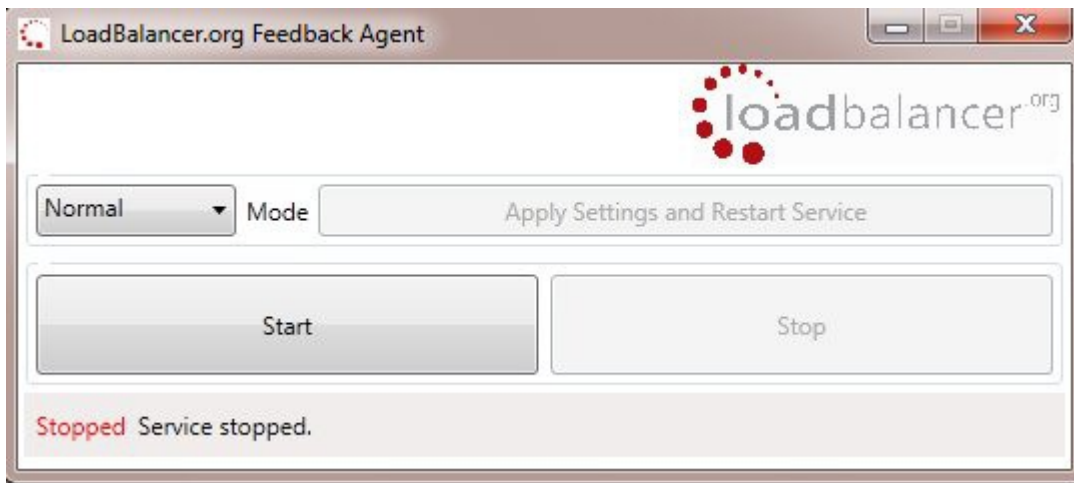


Click **Next** to start the installation

N.B. .NET Framework v3.5 is required by the agent and .NET Framework v4.0 is required by the Monitor

Starting the Agent

Once the installation has completed, you'll need to start the service on the Real Servers. The service is controlled by the Feedback Agent Monitor program that is also installed along with the Agent. The monitor can be accessed on the Windows server using: *All Programs > Loadbalancer.org > Monitor*. It's also possible to start the service using the services snap-in – the service is called 'Loadbalancer CPU monitor'.



- To start the service, click **Start**
- To stop the service, click **Stop**

Linux / Unix Agent

The Linux feedback agent files are available in 2 versions, the file locations for both versions are:

v3.1 (for appliances prior to v7.6)

readme file: <http://downloads.loadbalancer.org/agent/linux/v3.1/readme.txt>
 xinetd file: <http://downloads.loadbalancer.org/agent/linux/v3.1/lb-feedback>
 feedback script: <http://downloads.loadbalancer.org/agent/linux/v3.1/lb-feedback.sh>

v4.1 (for appliances v7.6 and later)

readme file: <http://downloads.loadbalancer.org/agent/linux/v4.1/readme.txt>
 xinetd file: <http://downloads.loadbalancer.org/agent/linux/v4.1/lb-feedback>
 feedback script: <http://downloads.loadbalancer.org/agent/linux/v4.1/lb-feedback.sh>

Installation

N.B. The agent files must be installed on all Real Servers, not the load balancer.

```
# Install xinetd
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.
```

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.

To Configure Virtual Services to use Agent / HTTP Feedback

- Go to *Cluster Configuration > Layer 4 – Virtual Services*
- Click **[Modify]** next to the Virtual Service
- Change the Feedback Method to **Agent** or **HTTP** as required

- Click **Update**

System Overview – Monitoring Server Weights

Prior to installing & activating the agent the *System Overview* would look similar to the following. Server weights are shown at the default of value of 1.

Label: HTTP-Cluster IP: 192.168.110.230 Method: Layer 4 Ports: 80 Mode: DR Protocol: TCP							
Rip Label	IP	Ports	Weight				
rip1	192.168.110.237	80	1	Drain	Halt		
rip2	192.168.110.238	80	1	Drain	Halt		

Once the agents are installed on the Real Servers and the feedback method is changed, the weights are updated:

*N.B. If the servers weights were set to 10 before the agent was installed, they would show an initial weight of 100 under no load once the agent is installed. This is calculated as $(100/10)*10$.*

Label: HTTP-Cluster IP: 192.168.110.230 Method: Layer 4 Ports: 80 Mode: DR Protocol: TCP							
Rip Label	IP	Ports	Weight				
rip1	192.168.110.237	80	10	Drain	Halt		
rip2	192.168.110.238	80	10	Drain	Halt		

If one of the Real Servers is heavily loaded, the weight is adjusted accordingly – a lower weight causes less sessions for that server. Here, CPU utilization on rip2 is high so the weight has been automatically reduced to 1:

Label: HTTP-Cluster IP: 192.168.110.230 Method: Layer 4 Ports: 80 Mode: DR Protocol: TCP							
Rip Label	IP	Ports	Weight				
rip1	192.168.110.237	80	10	Drain	Halt		
rip2	192.168.110.238	80	1	Drain	Halt		

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.

Layer 4

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

Add a TCP based Virtual Service & 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 Service & 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 Service:

```
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 Service 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             Mass     1         0         0
```

```
  -> 192.168.70.133:80             Mass     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
```

To view HAProxy's running configuration:

```
echo "show info" | socat unix-connect:/var/run/haproxy.stat stdio
```

To clear HAProxy's statistics:

```
echo "clear counters all" | socat unix-connect:/var/run/haproxy.stat stdio
```

N.B. 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>



IMPORTANT – Please note that since these changes are being made directly to the running configuration, the services that are displayed in the System Overview will no longer match the running configuration.



For additional assistance don't hesitate to contact: support@loadbalancer.org.

Chapter 7 – Real Server Health Monitoring

Introduction


The appliance supports a range of health-check options to check and verify the health of Real Servers. These range from simple ping checks to more complex negotiate options to determine that the underlying daemon / service is running. The specific options available depend on whether services are deployed at Layer 4 or Layer 7, details of both are covered in the following sections.


Real Server health Monitoring – Using System Overview

The System Overview is available in the WUI. It includes a visual display indicating the health status of all Virtual and Real Servers as shown in the example below:

SYSTEM OVERVIEW

2013-03-21 10:18:43 UTC

Label: RDP-Cluster IP: 192.168.100.102 Method: Layer 7 Ports: 3389 Mode: Proxy Protocol: OTHER TCP 


Label: HTTP-Cluster IP: 192.168.111.100 Method: Layer 4 Ports: 80 Mode: DR Protocol: TCP 





Key: Cluster healthy Cluster needs attention Cluster is down Real Server taken offline


Clicking on each Virtual Service expands the view so that the associated Real Servers can also be seen:





SYSTEM OVERVIEW

2013-03-21 10:19:43 UTC

Label: RDP-Cluster IP: 192.168.100.102 Method: Layer 7 Ports: 3389 Mode: Proxy Protocol: OTHER TCP 

Rip Label	IP	Ports	Weight				
TS1	192.168.110.238	3389	1	Drain	Halt		
TS2	192.168.110.237	3389	1	Drain	Halt		

Label: HTTP-Cluster IP: 192.168.111.100 Method: Layer 4 Ports: 80 Mode: DR Protocol: TCP 

Rip Label	IP	Ports	Weight				
IIS1	192.168.110.237	80	1	Drain	Halt		
IIS2	192.168.110.238	80	1	Drain	Halt		

Key: Cluster healthy Cluster needs attention Cluster is down Real Server taken offline

The various colors used to indicate status are:

- **Green** – All Real Servers in the cluster are healthy
- **Yellow** – One or more Real Servers in the cluster has failed or has been taken offline using *Halt* or *Drain*
- **Red** – All Real Servers in the cluster have failed
- **Blue** – All Real Servers have been taken offline using *Drain* or *Halt* (see below)

To control the status of each Real Server, the *Drain* and *Halt* options can be used:

- **Drain** – This option allows existing connections to close gracefully and prevents new connections
- **Halt** – This options prevents new connections and drops all existing connections immediately without waiting



NOTE: If you drain or halt all the Real Servers at layer 4, the fallback server will NOT be activated, the fallback server only comes into effect when all servers fail their health-check. At layer 7, the fallback page is displayed when either all servers fail or when all servers are taken offline.

Layer 4 Services

At layer 4, Real Server health checking is provided by Ldirectord. This is integrated into Loadbalancer.org appliances and allows a full range of options to check that Real Servers are operational.

To configure health checks use the WUI option: *Cluster Configuration > Virtual Services > Modify*

Health Checks		
Check Type	<input type="text" value="Connect to port"/>	
Check Port	<input type="text"/>	
External Script command	<input type="text"/>	
Negotiate Check Options		
Negotiate Check Service	<input type="text" value="HTTP"/>	
Virtual Host	<input type="text"/>	
Database Name	<input type="text"/>	
Radius Secret	<input type="text"/>	
Login	<input type="text"/>	
Password	<input type="text"/>	
Request to send	<input type="text" value="check.txt"/>	
Response expected	<input type="text" value="OK"/>	

Default Health Check

By default, a TCP connect health check is used for newly created layer 4 Virtual Services.

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 it's 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 file path 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 check

10 Connects, 1 Negotiate – Do 10 connect checks and then 1 negotiate check

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. Leaving the field blank will cause the health-check to occur on the port specified for the Real Server (note that in DR mode there is no Real Server port field since port re-mapping is not possible, the port specified for the Virtual Service is used).

External Script 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.

The following example illustrates how scripts can be constructed. This script uses the Linux command 'wget' to connect to the Real Server, then uses the Linux command 'grep' to look for the text 'OK' in the file 'check.txt'. The variable 'EXIT_CODE' which indicates a pass or fail is then returned to Ldirectord to control whether the server should be left online or removed.

```
#!/bin/bash
# Variables
REALIP="$3"
PORT="$2"
REQUEST="check.txt"
RESPONSE="OK"

# Get the Page/File
wget q -O -header="Host: host.domain.co.uk" http://$REALIP:$PORT/$REQUEST |grep -e $RESPONSE
if [ "$?" -eq "0" ]; then
EXIT_CODE="0"
else
EXIT_CODE="1"
fi

exit $EXIT_CODE
```

NOTE:

\$2 and \$3 are Ldirectord variables that are passed to the script. The following Ldirectord variables are available and can be used as required:

\$1 – the VIP address

\$2 – the VIP port

\$3 – the RIP address

Negotiate Check Service

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

- 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** -

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.

Database Name

The database to use for the MySQL Negotiate check. This is a required option if MySQL is selected under Negotiate Check Service above.

Radius Secret

The secret to use with Radius servers.

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 *Negotiate Check Service*. 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 SELECT query (N.B. the response expected field is not used by the SQL health check since the data returned is not read, the answer must simply be 1 or more rows). 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 check 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 *Request to Send* 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.

Additional Health Check Settings

Additional Layer 4 health check options such as Check Interval, Failure Count etc. are available using the WUI option: *Cluster Configuration > Layer 4 – Advanced Configuration*



For more details of these options, please refer to pages 59-60.

Layer 7 Services

At layer 7, Real Server health checking is handled by HAProxy. This is integrated into Loadbalancer.org appliances and allows a range of options to check that Real Servers are operational.

To configure health checks use the WUI option: *Cluster Configuration > Virtual Services > Modify*

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

Default Health Check

By default, a TCP connect health check is used for newly created layer 7 Virtual Services.

Check Port

Specify a different port for health checks. If this field is left blank, health checks occur on the port specified for each Real Server. If the VIP includes multiple ports (e.g. 80 & 443) by default the check occurs on the first port listed. If a different port must be checked, it can be specified here.

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.

Request to Send / Response Expected Example:

If the server has a virtual directory called /customers, with a default page that contains the word 'welcome' the required setup would be as follows:

Request to send: **customers**
Response expected: **welcome**

These settings would configure the following check directives in the HAProxy configuration file:

```
option httpchk GET /customers HTTP/1.0
http-check expect rstring welcome
```

(N.B. the back-slash character before 'customers' is added automatically)

Provided that the load balancer can access the page and see the text 'welcome', the health-check would pass.

Additional Health Check Settings

Additional Layer 7 health check options such as the check interval and failure count are available using the WUI option: *Cluster Configuration > Layer 7 – Advanced Configuration*



For more details of these options, please refer to page 94.

Simulating Health-Check Failures

It may not always be possible to take a server offline to check that health-checks are working correctly. In these cases, firewall rules can be used. The following rules can be configured at the console, using SSH or via the WUI under *Local Configuration > Execute a Shell Command*

To block access to a particular Real Server port:

```
iptables -A OUTPUT -p tcp --dport <Check Port> -d <REAL-SERVER-IP> -j DROP
```

e.g. `iptables -A OUTPUT -p tcp --dport 80 -d 192.168.65.60 -j DROP`

To re-enable access to a particular Real Server port:

```
iptables -D OUTPUT -p tcp --dport <Check Port> -d <REAL-SERVER-IP> -j DROP
```

e.g. `iptables -D OUTPUT -p tcp --dport 80 -d 192.168.65.60 -j DROP`

N.B. Make sure these rule are cleared after testing & verification is complete!

Fallback Server Settings

The appliance uses NGINX for the local fallback server. The fallback server is activated under the following conditions for Layer 4 & Layer 7 Virtual Services:

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 or preventing the load balancer completing its health-checks by blocking access using the firewall (see page 51).

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.

The local fallback page can be modified using the WUI option: *Maintenance > Fallback Page*

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>
```

Update

- The local fallback server is an NGINX instance that by default listens on port 9081
- If a layer 4 VIP is added that listens on port 80, NGINX is automatically configured to listen on ports 9081 & 80
- You can use any valid HTML for the default page, simply copy and paste the required HTML into the Fallback Page using the Maintenance menu

N.B. If you have a master and slave load balancer then you must change this on both servers.

Additional Fallback Server Notes:

Using the load balancers built-in Fallback Server:

- 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 be one of your internal servers.

Using an External, Dedicated Server:

- 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.
- 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.

Chapter 8 – Appliance Clustering for HA

Introduction

Appliances can be deployed as single units or as a clustered pair. We always recommend a clustered pair to avoid introducing a single point of failure.

Clustered Pair Considerations

When configured as a clustered pair, the appliances work in Active-Passive mode. In this mode the master unit handles all traffic under normal circumstances. If the master unit fails, the passive slave unit becomes active and handles all traffic.



NOTE: From v7.5, auto fail-back is not enabled by default. For a repaired master to become active and the slave unit to return to passive mode, the `hb_takeover` command must be used. This ensures control of the failback process. For more details refer to page 143.

Master / Slave Operation

Heartbeat

When a clustered pair is deployed rather than a single appliance, if the load balancers are configured using the setup wizard, heartbeat is configured to use the serial interface for heartbeat communication between master and slave. If the load balancers are configured manually (rather than using the wizard) then ucast over the network is used instead. (for the VA, ucast over the network is used in all cases). The link between the two appliances enables the state of each to be monitored by the other and permits a failover to the slave unit if the master unit should fail. It's also possible to configure both serial and ucast and as well as checks to a common node such as the default gateway. This allows a number of checks to be configured to help ensure that failover only occurs when needed and 'split brain' (i.e. master and slave both active) scenarios are avoided.

Master Slave Replication

Once the clustered pair is correctly configured, all settings related to the layer 4 and layer 7 load balanced services are automatically replicated from master to slave. This ensures that should the master unit fail, the slave is already configured to run the same services.

Settings that are NOT Replicated

Appliance specific settings that are not replicated (and therefore must be manually configured on the slave unit) are listed below:

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

Configuring Heartbeat

To configure Heartbeat:

- In the WUI, open *Cluster Configuration > Heartbeat Configuration*

Slave Load Balancer Address	<input type="text"/>	?
Communication method		
Serial	<input type="checkbox"/>	?
UDP Unicast	<input checked="" type="checkbox"/>	?
UDP Broadcast (Deprecated)	Off ▾	?
UDP Port for broadcast & unicast	<input type="text" value="6694"/>	?
Timers		
Keepalive	<input type="text" value="3"/> seconds	?
Dead time	<input type="text" value="10"/> seconds	?
Warn time	<input type="text" value="5"/> seconds	?
Ping node	<input type="text"/>	?
Automatic Fail-back	<input type="checkbox"/>	?

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

Slave Load Balancer Address

Slave Load Balancer Address – Specify the slave load balancers IP address. The slave load balancers IP address is required to enable replication of configuration data. This must be specified on the master unit only.

Communication Method

Serial communication – Enable or disable heartbeat master/slave communication over the serial port (hardware appliance only). 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.

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).

Broadcast communication (Deprecated) - Enable broadcast heartbeat master/slave communication, and choose the interface. This option is deprecated – please migrate to Unicast. 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. Please note that the broadcast option is deprecated as of v7.3. This option will be removed in a forthcoming release.

Unicast communication – 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.

UDP Port for unicast & broadcast – The UDP port number used by heartbeat for network communication over unicast or broadcast. By default, heartbeat uses port 6694/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.

Timers

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 unexpected failovers.

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.

Other Settings

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.



When configured manually (i.e. not using the setup wizard), the default heartbeat communication method for both the hardware and virtual appliance is ucast.



When the setup wizard is used to configure a hardware based appliance, Heartbeat is configured to use the serial cable. When the wizard is used to configure a virtual appliance, Heartbeat is configured to use ucast.

Adding a Slave Unit after the Master has been Configured

To add a slave unit to an existing master unit to create a highly available clustered pair, follow the steps below:

1) Slave Appliance:

- Power up the additional appliance that will be used as the Slave
- Set the appliance's network settings using one of the methods described on page 28
- Once the network is configured, connect to the WUI using a browser: **http://<IP address>:9080**
- Using the WUI option: *Local Configuration > Hostname & DNS* change the *Role* to **slave** and configure an appropriate DNS server
- Click **Update** to apply these settings

*N.B. If the hostname has not been manually changed it will automatically be set to 'lbslave' once **Update** is clicked. If the hostname has been changed, this value will remain*

2) Master Appliance:

- Using the WUI option: *Cluster Configuration > Heartbeat Configuration* enter the slave units IP address in the field *Slave Load Balancer Address*
- By default, when configured manually (i.e. not using the setup wizard) both the virtual appliance and the hardware appliance use ucast over the network for heartbeat communication. This can be changed if required using the WUI option: *Cluster Configuration > Heartbeat*
- For a hardware clustered pair, if heartbeat is configured to use serial, connect a null modem cable (one cable is supplied with each appliance) between master & slave
- Once the required method is configured click **Modify Heartbeat Configuration** to apply the heartbeat settings and replicate these settings to the slave unit
- Open the WUI option: *Maintenance > Backup & Restore* and click **Synchronize Configuration with peer** – this will copy all layer 4 and layer 7 services to the slave unit



IMPORTANT! The remaining step should be done during a maintenance window since heartbeat will be restarted which will effect all load balanced services

- Once the synchronization is complete restart heartbeat as directed using the **Restart Heartbeat** button
- Once settled verify that the master displays: **Master | Active | Link** and the slave displays: **Slave | Passive | Link**



NOTE: It's highly recommended that the clustered pair is fully tested to ensure that failover to the slave works as expected. For more details on this, please refer to page 143.

Clustered Pair Diagnostics

Heartbeat State Diagnostics

The status of the appliance is shown at the top of the screen. For a working pair, the normal view is shown below:



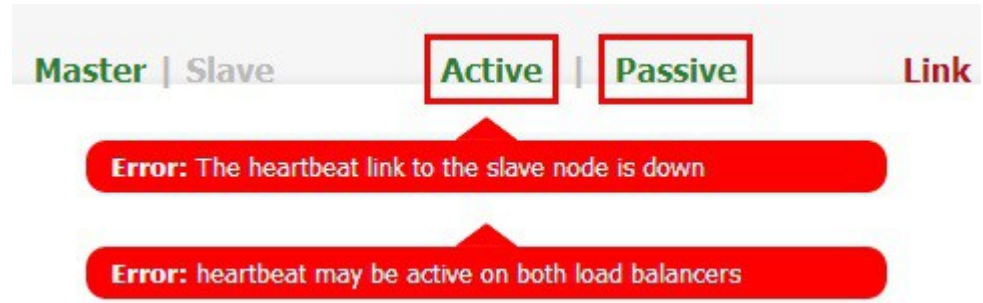
This shows that the master unit is active and that the heartbeat link is up between master & slave.

Other possible 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. Action: check & verify the heartbeat configuration
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. Action: check & verify the heartbeat configuration, check the serial cable (if applicable), check heartbeat logs & if required restart heartbeat on both units
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. Action: check & verify the heartbeat configuration, check the serial cable (if applicable), check heartbeat logs & if required restart heartbeat on both units

Split Brain Scenarios

Split brain can occur if heartbeat on the master / slave can no longer communicate with one another. In this case both units will bring up the Virtual Services and system status will show as follows:



When heartbeat communication is re-established, heartbeat will automatically attempt to resolve the split brain and ensure that only one of the units is active. If heartbeat fails to do this, the system status will show as follows on both units:



The **Take Over** button can then be used on either master or slave to attempt to force that unit to become active. If this fails, it's likely that heartbeat needs to be restarted on both units to enable heartbeat to completely re-synchronize.



NOTE: If using the **Take Over** button does not resolve the split brain condition, we recommend contacting support@loadbalancer.org. Our Engineers can then check the configuration and advise further.

Forcing Master / Slave Failover & Failback

To force the slave to become active & the master to become passive:

Run the following command on the slave:

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

To force the master to become active & the slave to become passive:

Run the following command on the master:

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

N.B. these commands can either be run on the console, at a terminal session or via the WUI using: Local Configuration > Execute Shell Command

Testing & Verifying Master / Slave Replication & Failover

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's 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.

When testing appliance fail-over, if heartbeat is configured to use the serial link don't just pull the serial cable out. This will not cause a fail-over but will cause a split brain (i.e. both units active) to occur. It's also possible to configure fail-over on network failure but this is not enabled by default. To enable this, a ping node must be configured under *Cluster Configuration > Heartbeat Configuration*.

I) Verify Basic Settings:

On the master click *System Overview* and verify that the system status appears as follows:



Master | Slave Active | Passive Link

Also check the system status in the same way on the slave unit:



Master | Slave Active | Passive Link

II) Verify Replication:

Verify that the load balanced services have 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 as on the master.

III) Verify Failover to the Slave – Using the 'hb_takeover' command:

1) on the slave using the WUI option: *Local Configuration* > *Execute a Shell Command* run the command:

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

2) now verify that the slave's status has changed to *Active* as follows:

Master | **Slave** **Active** | Passive **Link**

3) and the master has changed to *Passive*:

Master | Slave Active | **Passive** **Link**

4) 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

IV) Verify Failback to the Master – Using the 'hb_takeover' command:

1) on the master using the WUI option: *Edit Configuration* > *Execute a Shell Command* run the command:

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

2) now verify that the master's status has changed to *Active* as follows:

Master | Slave **Active** | Passive **Link**

3) and the slave has changed to *Passive*:

Master | **Slave** Active | **Passive** **Link**

4) 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

V) Verify Failover to the Slave – When powering down the Master:

- 1) power down the master using the WUI option: *Maintenance > System Control > Halt Server*
- 2) verify that the slave's status has changed to *Active* as follows:



- 3) on the slave using the WUI option: *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up

VI) Verify the Slave remains active – When powering up the Master:

- 1) power up the master
- 2) verify that the slave's status remains *Active* and the master assumes a *Passive* state

N.B. From v7.5 Cluster Configuration > Heartbeat Configuration > Automatic Fail-back is disabled by default. This ensures that the failback process can be handed in a controlled way, although this setting can be changed if preferred.

- 3) now force the master unit to become active using the WUI option: *Edit Configuration > Execute a Shell Command* to run the command on the master:

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

- 4) now verify that the master's status has changed to *Active* as follows:



- 5) and the slave has changed to *Passive*:



- 6) 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

Chapter 9 – Application Specific Settings

FTP

FTP is a multi-port service in both active and passive modes:

active 20,21

passive 21,high_port

Layer 4 Virtual Services for FTP

When configuring a Virtual Service at layer 4 for FTP, simply setup a layer 4 VIP in the normal way and set the Virtual Service / Real Server port field to port 21. Where Firewall Marks are required to handle other FTP ports, these will be configured automatically. This applies to both active and passive mode. In NAT mode, the ip_vs_ftp module is used to ensure that the client connects back via the load balancer rather than attempting to connect directly to the Real Server.

FTP Layer 4 Negotiate Health Check

You can modify the layer 4 Virtual Service so that rather than doing a simple socket connect check, it will attempt to log into the FTP server and read a file for a specific response:

Health Checks	
Check Type	Negotiate connection ?
Check Port	21 ?
External Script command	<input type="text"/> ?
Negotiate Check Options	
Negotiate Check Service	FTP ?
Virtual Host	<input type="text"/> ?
Database Name	<input type="text"/> ?
Radius Secret	<input type="text"/> ?
Login	<input type="text"/> ?
Password	<input type="text"/> ?
Request to send	check.txt ?
Response expected	OK ?

Key Points:

- Change the *check type* to Negotiate Connection
- Make sure the *Negotiate Check Service* is set to FTP
- Specify a suitable *login* and *password* for the FTP server
- Specify the file to check using the *Request to send* field (defaults to the root directory)
- The file is parsed for the *Response expected* that you specify

FTP Recommended Persistence Settings

When using multiple FTP servers in a cluster you should be aware of the effects of a client switching to a different 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 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.

Layer 7 Virtual Services for FTP

Active Mode

In active mode, the FTP server connects back to the client, so it must be aware of the clients IP address. To achieve this, TProxy must be enabled to make the load balancer transparent at layer 7. For this to work, two subnets must be used – the Virtual Server (VIP) in one subnet, the RIPv (i.e. the FTP servers) in another. For more details on TProxy, please refer to pages 117-119.

Also, to ensure that the client receives a connection from the same address that it established the control connection to, an iptables SNAT rule must be defined in the firewall script for each FTP server. The format of the required rule is as follows:

```
iptables -t nat -A POSTROUTING -p tcp -s <FTP-Server-IP> -j SNAT --to-source <FTP-VIP>
```

e.g.

```
iptables -t nat -A POSTROUTING -p tcp -s 10.20.1.1 -j SNAT --to-source 192.168.2.180
```

(one rule must be added for each FTP server in the cluster)

N.B. These rules can be added to the firewall script using the WUI option: Maintenance > Firewall Script

Active Mode – Key Points:

- Use separate subnets for the VIP & RIPv
- Enable TProxy
- Set the default gateway on the FTP servers to be an IP on the load balancer (ideally a floating IP to permit failover to the slave unit)
- Setup a layer 7 VIP listening on port 21 & configure the RIPv also to listen on port 21
- Ensure the Layer 7 Protocol is set to 'Other TCP'
- Increase the default client & server HAProxy timeouts to 5 minutes
- Add the SNAT firewall rules for each FTP server

Windows 2008 Example

- Create a L7 VIP with the following settings changing the name and IP address as required:

Label	FTP-ClusterACTV	?
Virtual Service IP address	192.168.2.180	?
Virtual Service Ports	21	?
Persistence mode	Source IP	?
Fallback Server	127.0.0.1	?
Fallback Server Port	9081	?
Proxy Protocol	<input type="checkbox"/>	?

- Define the FTP servers as RIPs for the VIP just created as illustrated below (these must be on a different subnet to the VIP to enable TProxy to work correctly):

Label	ftp1	?
Real Server IP Address	10.10.1.1	?
Real Server Port	21	?
Weight	1	?

- Enable TProxy using the WUI option: *Cluster Configuration > Layer 7 – Advanced Configuration*
- Set *Client Timeout* and *Real Server timeout* to **5m** (i.e. 5 minutes) using the WUI option: *Cluster Configuration > Layer 7 – Advanced Configuration*
- Now restart HAProxy using the WUI option: *Maintenance > Restart Services*
- Define a SNAT rule for each FTP server using the WUI option: *Maintenance > Firewall Script*

e.g.

```
iptables -t nat -A POSTROUTING -p tcp -s 10.10.1.1 -j SNAT --to-source 192.168.2.180
```

```
iptables -t nat -A POSTROUTING -p tcp -s 10.10.1.2 -j SNAT --to-source 192.168.2.180
```

- Configure each FTP servers default gateway to be the load balancer (ideally this should be a floating IP address. This can be added using the WUI option: *Cluster Configuration > Floating IPs*)
- Active FTP clients should now be able to connect to the VIP address (192.168.2.180) and view the directory listing successfully

Passive Mode

In passive mode all connections are initiated by the client. The server passes the client a port to use for the inbound data connection. By default, FTP servers can use a wide range of ports for the inbound connection and it's often useful to limit this range. The following section "Limiting Passive FTP ports" on page 152 covers this for a range of OS's & FTP servers.

Passive Mode – Key Points:

N.B. This method configures HAProxy to listen on port 21 (control channel) and all passive ports (data channel)

- It's sensible to use a controlled passive port range
- Configure the VIP to listen on port 21 and also the passive range selected, e.g. 50000-50100
- Configure the RIPv without specifying a port
- Ensure the Layer 7 Protocol is set to 'Other TCP'
- If transparency is required (for passive mode this is optional), enable TProxy using the WUI option:
Cluster Configuration > Layer 7 – Advanced Configuration

***N.B.** If TProxy is enabled, make sure that the RIPv (i.e. the FTP servers) are located in a different subnet to the Virtual Server (VIP). The default gateway on each FTP server must also be set to be an IP on the load balancer – preferably a floating IP which then allows failover to the slave unit (see page 117 for more details on using TProxy)*

- Increase the default client & sever HAProxy timeouts to 5 minutes
- To ensure the correct address is passed back to the client, on each FTP server specify the external address to be the VIP address.

e.g.

- for Windows 2008 use the **External IP address of Firewall** field

- for Linux vsftpd use the directive: `pasv_address=xxx.xxx.xxx.xxx`

- for Linux ProFTPd use the directive: `MasqueradeAddress=xxx.xxx.xxx.xxx`

Windows 2008 Example

- Create a L7 VIP with the following settings changing the name, IP address & passive port range as required:

Label	FTP-ClusterPASV	?
Virtual Service IP address	192.168.2.180	?
Virtual Service Ports	21,50000-50100	?
Persistence mode	Source IP	?
Fallback Server	127.0.0.1	?
Fallback Server Port	9081	?
Proxy Protocol	<input type="checkbox"/>	?

- Configure the VIP to listen on both the control port (21) and passive range (50000-50100) as shown
- Define the FTP servers as RIPs for the VIP just created leaving the port field blanks as illustrated below:

Label	ftp1	?
Real Server IP Address	10.10.1.1	?
Real Server Port		?
Weight	1	?

- Set *Client Timeout* and *Real Server timeout* to **5m** (i.e. 5 minutes) using the WUI option: *Cluster Configuration > Layer 7 – Advanced Configuration*
- Now restart HAProxy using the WUI option: *Maintenance > Restart Services*
- On each FTP server using IIS Manager define the same passive port range and set the external IP address to be the Virtual Server (VIP) address as shown below:

FTP Firewall Support

The settings on this page let you configure your FTP server to accept passive connections from an external firewall.

Data Channel Port Range:

50000-50100

Example: 5000-6000

External IP Address of Firewall:

192.168.2.180

Example: 10.0.0.1

N.B. The external IP address must be set to be the VIP address, this ensure that this IP address is passed back to the client to use for the subsequent inbound connection

- If TProxy is enabled, make sure the gateway of each FTP sever is set to be an IP on the load balancer (preferably a floating IP to allow failover to the slave unit)
- Now restart both IIS and the Microsoft FTP Service on each FTP server

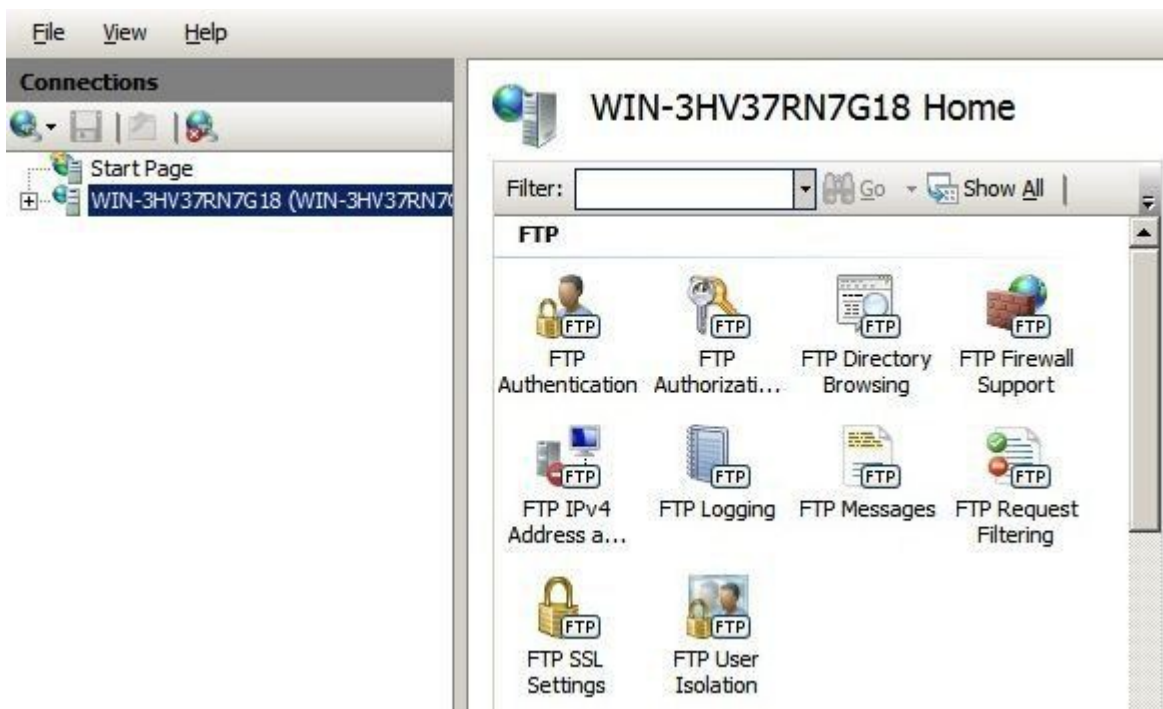
- Passive FTP clients should now be able to connect to the VIP address (192.168.2.180) and view the directory listing successfully

Limiting Passive FTP Ports

Limiting passive ports allows your firewall to be more tightly locked down. The following sections show how this is achieved for a range of Operating Systems / FTP servers.

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:

FTP Firewall Support

The settings on this page let you configure your FTP server to accept passive connections from an external firewall.

Data Channel Port Range:

Example: 5000-6000

External IP Address of Firewall:

Example: 10.0.0.1

Specify a suitable range, in the example above this is 50000-50100

IMPORTANT! - Make sure you restart IIS & the Microsoft FTP Service to apply these settings.

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 or higher must be installed for this to work)

N.B. The range that FTP will validate is from 5001 to 65535

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>
```

Terminal Services / Remote Desktop 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=7200* (2 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 2 hours, 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"/>	
Virtual Server IP address	<input type="text" value="192.168.2.165"/>	
Virtual Server Ports	<input type="text" value="3389"/>	
Persistent	<input type="text" value="Yes"/>	
Persistence Timeout	<input type="text" value="3600"/>	
Scheduler	<input type="text" value="Weighted Least Connection"/>	

Layer 7 – Microsoft Connection Broker / Session Directory

It's 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-Cluster"/>	
Virtual Service IP address	<input type="text" value="192.168.2.165"/>	
Virtual Service Ports	<input type="text" value="3389"/>	
Layer 7 Protocol	<input type="text" value="Other TCP"/>	
Persistence mode	<input type="text" value="MS Session Broker"/>	
Balance mode	<input type="text" value="Weighted Least Connections"/>	



For additional information, please refer to our RDP Deployment Guide available here:
http://www.loadbalancer.org/pdffiles/Microsoft_Terminal_Services_Deployment_Guide.pdf.

Layer 7 – RDP Cookies

The appliance also supports persistence based on RDP cookies. This method utilizes the cookie sent from the client in the initial Connection Request PDU (mstshash). 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. An associated persistence entry is also created in a stick table on the load balancer for each connection.

Label	<input type="text" value="RDP-Cluster"/>	
Virtual Service IP address	<input type="text" value="192.168.2.165"/>	
Virtual Service Ports	<input type="text" value="3389"/>	
Layer 7 Protocol	<input type="text" value="Other TCP"/>	
Persistence mode	<input type="text" value="RDP Client Cookie"/>	
Balance mode	<input type="text" value="Weighted Least Connections"/>	

Initial connections are distributed to the Real Servers based on the Balance mode selected. Client re-connects utilize the stick table to return the client to the same server first connected to. This enables clients to reconnect to their disconnected sessions.



NOTE: In certain scenarios depending on client version as well as the specific client & server settings, the RDP cookie (mstshash) is not consistently sent. For this reason RDP cookie persistence is not generally recommended.

Please also refer to our blog post on this topic: <http://blog.loadbalancer.org/microsoft-drops-support-for-mstshash-cookies/>

Other Applications

The appliance is able to support virtually any TCP or UDP based protocol which enables most applications to be load balanced. For a list of deployment guides currently available for popular applications such as Microsoft Exchange, IIS, Lync etc., please refer to page 10 earlier in this manual.



Don't hesitate to contact support@loadbalancer.org for advice on load balancing your application if it's not listed.

Chapter 10 – Configuration Examples

Introduction

This section presents three example configurations that illustrate how the appliance is configured.

Initial Network Settings

For details on configuring initial network settings and accessing the WUI please refer to page 28 and page 31.

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

This DR (Direct Return) mode example has one Virtual Service (VIP) with two Real Servers (RIPs). It's a straight forward deployment mode that can be used in many situations. It also offers the highest performance because return traffic passes directly from the Real Servers to the client rather than passing back via the load balancer.

Configuration Overview

- **Configure Network Settings** – a single Interface is needed, eth0 is normally used
- **Define the Virtual Service (VIP)** – all Real (back-end) Servers are accessed via this IP address
- **Define the Real Servers (RIPs)** – define the Real Servers that make up the cluster
- **Implement the required changes to the Real Servers** – for DR mode, the ARP issue must be solved

Network Settings

N.B. this step can be skipped if all network settings have already been configured

Configure the various network settings as outlined below:

- Using the WUI open *Local Configuration > Network Interface Configuration*

The screenshot shows the 'IP Address Assignment' section of the WUI. It features two rows for network interfaces: 'eth0' and 'eth1'. The 'eth0' row has a text input field containing '192.168.2.120/24'. The 'eth1' row has an empty text input field. Below the interface rows is a blue button labeled 'Configure Interfaces'.

- Specify the IP address & subnet 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**
- Using the WUI open *Local Configuration > Hostname & DNS*
- Specify the DNS server(s)

Domain Name Server – Primary	<input type="text" value="192.168.64.1"/>	
Domain Name Server – Secondary	<input type="text"/>	

- Click **Update**
- Using the WUI open *Local Configuration > Routing*






Default Gateway	
IP v4	<input type="text" value="192.168.2.1"/>
IP v6	<input type="text"/>

- Specify the Default Gateway
- Click **Configure Routing**

Virtual Service (VIP)

Next, configure the Virtual Service. 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 Service.

- Using the WUI open *Cluster Configuration > Layer 4 – Virtual Services* and click **[Add a new Virtual Service]**

Label	<input type="text" value="VIP1"/>	
Virtual Service IP address	<input type="text" value="192.168.2.130"/>	
Virtual Service 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"/>	

- Enter a suitable Label (name) for the VIP, e.g. **VIP1**
- Enter a valid IP address, e.g. **192.168.2.130**
- Enter a valid port, e.g. **80**
- Ensure that *Forwarding Method* is set to **Direct Routing** (*N.B. this is the default*)

Real Servers (RIPs)

Each Virtual Service requires a cluster of Real Servers (back-end servers) to forward the traffic to.

- Using the WUI open *Cluster Configuration > Layer 4 – Real Servers* and click **[Add a new Real Server]** next to the relevant Virtual Service

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, e.g. **RIP1**
- Enter a valid IP address, e.g. **192.168.2.150**
N.B. A port is not required since port redirection is not possible in DR mode. The port used will be the same as that configured for the VIP
- The weight defaults to 1 making the Real Server active immediately
- Leave *Minimum Connections* & *Maximum Connections* set to 0 which means unrestricted
- Click **Update**
- Repeat for the remaining Real Servers

Real Server Changes – Solve the ARP Problem

Since this example uses the one-arm DR mode load balancing method each Real Server requires the ARP problem to be solved:

- Each Real Server must be configured to respond to its own IP address *AND* the VIP address
- Each Real Server must be configured so that it only responds to ARP requests for its own IP address, it should NOT respond to ARP requests for the VIP address – the load balancer must respond to these requests



Failure to correctly configure the Real Servers to handle the ARP problem is the most common problem in DR configurations. Please refer to pages 62-81 for more details.

Basic Testing & Verification

Once configured, a few quick checks can be performed to verify the setup:

- Using *System Overview* check that the VIP & RIPv 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 Service
- Check *Reports > Layer 4 Current Connections* to ensure that client connections are reported in state 'ESTABLISHED'. If connections are in state 'SYN_RECEIVED', this normally indicates that the ARP problem on the Real Servers has not been correctly 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.

NOTE: 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 all Real Servers should be in the same subnet as the internal interface of the load balancer and the default gateway on each Real Server must be set to be the load balancer.

Configuration Overview





- **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 a secondary interface/alias
- **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 a secondary interface/alias
- **Configure the Master & Slave Heartbeat Settings** – set the heartbeat comms method
- **Define the Virtual Service (VIP)** – all Real Servers are accessed via this IP address
- **Define the Real Servers (RIPs)** – define the Real Servers that make up the cluster
- **Implement the required changes to the Real Servers** – in NAT mode, the Real Servers default gateway must be set to be the load balancer

Master Unit – Network Settings

- Using the WUI on the master unit, open *Local Configuration > Network Interface Configuration*

The screenshot shows the 'IP Address Assignment' section of the Network Interface Configuration WUI. It features two input fields for IP addresses and their corresponding subnets. The first field, labeled 'eth0', contains the IP address '192.168.2.120/24'. The second field, labeled 'eth1', contains the IP address '10.0.0.120/16'. Below these fields 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. **192.168.2.120/24**
- 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**
- Using the WUI open *Local Configuration > Hostname & DNS*

Hostname	<input type="text" value="lbmaster"/>	
Role	<input type="text" value="master"/>	
Domain Name Server – Primary	<input type="text" value="192.168.2.1"/>	
Domain Name Server – Secondary	<input type="text"/>	

- Ensure that *Role* is set to **master**
- Ensure that the DNS server(s) are set correctly
- Click **Update**
- Using the WUI open *Local Configuration > Routing*
- Specify the Default Gateway

Default Gateway	
IP v4	<input type="text" value="192.168.2.1"/>
IP v6	<input type="text"/>

- Click **Configure Routing**

Slave Unit – Network Settings

Configure the various network settings as outlined below:

- Using the WUI on the slave unit open *Local Configuration > Network Interface Configuration*

IP Address Assignment	
eth0	192.168.2.121/24
eth1	10.0.0.121/16

[Configure Interfaces](#)

- Specify the IP address & mask for eth0 – normally eth0 is configured as the *internal* interface although this is not mandatory, e.g. **192.168.2.121/24**
- 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**
- Using the WUI open *Local Configuration > Hostname & DNS*

Hostname	<input type="text" value="lbmaster"/>	?
Role	<input type="text" value="slave"/>	?
Domain Name Server – Primary	<input type="text" value="192.168.2.1"/>	?
Domain Name Server – Secondary	<input type="text"/>	?

[Update](#)

- Ensure that *Role* is set to **slave**
- Ensure that the DNS server(s) are set correctly
- Click **Update** – once clicked the Hostname field will automatically change to **lbslave**
- Using the WUI open *Local Configuration > Routing*

Default Gateway	
IP v4	<input type="text" value="192.168.2.1"/>
IP v6	<input type="text"/>

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

Master Unit – Heartbeat Settings

- Using the WUI on the master unit open *Cluster Configuration > Heartbeat Configuration*

Slave Load Balancer Address	<input type="text" value="192.168.2.121"/>	?
Communication method		
Serial	<input type="checkbox"/>	?
UDP Unicast	<input checked="" type="checkbox"/>	?
UDP Broadcast (<i>Deprecated</i>)	<input type="text" value="Off"/>	?
UDP Port for broadcast & unicast	<input type="text" value="6694"/>	?
Timers		
Keepalive	<input type="text" value="3"/> seconds	?
Dead time	<input type="text" value="10"/> seconds	?
Warn time	<input type="text" value="5"/> seconds	?
Ping node	<input type="text"/>	?
Automatic Fail-back	<input type="checkbox"/>	?

- Define the slave load balancers IP address in the *Slave Load Balancer Address* field, e.g. **192.168.2.121**
- Set the heartbeat communications method as required. The default when configured manually (i.e. not using the setup wizard) is UDP unicast (i.e. via the network)
- Click **Modify Heartbeat Configuration**, this will apply the heartbeat configuration on the local master and copy and apply the heartbeat configuration to the slave
- Now click **Restart Heartbeat** as prompted in the blue commit changes box – this will restart heartbeat both locally and on the slave unit to ensure that heartbeat synchronization occurs successfully

Checking the Status

A successfully configured clustered pair will display the following status:

On the Master unit:









On the Slave unit:



Virtual Service (VIP)

Next, configure the Virtual Service. This is the IP address that is presented to clients. Any packet arriving at the load balancer with that IP address / port number will be handled by the Real Servers associated with the Virtual Service.

- Using the WUI open *Cluster Configuration > Layer 4 – Virtual Services* and click **[Add a new Virtual Service]**

Label	<input type="text" value="VIP1"/>	
Virtual Service IP address	<input type="text" value="192.168.2.130"/>	
Virtual Service Ports	<input type="text" value="80"/>	
Forwarding Method	<input type="text" value="NAT"/>	
Persistent	<input type="text" value="no"/>	
Protocol	<input type="text" value="TCP"/>	

- Enter a suitable label (name) for the VIP, e.g. **VIP1**
- Enter a valid IP address, e.g. **192.168.2.130**
- Enter a valid port, e.g. **80**
- Ensure that *Forwarding Method* is set to **NAT**
- Click **Update**, this will save the VIP locally and also replicate to the slave

Real Servers (RIP)

Each Virtual Service requires a cluster of Real Servers (back-end servers) to forward the traffic to.

- Open *Cluster Configuration > Layer 4 – Real Servers* and 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, e.g. **RIP1**
- Enter a valid IP address, e.g. **192.168.2.150**
- Enter a valid port, e.g. **80**
- *Weight* defaults to 1 making the Real Server active immediately
- Leave *Minimum Connections* & *Maximum Connections* set to 0 which means unrestricted
- Click **Update**, this will save the RIP locally and also replicate to the slave
- Repeat for the remaining Real Servers

Real Server Changes – Set the Default Gateway

When using NAT mode, each Real Servers default gateway must be changed to be the load balancer. For 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 to use as the default gateway, use *Cluster Configuration > Floating IP's*.

FLOATING IPS

New Floating IP	<input type="text" value="192.168.2.254"/>
<input type="button" value="Add Floating IP"/>	

Define the IP address that you'd like to use for the default gateway, then click **Update**. Now configure the default gateway on each Real Server to use this address.

Verify the Slave Configuration

To verify that the new VIP & RIP have been replicated correctly, open the WUI on the slave and open *Cluster Configuration > Layer 4 – Virtual Services* and *Cluster Configuration > Layer 4 – Real Servers* and check that your configuration appears there also. For a correctly configured pair, the VIPs and RIPs are automatically replicated to the slave as they are defined on the master.

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 *Maintenance > Backup & Restore* and click **Synchronise Configuration with peer**. This will force the VIPs & RIPs to be copied from the master to the slave, then check again.

Basic Testing & Verification

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

- On the master, use *System Overview* to check that the VIP & RIPs 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 Service
- On the master, check *Reports > Layer 4 Current Connections* to ensure that 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 address on the load balancer.

Example 3 – One-Arm SNAT Mode & SSL Termination (Single Appliance)

This example uses HAProxy and STunnel at layer 7. STunnel is used to terminate SSL on the load balancer. STunnel then passes un-encrypted HTTP traffic to the 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.

N.B. Pound can also be used for SSL termination, although STunnel is the preferred and default method

In this example it's assumed that the Real Server application has not been designed to track & share session details between Real Servers. Therefore, cookie based persistence will be enabled on the load balancer to ensure that clients connect to the same Real Server on each subsequent connection (within the persistence timeout window). If persistence is not configured then new connections may get distributed to a different Real Server which may 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 Servers.



We generally recommend that SSL is terminated on the real servers rather than on the load balancer. This ensures that the SSL load is distributed and also ensures scalability.

Configuration Overview

- **Configure Network Settings** – A single Interface is needed, eth0 is normally used
- **Define the Virtual Service (VIP)** – All Real Servers are accessed via this IP address
- **Define the Real Servers (RIPs)** – Define the Real Servers that make up the cluster
- **Configure SSL Termination** – Configure STunnel for SSL termination

Network Settings

Configure the various network settings as outlined below:

- Using the WUI open *Local 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 one-arm configurations although this is not mandatory, e.g. **192.168.2.120/24**
- Click **Configure Interfaces**
- Using the WUI open *Local Configuration > DNS & Hostname*
- Specify the DNS server(s)

Domain Name Server – Primary	192.168.64.1	
Domain Name Server – Secondary		

Update

- Click **Update**
- Using the WUI open *Local Configuration > Routing*

Default Gateway






IP v4	192.168.2.1
IP v6	

- Specify the Default Gateway
- Click **Configure Routing**

Virtual Service (VIP)

Next, configure the Virtual Service. 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 Service.

- Using the WUI open *Cluster Configuration > Layer 7 – Virtual Services* and click **[Add a new Virtual Service]**


Label	<input type="text" value="VIP1"/>	
Virtual Service IP address	<input type="text" value="192.168.2.130"/>	
Virtual Service Ports	<input type="text" value="80"/>	
Persistence mode	<input type="text" value="HTTP Cookie"/>	
Cookie Name	<input type="text" value="SERVERID"/>	
Fallback Server	<input type="text" value="127.0.0.1"/>	
Fallback Server Port	<input type="text" value="9081"/>	

- Enter a suitable Label (name) for the VIP, e.g. **VIP1**
- Enter a valid IP address, e.g. **192.168.2.130**
- Enter a valid port, e.g. **80**
- Set *Persistence mode* to **HTTP Cookie**
- Click **Update**

Real Servers (RIP)


Each Virtual Service requires a cluster of Real Servers (back-end servers) to forward the traffic to.

- Using the WUI open *Cluster Configuration > Layer 7 – Real Servers* and 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"/>	

- Enter a suitable Label (name) for the RIP, e.g. **RIP1**















- Enter a valid IP address, e.g. **192.168.2.150**
N.B. In this mode it's 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 as soon as HAProxy is restarted
- Click **Update**
- Repeat for the remaining Real Servers
- Restart HAProxy to apply the new settings using the link provided in the blue box

 The label (name) used for the VIP is also used as the contents of the HTTP session cookie created when cookie persistence is used.

SSL Termination

An STunnel (default) or Pound VIP can be 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.

- Open *Cluster Configuration > SSL Termination* and click **[Add a new Virtual Service]**

Label	<input type="text" value="SSL1"/>	
Virtual Service IP address	<input type="text" value="192.168.2.130"/>	
Virtual Server Port	<input type="text" value="443"/>	
Backend Virtual Server IP Address	<input type="text" value="192.168.2.130"/>	
Backend Virtual Server Port	<input type="text" value="80"/>	
Ciphers to use	<input type="text"/>	
Do not insert empty fragments	<input type="checkbox"/>	
SSL Terminator	<input type="radio"/> Pound <input checked="" type="radio"/> STunnel	
Delay DNS Lookups	<input type="checkbox"/>	
Disable SSLv2 Ciphers	<input checked="" type="checkbox"/>	
Allow Client Renegotiation	<input checked="" type="checkbox"/>	
Disable Renegotiation	<input type="checkbox"/>	
Time To Close	<input type="text" value="0"/>	
Set as Transparent Proxy	<input type="checkbox"/>	

- Set *Virtual Service IP address* to be the same as the layer 7 VIP created earlier, i.e. **192.168.2.130**
- Leave *Virtual Service Port* set to **443**
- Set *Backend Virtual Service IP address* to be the same as the layer 7 VIP created earlier, i.e. **192.168.2.130**
- Leave *Backend Virtual Service Port* set to **80**
- Leave the other settings at their default values
- Click **Update**
- Restart STunnel to apply the new settings using the link provided in the blue box

When creating the SSL Virtual Service, by default a 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 please refer to page 104.

Basic Testing & Verification

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

- Using *System Overview*, verify 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 Service using HTTP
- Using a browser, navigate to the STunnel SSL VIP address, i.e. **https://192.168.2.130** to verify that you can reach the Real Servers via the Virtual Service using HTTPS
- Check / verify the certificate details

Chapter 11 – Testing Load Balanced Services

Testing Load Balanced Services

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

Use two or more clients to do the testing. Open up a web browser on each test clients and enter the URL for the VIP e.g. **http://192.168.1.20**.

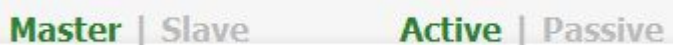
Each client should see a different server name because of the load balancing algorithm in use i.e. they are being load balanced across the cluster.

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.

Connection Error Diagnosis

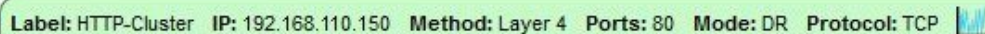
If you're unable to connect when trying to access the VIP then:

1. Make sure that the device is active. This can be checked in the WUI. For a typical deployment, the status bar should report **Master & Active** as shown below:

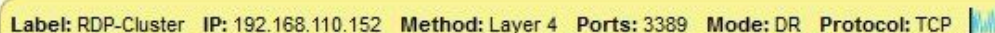


The image shows a horizontal status bar with two sections. The left section contains the text 'Master | Slave' and the right section contains 'Active | Passive'. The words 'Master' and 'Active' are highlighted in green, while 'Slave' and 'Passive' are in a lighter shade.


2. Also check *View Configuration > Network Configuration* to verify that the VIP is active on the load balancer, if not check *Logs > Heartbeat* for errors.
3. Check *System Overview* and make sure that none of your VIPs are highlighted in red. If they are, the entire cluster is down (i.e. both Real Servers). Green indicates a healthy cluster, yellow indicates that your cluster may need attention (one or more of the Real Servers may be down), and blue indicates a Real Server has been deliberately taken offline (by using either Halt or Drain).



This is a screenshot of a status bar for an HTTP-Cluster. The text reads: 'Label: HTTP-Cluster IP: 192.168.110.150 Method: Layer 4 Ports: 80 Mode: DR Protocol: TCP'. To the right of the text is a small icon of a bar chart. The entire bar has a green background.



This is a screenshot of a status bar for an RDP-Cluster. The text reads: 'Label: RDP-Cluster IP: 192.168.110.152 Method: Layer 4 Ports: 3389 Mode: DR Protocol: TCP'. To the right of the text is a small icon of a bar chart. The entire bar has a yellow background.



This is a screenshot of a status bar for an FTP-Cluster. The text reads: 'Label: FTP-Cluster IP: 192.168.110.154 Method: Layer 4 Ports: 21 Mode: DR Protocol: TCP'. To the right of the text is a small icon of a bar chart. The entire bar has a red background.

4. If the VIP is still not working, for Layer 4 VIPs check *Reports > Layer 4 Current Connections* to view the current traffic in detail. Any packets marked SYN_RECV imply incorrect Real Server configuration – if using Layer 4 DR mode see pages 62-81 on solving the ARP problem.

For Layer 4 NAT mode make sure that the default gateway on all Real Servers is set to be an IP address on the load balancer.

For Layer 7 VIPs, check *Reports > Layer 7 Status*. The default credentials required are

username: loadbalancer
password: loadbalancer

This will open a second tab in the browser and display a statistics report as shown in the example below:

Statistics Report for pid 3261

> General process information

pid = 3261 (process #1, nbproc = 1)
 uptime = 0d 0h00m42s
 system limits: memmax = unlimited; ulimit-n = 81000
 maxsock = 80024; maxconn = 40000; maxpipes = 0
 current conns = 1; current pipes = 0/0; conn rate = 2/sec
 Running tasks: 1/5; idle = 100 %

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 \(v1.6\)](#)
- [Online manual](#)

Note: UP with load-balancing disabled is reported as "NO LB".

	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	Thrtie		
Frontend				0	15	-	0	4	40 000	56		21 696	3 385 782	0	0	0	0	0	0	0	OPEN										
backup	0	0	-	0	0	-	0	0	-	0	0	0	0	0	0	0	0	0	0	0			1	-	Y						
RIP1	0	0	-	0	16	-	0	2	-	56	56	21 696	3 385 782	0	0	0	0	0	0	0	42s UP	L4OK in 0ms	1	Y	-	0	0	0	0s	-	
Backend	0	0		0	16		0	2	4 000	56		21 696	3 385 782	0	0	0	0	0	0	0	42s UP		1	1	1		0	0s			

	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	Thrtie		
Frontend				2	4	-	1	1	2 000	8		1 464	33 111	0	0	4	0	0	0	0	OPEN										
Backend	0	0		0	0		0	0	200	0	0	1 464	33 111	0	0	0	0	0	0	0	42s UP		0	0	0		0	0			

System Overview

Using *System Overview* check that when you Halt one of the Real Servers the connections are redirected to the other server in the cluster.

Remove the network cable from one of the web servers or stop the web service / process, 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 (since one has been removed from the load balancing list).

Replace the network cable, wait a few seconds and then refresh the browsers again. After a few refreshes they should again show different web servers.

The *System Overview* will also show the updated status as these tests are performed:

Label: HTTP-Cluster IP: 192.168.110.150 Method: Layer 4 Ports: 80 Mode: DR Protocol: TCP									
Rip Label	IP	Ports	Weight						
Alpha	192.168.110.237	80	1	Drain	Halt	↑			
Bravo	192.168.110.238	80	0	Online	Halt	⚙️			
Charlie	192.168.110.239	80	1	Drain	Halt	↓			

In this example:

'Alpha' is green which indicates that the Real Server is operating normally.

'Bravo' is blue, this indicates that the Real Server has been either Halted or Drained. in this example Drain has been used. If Halt was used, 'Halt' would be displayed in the Weight column rather than a weight of 0.

'Charlie' is down (red). This implies that the Real Server has failed a health check. This can be investigated using *Logs > Layer 4* or *Logs > Layer 7* as appropriate. If you know the Real Server should be active, you may need to increase the health check time-outs using *Cluster Configuration > Layer 4 – Advanced Configuration* or for Layer 7 VIPs using *Cluster Configuration > Layer 7 – Advanced Configuration*.

Using Log Files

The appliance includes several log files that are very useful when diagnosing issues. Please refer to the next chapter (chapter 11) for more details on the logs available.

Using Reports

The appliance includes several reports that are very useful when diagnosing issues. Please refer to the next chapter (chapter 11) for more details on the reports available.



For testing a clustered pair to make sure failover and failback is working correctly, please refer to Chapter 8 – Appliance Clustering for HA starting on page 136.

Chapter 12 – Appliance Monitoring

Appliance Log Files

All reports can be accessed using the *Reports* option in the WUI.

N.B. all logs are located in /var/log on the appliance

Load Balancer

File: /var/log/lbadmin.log

The lbadmin log shows all changes made to the appliances configuration. This is very useful for tracking changes made to the configuration.

Layer 4

File: /var/log/Ldirectord.log

The Ldirectord log shows the output from the health checking daemon. This is useful for checking the health your Real Servers or pinning down any configuration errors. The logging here can be quite verbose but it clearly shows exactly what the health checking process is doing.

Layer 7

File: /var/log/haproxy.log

If activated via *Cluster Configuration > Layer 7 – Advanced Configuration*, this will show the contents of the HAProxy log. This is a very detailed log of all HAProxy transactions. It's also possible to configure HAProxy to log errors only.

SSL Termination (Pound)

File: /var/log/Poundssl.log

If activated via *Cluster Configuration > SSL – Advanced Configuration*, this will show the contents of the Pound log. This is a very detailed log of all Pound SSL transactions.

SSL Termination (STunnel)

File: /var/log/STunnel.log

If activated via *Edit Configuration > SSL – Advanced Configuration*, this will show the contents of the STunnel log. The required debug level can also be set.

Heartbeat

File: /var/log/ha.log

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.

Appliance Reports

All reports can be accessed using the *Reports* option in the WUI.

Layer 4 Status

This 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.

Check Status

Virtual Service	Real Server	Forwarding Method	Weight	Active Connections	Inactive Connections
<hr/>					
<i>VIP1</i>					
192.168.110.192 port 80/tcp					
	rip1 10.0.0.100 port 80	Masq	1	17	32670
	rip2 10.0.0.101 port 80	Masq	1	18	31655

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.

Check Status

Virtual Service	Real Server	Connections / s	Incoming Packets / s	Outgoing Packets / s	Incoming Bytes / s	Outgoing Bytes / s
<hr/>						
<i>VIP1</i>						
192.168.110.192 port 80/tcp		211	2716	1636	147579	576740
	rip1 10.0.0.100 port 80	106	1361	820	73985	289216
	rip2 10.0.0.101 port 80	106	1354	816	73594	287523

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.

Virtual Service	Real Server	Connections	Incoming Packets	Outgoing Packets	Incoming Bytes	Outgoing Bytes
VIP1 192.168.111.125 port 80/tcp		2000	4996	0	255792	0
	Real_Server_1 192.168.110.240	2000	4996	0	255792	0

N.B. 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 shown above (as they do not pass through the load balancer). They will be seen for NAT mode since return traffic does pass back via the load balancer.

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. In NAT mode, this is a good indication that the Real Servers default gateway has not been configured to be the load balancer and therefore return traffic is not routed correctly.

IPVS connection entries

pro	expire	state	source	virtual	destination
TCP	00:57	SYN_RECV	192.168.64.7:3127	192.168.110.194:80	10.0.0.101:80
TCP	00:18	SYN_RECV	192.168.64.7:3106	192.168.110.194:80	10.0.0.101:80
TCP	00:57	SYN_RECV	192.168.64.7:3128	192.168.110.194:80	10.0.0.101:80
TCP	00:18	SYN_RECV	192.168.64.7:3104	192.168.110.194:80	10.0.0.101:80
TCP	00:57	SYN_RECV	192.168.64.7:3126	192.168.110.194:80	10.0.0.101:80
TCP	00:18	SYN_RECV	192.168.64.7:3105	192.168.110.194:80	10.0.0.101:80

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.

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

N.B. This password can be changed using the 'statistics password' field available under Cluster Configuration > Layer 7 – Advanced Configuration

HAProxy

Statistics Report for pid 19335

> General process information

pid = 19335 (process #1, nbproc = 1)
uptime = 0d 0h00m22s
system limits: memmax = unlimited; ulimit-n = 81000
maxsock = 80024; maxconn = 40000; maxpipes = 0
current conns = 2; current pipes = 0/0; conn rate = 2/sec
Running tasks: 2/6; idle = 100 %

■ 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
 Note: UP with load-balancing disabled is reported as "NOLE".

Display option:

- Hide 'DOWN' servers
- Refresh now
- CSV export

External resources:

- Primary site
- Updates (v1.6)
- Online manual

	Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Status	LastChk	Server							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr			Redis	Wght	Act	Bck	Chk	Dwn	Dvntme	Thrtle
Frontend							0	0	-	0	0	40 000	0	0	0	0	0	0	0	0	0	OPEN							
backup	0	0	-	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0				1	-	Y			-
rip1	0	0	-	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	22s UP	L4OK in 0ms	1	Y	-	0	0	0s	-
Backend	0	0	0	0	0	0	0	0	0	4 000	0	0	0	0	0	0	0	0	0	0	22s UP		1	1	1		0	0s	

	Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Status	LastChk	Server						
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr			Redis	Wght	Act	Bck	Chk	Dwn	Dvntme
Frontend				2	2	-	2	2	2	2 000	5	1 406	20 676	0	0	0	0	0	0	0	OPEN							
Backend	0	0	0	0	0	0	0	0	0	200	0	0	1 406	20 676	0	0	0	0	0	0	22s UP		0	0	0		0	

Layer 7 Stick Table

Displays the layer 7 stick tables. For example, if a layer 7 VIP is created using RDP cookie persistence, a stick table will be used. The related VIP is then available in the drop-down as shown below:

REPORTS > STICK TABLE (HAProxy)

HTTP-Cluster

1 Entries Returned (Max Entries Returned 1000)

ID	Key	Use	Expires	Server	Remove
0x1205f04	192.168.64.7	use=0	1790648	rip1	

NOTES:

- Stick tables are used when either source IP persistence or RDP cookie persistence is used with layer 7 Virtual Services
- Stick table entries can be removed by clicking the red 'X' in the remove column

Graphing






From v7.5 graphs are automatically configured when new Virtual and Real Servers are defined. There is no need to initialize graphs as in previous versions.

Graphs – Load Balanced Services

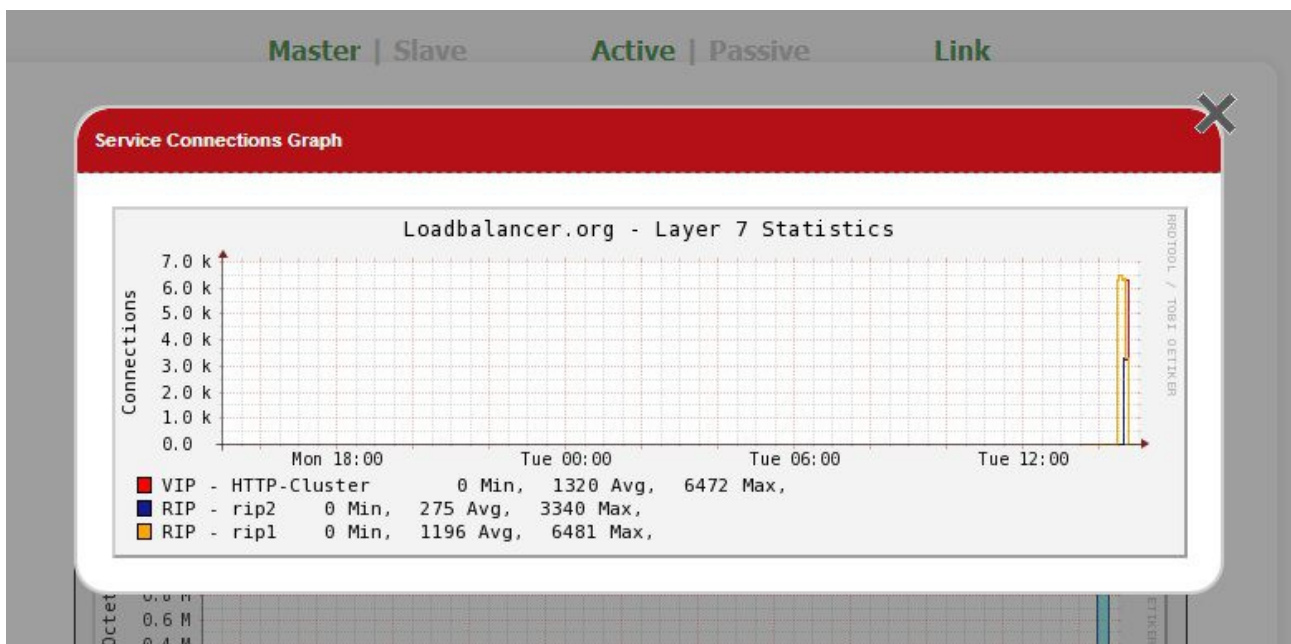
Graphs for the configured Virtual & Real Servers can be accessed either from the System Overview using the appropriate graph icon that appears next to each VIP and RIP or from the drop-down available in the WUI under *Reports > Graphing*.

Using the System Overview

The graph is displayed by clicking the relevant blue icon that's displayed next to each VIP / RIP:

Label: HTTP-Cluster IP: 192.168.111.125 Method: Layer 7 Ports: 80 Mode: Proxy Protocol: HTTP 							
Rip Label	IP	Ports	Weight				
rip1	192.168.110.237	80	1	Drain	Halt		
rip2	192.168.110.238	80	1	Drain	Halt		

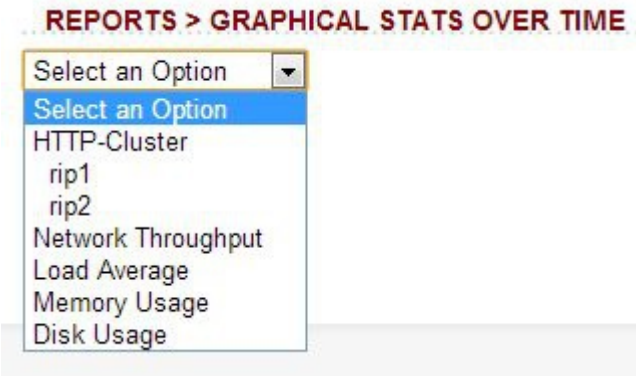
When this method is used, the daily connection graph (i.e. the last 24 hrs) is displayed for the particular VIP or RIP:



Clicking anywhere within this graph opens the complete list of graphs for the VIP / RIP in question. This is the same as selecting the VIP / RIP in the Reports menu as explained below.

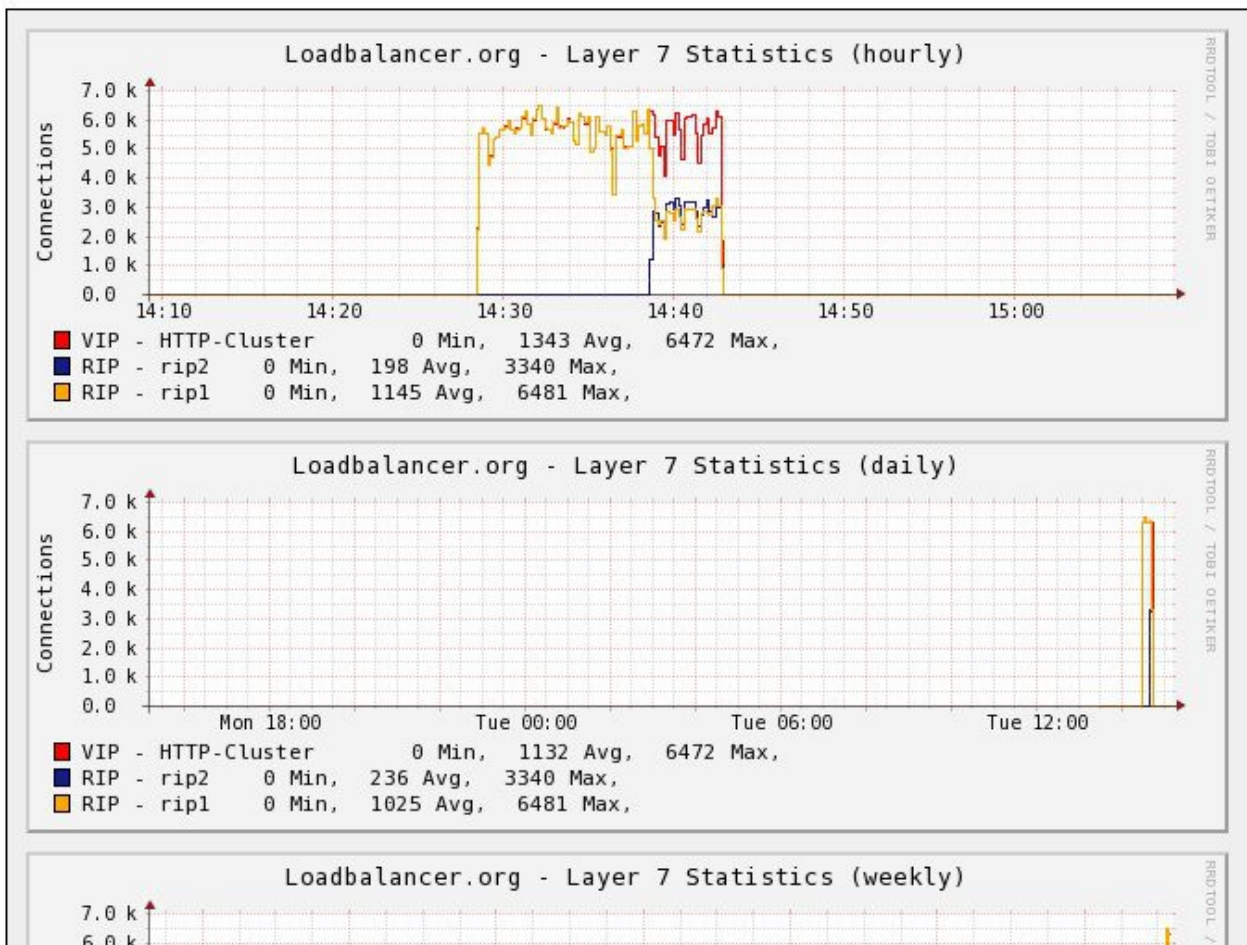
Using the Reports Menu

When selected, a drop-down similar to the following is displayed:



N.B. As VIPs & RIPs are added or removed, these are automatically added / removed from the drop-down list

When selected in this way, a complete list of graphs is displayed for the VIP / RIP selected as shown below:



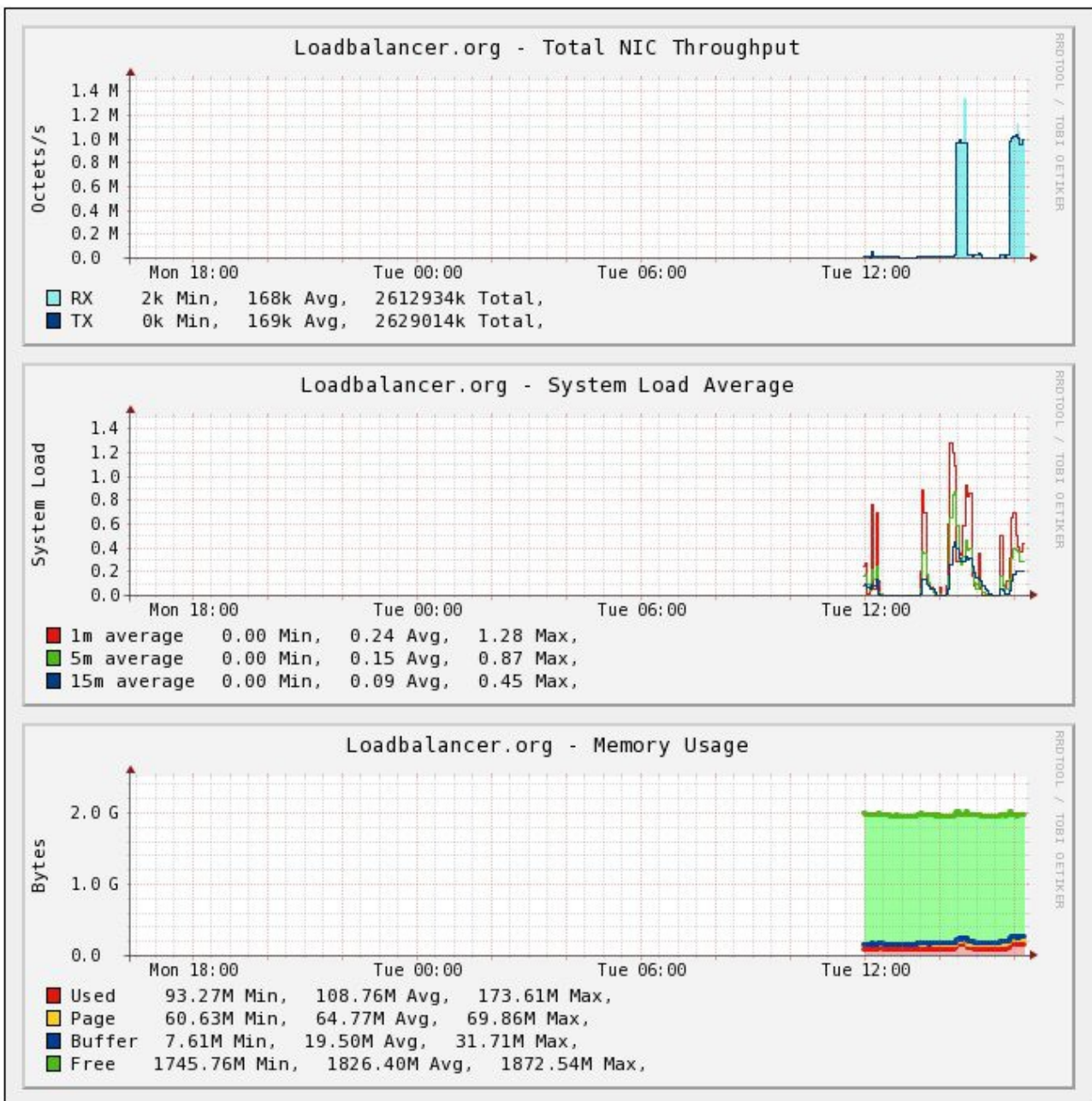
Graphs – Appliance Specific

Appliance specific graphs are available for the following statistics:

- Network Throughput
- Load Average
- Memory Usage
- Disk Usage

The first three graphs listed above are displayed in the System Overview. All four graphs can be accessed using the WUI option: *Reports > Graphing*.

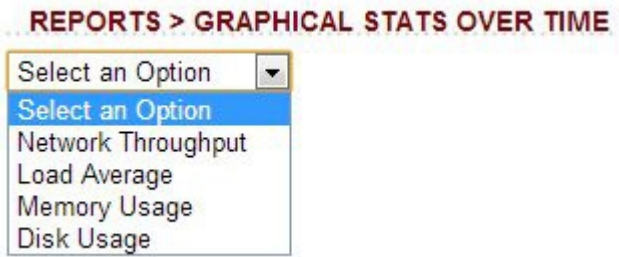
Using the System Overview



As shown above, daily graphs for Total NIC Throughput, System Load Average and Memory Usage are displayed by default in the System Overview. Clicking anyway within these graph opens the full list of related graphs. This is the same as selecting the graph in the Reports menu as explained below.

Using the Reports Menu

When selected, a drop-down including the following options is displayed:



Selecting one of these options results in a series of graphs for various periods including hourly, daily, weekly, monthly and yearly.

Graph Options

A number of graph options are available.

To change the settings:

- In the WUI, open *Local Configuration > Graphing*

The image shows a screenshot of a web interface for configuring graphing settings. The settings are organized into two sections: "Basic Configuration" and "Advanced Configuration".

Basic Configuration:

Layer 4	On	?
Layer 7	On	?
Interfaces	On	?
Load Average	On	?
Memory	On	?
Disk Usage	On	?

Advanced Configuration:

Interval	10	?
Timeout	2	?
Threads	6	?
Logging	Off	?

At the bottom of the form is an "Update" button.

- Statistics can be enabled (default) by selecting **On**
- Statistics can be disabled by selecting **Off**
- Statistics can be cleared by selecting **Delete**

Advanced Configuration Settings

Set Data Collector Interval Time – Specified in seconds. Change the interval for which data is recorded by the collector. This is a global value and will effect all collectors. Should only be changed if you have been told to do so by support.

WARNING – Changing this value will reset the RRD database files and you will loose all your previous data!!

Set Data Collector Timeout – Timeout for collector when querying the various services. Value in Seconds. Do not change unless advised to do so by support.

Set Collector Process Threads – Number of collector process threads to use for reading stats. Do not change unless advised to do so by support.

Enable Collector Logging – Enable logging for collectd. Warning this is incredibly verbose and should only be used for debugging.

SNMP Reporting

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

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

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

SNMP for Layer 4 Based Services

The root OID for Layer 4 based services is: 1.3.6.1.4.1.8225.4711

You can test if everything works by running the command shown below:

```
[root@lbmaster ~]# 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.
```

N.B. LVS-MIB.txt and other MIB files are available on the appliance in /usr/share/snmp/mibs/

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

Monitoring Layer 4 RIPs using SNMP

To list the Virtual Services use the command shown below:

```
[root@lbmaster ~]# snmpwalk -c public -v 2c -m LVS-MIB localhost 1.3.6.1.4.1.8225.4711.17.1.4
LVS-MIB::lvsServiceAddr.1 = IPAddress: 192.168.110.194
```

To list the Real Servers use the command shown below:

```
[root@lbmaster ~]# snmpwalk -c public -v 2c -m LVS-MIB localhost 1.3.6.1.4.1.8225.4711.18.1.3
LVS-MIB::lvsRealServerAddr.2.1 = IPAddress: 10.0.0.101
LVS-MIB::lvsRealServerAddr.2.2 = IPAddress: 10.0.0.100
```

This indicates that all servers are passing their health-check. If the check fails, that server will be omitted from the list as shown below:

```
[root@lbmaster ~]# snmpwalk -c public -v 2c -m LVS-MIB localhost 1.3.6.1.4.1.8225.4711.18.1.3
LVS-MIB::lvsRealServerAddr.2.1 = IPAddress: 10.0.0.100
```

In this case, 10.0.0.101 is now failing its health-check so has been omitted from the list.

SNMP for Layer 7 Based Services

The root OID for Layer 7 front-end services is: 1.3.6.1.4.1.29385.106.1.0

The root OID for Layer 7 back-end services is: 1.3.6.1.4.1.29385.106.1.1

Front end stats are returned by invoking:

```
[root@lbmaster ~]# snmpwalk -c public -v 2c localhost 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 -v 2c localhost 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.
```

Monitoring Layer 7 RIPs using SNMP

To list the Real Servers use the command shown below:

```
[root@lbmaster ~]# snmpwalk -c public -v 2c localhost 1.3.6.1.4.1.29385.106.1.2.1
SNMPv2-SMI::enterprises.29385.106.1.2.1.1.1 = STRING: "backup"
SNMPv2-SMI::enterprises.29385.106.1.2.1.1.2 = STRING: "IIS1"
SNMPv2-SMI::enterprises.29385.106.1.2.1.1.3 = STRING: "IIS2"
SNMPv2-SMI::enterprises.29385.106.1.2.1.2.1 = STRING: "backup"
SNMPv2-SMI::enterprises.29385.106.1.2.1.2.2 = STRING: "RDP1"
SNMPv2-SMI::enterprises.29385.106.1.2.1.2.3 = STRING: "RDP2"
```

To get the health status of each of these Real Servers use the command shown below:

```
[root@lbmaster ~]# snmpwalk -c public -v 2c localhost 1.3.6.1.4.1.29385.106.1.2.17
SNMPv2-SMI::enterprises.29385.106.1.2.17.1.1 = STRING: "no check"
SNMPv2-SMI::enterprises.29385.106.1.2.17.1.2 = STRING: "UP"
SNMPv2-SMI::enterprises.29385.106.1.2.17.1.3 = STRING: "DOWN"
SNMPv2-SMI::enterprises.29385.106.1.2.17.2.1 = STRING: "no check"
SNMPv2-SMI::enterprises.29385.106.1.2.17.2.2 = STRING: "DOWN"
SNMPv2-SMI::enterprises.29385.106.1.2.17.2.3 = STRING: "DOWN"
```

In this example, IIS1 is passing its health-check and IIS2, RDP1 & RDP2 are failing their health-checks.

Configuring Email Alerts











Email alerts can be configured for Layer 4 Virtual Services. This enables emails to be sent when Real Servers fail their health-checks and are removed from the table, and also when they subsequently start to pass checks and are re-added to the table. Settings can be configured globally that apply to all VIPs or individually to each VIP.

Email alerts for Layer 7 services is not currently directly supported, but it is possible to install tools such as the opensource utility Logwatch and configure this to monitor logs and send out alerts as required. For more details on using this please contact our support team: support@loadbalancer.org

Global Alerts

To configure global email alerts:

- In the WUI, open *Cluster Configuration > Layer 4 Advanced Configuration*

Layer 4		
Lock Idirectord Configuration	<input type="checkbox"/>	
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 Alert Source Address	<input type="text" value="LB1@loadbalancer.org"/>	
Email Alert Destination Address	<input type="text" value="alerts@loadbalancer.org"/>	
Auto-NAT	<input type="text" value="off"/>	
Multi-threaded	<input type="text" value="yes"/>	

- Enter an appropriate email address in the *Email Alert Source Address* field
e.g. **LB1@loadbalancer.org**
- Enter an appropriate email address in the *Email Alert Destination Address* field
e.g. **alerts@loadbalancer.org**
- Click **Update**

N.B. Make sure that you also configure an SMTP smart host using the WUI option: Local Configuration > Physical Advanced configuration > Smart Host. This will be auto-configured (if a DNS server has already been defined) to the MX record of the destination address domain name.

VIP Level Alerts

To configure VIP level email alerts:

- In the WUI, open *Cluster Configuration > Layer 4 Advanced Configuration*
- Enter an appropriate email address in the *Email Alert Source Address* field
e.g. **LB1@loadbalancer.org**
- In the WUI, open *Cluster Configuration > Layer 4 Virtual Service* and click **[Modify]** next to the VIP to be configured

Label	<input type="text" value="L4-HTTP"/>	?
Virtual Service IP address	<input type="text" value="192.168.111.125"/>	?
Virtual Service Ports	<input type="text" value="80"/>	?
Protocol	<input type="text" value="TCP"/>	?
Forwarding Method	<input type="text" value="Direct Routing"/>	?
Balance Mode	<input type="text" value="Weighted Least Connection"/>	?
Feedback Method	<input type="text" value="None"/>	?
Persistent	<input type="text" value="No"/>	?
Persistence Timeout	<input type="text" value="300"/> seconds	?
Persistence Granularity	<input type="text"/>	?
Fallback Server		
IP Address	<input type="text" value="127.0.0.1"/>	?
Port	<input type="text"/>	?
Email Alert Destination Address	<input type="text" value="alerts1@loadbalancer.org"/>	?

- Enter an appropriate email address in the *Email Alert Destination Address* field
e.g. **alerts1@loadbalancer.org**
- Click **Update**

N.B. Make sure that you also configure an SMTP smart host using the WUI option: Local Configuration > Physical Advanced configuration > Smart Host. This will be auto-configured (if a DNS server has already been defined) to the MX record of the destination address domain name.

Chapter 13 – Useful Tools & Utilities

Useful Diagnostics Tools

Full root access to the Appliance is supported which enables many useful commands to be run directly at the console or via an SSH session. Many commands can also be run using the WUI option: *Local Configuration > Execute Shell Command*. Several commonly used examples are listed below.

Netstat

Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. Useful to check that services are listening on the correct IP / port.

e.g. `netstat -anp`

Output:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:7777            0.0.0.0:*                LISTEN      17784/haproxy
tcp        0      0 127.0.0.1:7778         0.0.0.0:*                LISTEN      17784/haproxy
tcp        0      0 127.0.0.1:199         0.0.0.0:*                LISTEN      24881/snmpd
tcp        0      0 192.168.111.129:82    0.0.0.0:*                LISTEN      17784/haproxy
tcp        0      0 192.168.111.128:82    0.0.0.0:*                LISTEN      17784/haproxy
tcp        0      0 0.0.0.0:22            0.0.0.0:*                LISTEN      1789/sshd
tcp        0      0 0.0.0.0:9081          0.0.0.0:*                LISTEN      12863/nginx
tcp        0      0 192.168.110.232:443   0.0.0.0:*                LISTEN      10199/Pound
tcp        0      0 192.168.110.230:443   0.0.0.0:*                LISTEN      10113/STunnel
tcp        0      0 192.168.67.22:22      192.168.64.7:3424       ESTABLISHED 24798/sshd
tcp        0      1 192.168.67.22:49118   192.168.110.238:3389    SYN_SENT    17784/haproxy
tcp        0      1 192.168.67.22:50915   192.168.110.237:3389    SYN_SENT    17784/haproxy
tcp        0      0 :::9443                :::*                    LISTEN      1946/httpd
tcp        0      0 :::22                  :::*                    LISTEN      1789/sshd
tcp        0      0 :::9080                 :::*                    LISTEN      1946/httpd
tcp        0      0 :::ffff:192.168.67.22:9080  :::ffff:192.168.64.7:11322 ESTABLISHED 19953/httpd
etc.
```

Telnet

The telnet command is used to communicate with another host using the TELNET protocol. Useful for testing that a connection to a specific port can be made. Note that this command should be run from the console or a terminal session rather than via the WUI.

e.g. `telnet 192.168.100.10 80`

In this example, 192.168.100.10 is a Real Server, the command is useful to ensure that the load balancer is able to successfully connect to this server on port 80.

```
[root@lbmaster ~]# telnet 192.168.100.10 80
Trying 192.168.100.10...
Connected to 192.168.100.10.
Escape character is '^['.
```


Tcpdump

Tcpdump enables network traffic to be dumped to a file for analysis. Filters can also be applied if required to select which traffic is captured. Very useful tool when diagnosing network issues. Note that this command should be run from the console or a terminal session rather than via the WUI.

e.g. `tcpdump -i any -s 0 -w tcpdump-file.pcap`

This command captures all network traffic on all interfaces using the maximum packet size of 65535 bytes and dumps it to a file called `tcpdump-file.pcap`. To end the capture use CTRL+C.

Our support department may ask you to run this command and send the resulting output file to help them diagnose certain network issues.

Ethtool

Ethtool is used for querying settings of an Ethernet device and changing them.

e.g. `ethtool eth0`

Output:

```
Settings for eth0:
    Supported ports: [ TP ]
    Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full

    Supports auto-negotiation: Yes

    Advertised link modes:  10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full

    Advertised pause frame use: No
    Advertised auto-negotiation: Yes

    Speed: 100Mb/s
    Duplex: Full
    Port: Twisted Pair
    PHYAD: 1
    Transceiver: internal
    Auto-negotiation: on
    MDI-X: off
    Supports Wake-on: pumbag
    Wake-on: g

    Current message level: 0x00000001 (1)
                           drv

    Link detected: yes
```

Wireshark

Wireshark is an open source application that can be used to analyze tcpdump output files. It can be downloaded from the following location:

<http://www.wireshark.org/download.html>

Windows Specific Tools

WinSCP

WinSCP is an open source application that allows files to be uploaded/downloaded to/from the load balancer using Windows. It can be downloaded from the following location:

<http://winscp.net/eng/download.php>

PuTTY

PuTTY is an open source SSH client for Windows. It can be downloaded from the following location:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Remote Support Tools

Loadbalancer.org Support Department use **Teamviewer** for remote desktop support. The client software for various client versions is available at the following links:

Windows clients:

<http://us.loadbalancer.org/download/support/Loadbalancerv7QS-Win.exe>

Mac clients:

<http://us.loadbalancer.org/download/support/Loadbalancerv7QS-Mac.zip>

Linux clients:

<http://www.teamviewer.com/en/download/index.aspx>

(please download the latest v7 TeamViewer client available for your distribution)

Once downloaded, the client should be installed on a local machine that has access to the load balancer's WUI and also to the load balancer via SSH (Putty, WinSCP for Windows). Our Support Engineers will provide guidance as required.

Chapter 14 – Backup & Restore and Disaster Recovery

Introduction

The appliance uses various configuration files to store all settings. Files that must be backed up to enable a full restore are as follows:

XML configuration File – This is the main file for the appliance. All configuration details including local settings and load balanced services settings are stored here. This file can be backed up using the WUI.

Firewall Script – If manual changes such as manual firewall marks have been made, this file is also important. This file can be backed up using the WUI.

SSL Certificate PEM files – If SSL is terminated on the appliance, these files are also important. These files can be backed up using the WUI.

Backup & Restore

The WUI can be used for to perform backup and restore functions. To access these options:

- In the WUI, open *Maintenance > Backup & Restore*

BACKUP & RESTORE

Backup

- Download XML configuration file
- Download Firewall script
- Download SSL Certificates
- Make local XML backup
- Make local Firewall script backup

Restore

- Upload XML file & Restore: No file chosen
- Restore from the last local XML backup
- Restore Manufacturer's defaults

Synchronisation

- Synchronise configuration with peer

Backup Options

Download XML configuration file – download and save the load balancer's XML configuration file

Download Firewall script – download and save load balancer's firewall script

Download SSL Certificates – download and save the load balancer's SSL certificates

Make local XML Backup – creates a local backup of the current XML file in /etc/loadbalancer.org/userbkup

Make local Firewall Script Backup – creates a local backup of the current rc.firewall in /etc/loadbalancer.org/userbkup

Restore Options

Upload XML file & Restore – upload an XML file and restore load balancer settings

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

N.B. The XML restore feature is not backward compatible with previous major versions of the software, i.e. it's not possible to restore a V6.x XML file to a v7.x appliance.

Synchronization Options

Synchronize Configuration with peer – replicate the load balanced services configuration to the slave device.



For details of which settings are NOT replicated from master to slave when using this option, please refer to page 137.

Restoring XML Files

The screen shot below shows the process when restoring XML files. Once complete, heartbeat must be restarted to complete the restore, this will bring up the floating IP's associated with each Virtual Service.

Restoring network interfaces...

If the restored configuration removes the IP address that you are using to connect to the web interface, you will need to reconnect to the load balancer on one of its new IP addresses.

Restoring heartbeat configuration...

Restoring Layer 4 configuration...

Restoring HAProxy configuration...

Restoring Pound configuration...

Restoring STunnel configuration...

Restoring Graph configuration...

Restoring Syslog configuration...

Restarting services...

Information: Restored configuration from uploaded file.

Warning: Please note that heartbeat has been stopped to prevent interference with a running peer. When the configuration of this node is correct, heartbeat **must be restarted**.

Once complete, heartbeat must be restarted as directed to bring up the floating IP's.



For details on recovering master & slave for a clustered pair, please refer to pages 202-205.

Disaster Recovery

Being Prepared

To be able to quickly recover your appliance when a disaster occurs it's important that you create a backup of the XML file as well as other relevant configuration files and keep them stored in a secure 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

Backup the XML configuration file:

- Select *Maintenance* > *Backup & Restore* and click **Download XML configuration file**
- Select an appropriate location to store the file
- Update the filename if required then save the file

If manual firewall marks have been configured or any other manual firewall script changes have been made, backup the firewall configuration:

- Select *Maintenance* > *Backup & Restore* and click **Download Firewall Script**
- Select an appropriate location to store the file
- Update the filename if required then save the file

If you're terminating SSL on the load balancer, backup your certificates as well:

- Select *Maintenance* > *Backup & Restore* and click **Download SSL Certificates**
- Select an appropriate location to store the file
- Update the filename if required then save the file

Using wget to Copy the Files

It's also possible to use wget from remote Linux host to pull the XML configuration file and firewall script from the appliance:

```
wget --user=loadbalancer --password=loadbalancer http://<IP>:9080/lbadmin/config/getxmlconfig.php -O lb_config.xml
```

```
wget --user=loadbalancer --password=loadbalancer http://<IP>:9080/lbadmin/config/getfirewall.php -O rc.firewall
```

N.B. Replace the password 'loadbalancer' with your password if its been changed.

Backing up locally on the Load Balancer

To create local backups of the various configuration files, follow these steps:

Log in to the web interface:

Username: loadbalancer
Password: loadbalancer

- Select *Maintenance > Backup & Restore* and click **Make local XML backup**
- Select *Maintenance > Backup & Restore >* and click **Make local Firewall Script backup**

A copy of these files will be stored in `/etc/loadbalancer.org/userbkup`

Appliance Recovery using a USB Memory Stick



This will only work on 64Bit hardware. From v6.x onwards, all appliances are 64Bit. If you're running an older version, this may or may not be possible depending on the hardware.

Checking older hardware for Compatibility

If you are running v5.x and wish to determine whether your appliance is 64Bit and can be upgraded to the latest version, use the following command:

```
grep flags /proc/cpuinfo
```

This can be run from the WUI using *Local Configuration > Execute Shell command*, at the console or via a terminal session.

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) at least 4GB in size. If you are already running v6.x or later then you will already have this and should be able to simply re-image your current drive / disk module.

If you're upgrading from v5 you may need to upgrade the storage device and possibly the hardware.

Obtaining the latest disk image

The latest disk image can be downloaded from our website – please contact support@loadbalancer.org for more details.

Extracting the image from the compressed archive

Extract the image using tar under Linux or something like WinRar or 7-Zip under Windows (not the built-in Windows extractor).

Preparing the USB stick

Under Linux:

after formatting the USB stick run the command:

```
dd if=/imagefilename.img of=/dev/nameofusbdisk
```

e.g.

```
dd if=/tmp/v7.5.0_r3368.img of=/dev/sda
```

Do not use /dev/sdax where x is a number, for example – /dev/sda1 as this will install to a partition on your usb stick. Use the whole disk /dev/sda Instead.

NOTE: Be careful using this command – make sure you specify the correct disk!

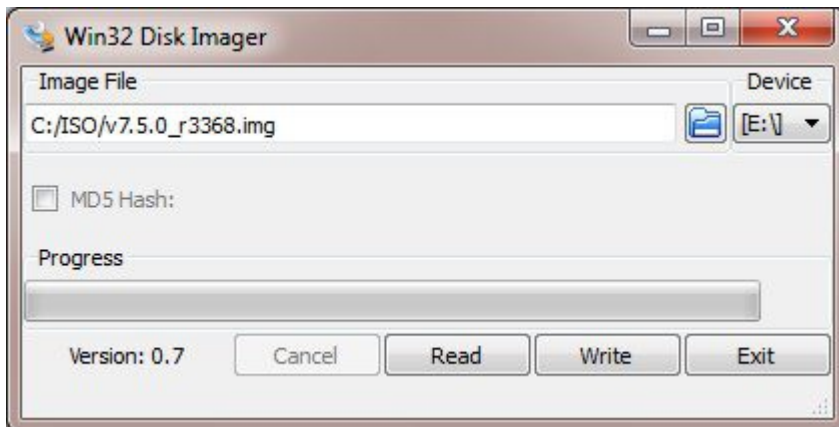
Under Windows:

For Windows, a third party image writer must be used. Several free ones are available, two examples are 'HDD Raw Copy Tool' and 'Win32 Disk Imager'. The screen shots below show Win32 Disk Imager.

Download the application from here: <http://sourceforge.net/projects/win32diskimager/>

Extract the archive

Run the executable, select the image file and set the appropriate output Device:



Click Write

NOTE: Be careful using this utility – make sure you specify the correct disk!

Using the USB Stick to restore the Appliance

- Change the appliance's BIOS settings to boot from USB first (on some models the stick must be plugged in to allow it to be selected as a boot device)
- Boot the appliance, after the initial boot messages the following prompt will appear:

```
DO YOU WISH TO CONTINUE?
```

```
Please enter yes or no
```


- Type **yes** and press <Enter>
- The installation will take around 2-3 minutes, once complete the following message will be displayed:
`Installation Finished`
- As directed, press any key to shutdown the load balancer
- Once shutdown, remove the USB stick
- Power up the appliance
- Login at the console:
Username: root
Password: loadbalancer
- Run the following command:
`lbrestore <Enter>`
- Reboot the appliance



NOTE: If your appliance is either an Enterprise, MAX or 10G please contact support@loadbalancer.org for information on completing the licensing process.

The Enterprise R16 will be licensed automatically.

***** The restore process is now complete *****

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 pages 198-199.



NOTE: If a backup copy from the master is not available, It's also possible to use the XML file 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 detailed below.

Steps (with example IP addresses) to modify a copy of the slaves XML file for use on the master:

find & Change:

```
<physical>
  <network>
    <role>slave</role>
    <hostname>lslave</hostname>
    <master>192.168.67.23</master>
    <slave></slave>
```

To:

```
<physical>
  <network>
    <role>master</role>
    <hostname>lbmaster</hostname>
    <master>192.168.67.22</master>
    <slave>192.168.67.23</slave>
```

(i.e. change the role to master)
(i.e. change the hostname to lbmaster)
(i.e. change to the masters IP address)
(i.e. add the slaves IP address)

Find & Change:

```
<rip>
  <eth0>10.0.0.23/16</eth0>
  <eth1>192.168.67.23/18</eth1>
</rip>
```

To:

```
<rip>
  <eth0>10.0.0.22/16</eth0>
  <eth1>192.168.67.22/18</eth1>
</rip>
```

(i.e. change to the masters IP address)
(i.e. change to the masters IP address)

N.B. for the MAX & 10G you may also need to change eth2 & eth3 in the same way if these interfaces are also used.

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're having with the master
- Connect the power lead, mouse, monitor and keyboard
- Power up the repaired master appliance
- If the SSD / HD failed and has been replaced, follow the steps on pages 199-201 to restore the image from USB stick
- Login to the console as:

Username: setup
Password: setup

now run through the network setup wizard to configure the initial network setting

- Open the WUI of the appliance (replace with your IP address) using: **http://192.168.2.21:9080**
- Login to the WUI as:

Username: loadbalancer
Password: loadbalancer

- In the WUI go to: *Maintenance > Backup & Restore*
- At the *Upload XML file & Restore:* option, browse to and select your backup XML file
- Click **Upload** and confirm this is what you want to do at the check prompt, the file will now be restored
- Once complete, restart heartbeat on the master as directed in the yellow box. This can be done using the WUI option: *Maintenance > Restart Services* and clicking **Restart Heartbeat**



Once heartbeat has restarted, the master and slave will be automatically re-synchronized, the master will be passive and the slave will remain active

- To force the master to go active and the slave passive, during a maintenance window run the following command on the master, this can be done via the WUI option: *Local Configuration > Execute a shell command*

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

- Verify that the master displays: **Master | Active | Link** and the slave displays: **Slave | Passive | Link**

Disaster Recovery After Slave Failure

If the slave unit has failed, the master will continue to provide load balancing services as normal. However it's 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:

OPTION 1 – Repair the unit, then restore the slave's XML backup file

or

OPTION 2 – Repair the unit, then use the 'Synchronize Configuration with peer' option on the master to re-synchronize the slave

Option 1 – Using the XML Backup

- Locate your backup copy of the slave's configuration file (lb_config.xml)
- If the failed slave is still on, power it down
- Disconnect all cables
- Repair the problems you're having with the slave
- Connect the power lead, mouse, monitor and keyboard
- Power up the repaired slave appliance
- If the SSD / HD failed and has been replaced, follow the steps on pages 199-201 to restore the image from USB stick
- Login to the console as:

Username: setup

Password: setup

now run through the network setup wizard to configure the initial network setting

- Open the WUI of the appliance (replace with your IP address) using: **http://192.168.2.21:9080**
- Login to the WUI as:

Username: loadbalancer

Password: loadbalancer

- In the WUI go to: *Maintenance > Backup & Restore*
- At the *Upload XML file & Restore:* option, browse to and select your backup XML file
- Click **Upload** and confirm this is what you want to do at the check prompt, the file will now be restored
- Once complete, restart heartbeat on the slave as directed in the yellow box. This can be done using the WUI option: *Maintenance > Restart Services* and clicking **Restart Heartbeat**



Once heartbeat has restarted, the master and slave will be automatically re-synchronized, the master will remain active and the slave will be passive

- Verify that the master displays: **Master | Active | Link** and the slave displays: **Slave | Passive | Link**

Option 2 – Synchronizing From the Master

- If the failed slave is still on, power it down
- Disconnect all cables
- Repair the problems you're having with the slave
- Connect the power lead, mouse, monitor and keyboard
- Power up the repaired slave appliance
- If the SSD / HD failed and has been replaced, follow the steps on pages 199-201 to restore the image from USB stick
- login to the console as:

Username: setup

Password: setup

now run through the network setup wizard to configure the initial network setting

Now follow the steps in the section '*Adding a Slave Unit after the master has been configured*' on page 140.

Chapter 15 – Technical Support

Introduction

Loadbalancer.org have a team of very experienced support Engineers who are available to assist with your load balancer deployment.

Unlimited support is available as follows:

- During the complimentary 90 day installation support period included with the product when no other support package is purchased
- During the cover period of any active support package
(to purchase a support package, please contact: sales@loadbalancer.org)
- During the 30 day Virtual Appliance trial period
(to download the trial please go to: www.loadbalancer.org/downloads.php)

WUI Support Options

Contact Us

This option provides details on how to contact Loadbalancer.org, how to report any issues and what information we'll need to resolve issues as quickly as we can. As mentioned here, the Loadbalancer.org support team can be contacted using the email address: support@loadbalancer.org

Sending an email to this address creates a ticket in our help desk system and enables all technical support staff to view the case. This is the most efficient way to contact support and guarantees that any reported issues will be acted upon and addressed as quickly and efficiently as possible.

For Support please email - support@loadbalancer.org

Contact Support Procedure - If your appliance is version 7.1 or later please follow the below procedure for contacting support -

- Please Compose an email to support@loadbalancer.org detailing the issue that you are seeing or the question you may have. (be specific as possible, you can never have too much detail)
- Next under the support menu click on Technical Support Download. (This will compress all of your log files and configuration files ready to be sent to us).
- Wait for the Loading icon to be replaced with a link to download the file N.B this can take up to 15 mins depending on the size of your logs and complexity of your configuration (during this time please do not refresh the page).
- Attach the downloaded file to your email and send it to support@loadbalancer.org

By Completing the above steps it will enable us to assess the situation and make recommendations for solutions as efficiently as possible.

Technical Support Download

This option enables the Support Download to be created. The download is a compressed archive containing all log files and configuration files from the appliance and should be attached to your email.

When contacting Loadbalancer.org support, you may be asked to supply the load balancer's configuration and log files. This page generates an archive of all the required files, which can then be downloaded to your PC.

- Please click the button below to generate the archive.
- The load balancer will collect the configuration files and logs into a compressed archive.
- When this is complete, you will be presented with a download link. Please save this to your PC.
- Send the archive by email to Loadbalancer.org support. If this is your first contact with support on this issue, please include your company name and details of the problem you are experiencing.

Note: Generating the archive may take several minutes on a load balancer with extensive log files. Please do not refresh the page whilst the Loadbalancer.org icon is spinning.

Generate Archive

Please click the button above to start the process.

To generate the archive, click the **Generate Archive** button.

Once complete, a link will be available to download the archive:

Generate Archive

Download support archive: master_2013-04-03_15_45_14+0100.tar.bz2

Once downloaded, attach the file to your email when contacting support, or if the file is large, it can be posted to our website – please ask our Engineers about this option.

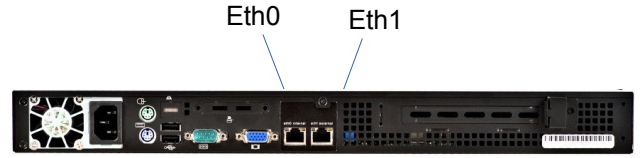
Appendix

Company Contact Information

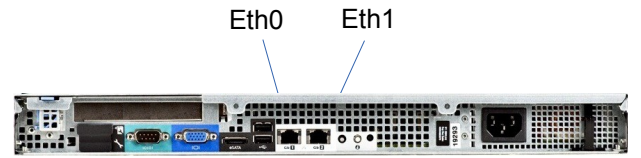
Website	URL : www.loadbalancer.org
North America (US)	Loadbalancer.org, Inc. 270 Presidential Drive Wilmington, DE 19807 USA Tel : +1 866.229.8562 (24x7) Fax : +1 302.213.0122 Email (sales) : sales@loadbalancer.org Email (support) : support@loadbalancer.org
North America (Canada)	Loadbalancer.org Ltd 300-422 Richards Street Vancouver, BC V6B 2Z4 Canada Tel : +1 604.629.7575 Fax : +1 302.213.0122 Email (sales) : sales@loadbalancer.org Email (support) : support@loadbalancer.org
Europe (UK)	Loadbalancer.org Ltd. Portsmouth Technopole Kingston Crescent Portsmouth PO2 8FA England, UK Tel : +44(0)870 4438779 (24x7) Fax : +44(0)870 4327672 Email (sales) : sales@loadbalancer.org Email (support) : support@loadbalancer.org
Europe (Germany)	Loadbalancer.org GmbH Alt Pempelfort 2 40211 Düsseldorf Germany Tel : +49 (0)30 920 383 6494 Fax : +49 (0)30 920 383 6495 Email (sales) : vertrieb@loadbalancer.org Email (support) : support@loadbalancer.org

Front & Rear Panel Layouts

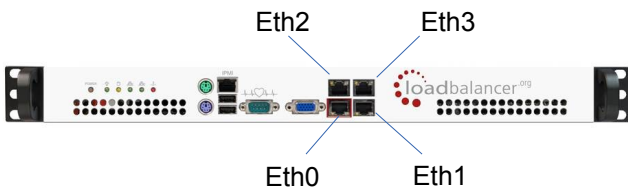
Enterprise / Enterprise R16 – Supermicro



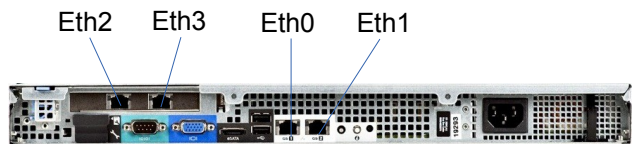
Enterprise – Dell



Enterprise Max – Supermicro



Enterprise Max / 10G – Dell



IPMI (Remote Management for Supermicro) Configuration

Supermicro based appliances include an IPMI module to allow remote control & management. This can either be accessed via the dedicated IPMI Ethernet interface or via one of the standard Ethernet interfaces in bridged mode.

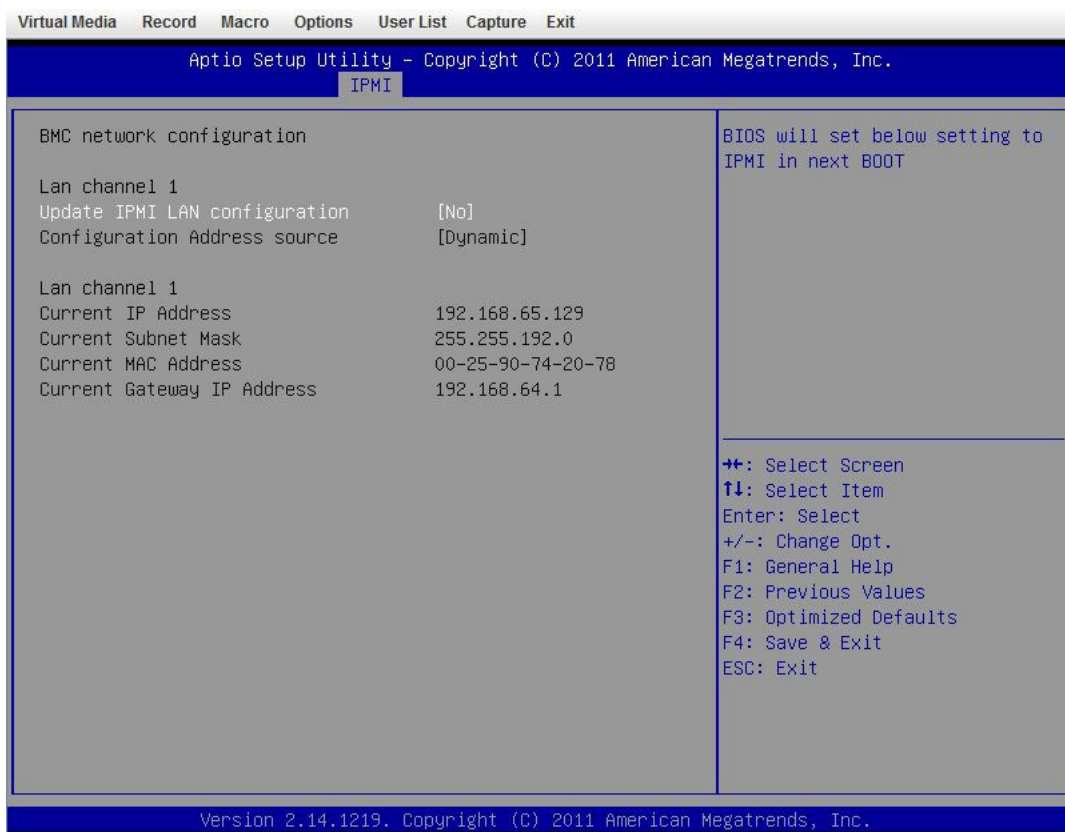
To use the dedicated IPMI interface, ensure that a network cable is plugged into the interface before powering up the appliance.

N.B. The Enterprise & Enterprise R16 do not have a dedicated IPMI interface.

Configuring the IP Address

By default the IP address is set using DHCP. The address allocated is displayed in the IPMI sub-menu in system setup. If preferred, a static IP address can also be set using the same menu. To access system setup, hit as directed at boot time.

IPMI BIOS Menu – Enterprise MAX:



To set the address

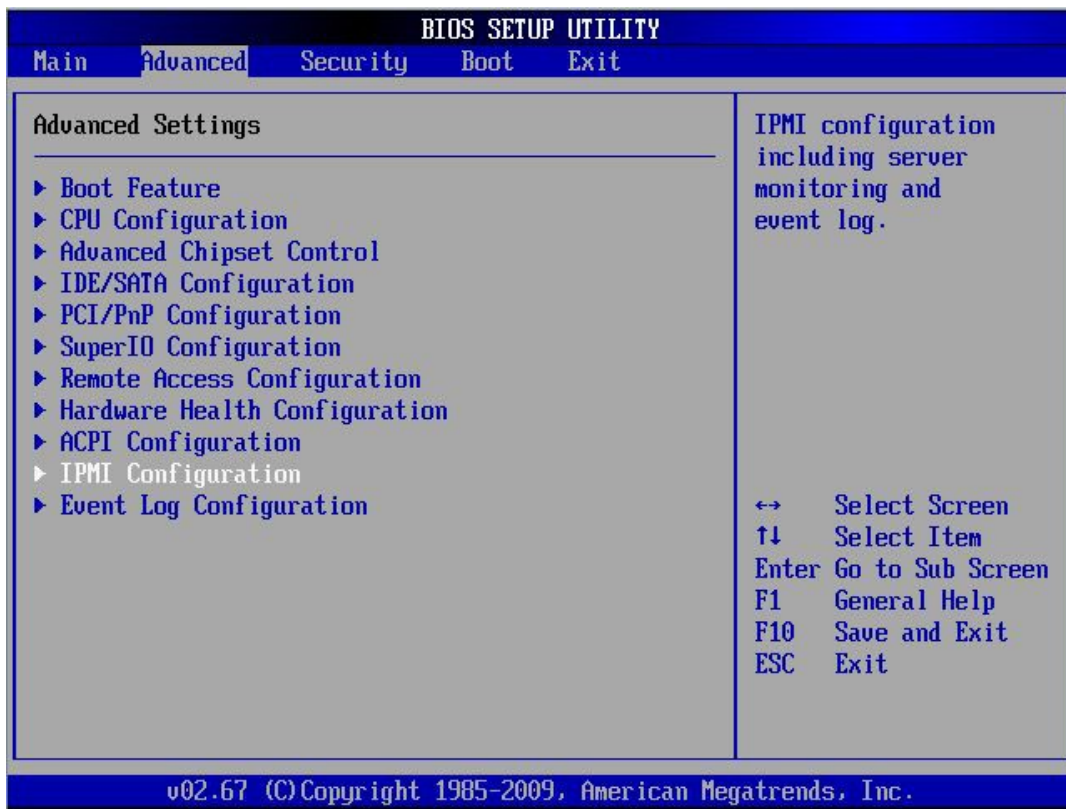
change **Update IPMI LAN configuration** to 'Yes'

change **Configuration Address Source** to 'Static'

now set the IP address, mask etc. as required.

```
Lan channel 1
Update IPMI LAN configuration      [Yes]
Configuration Address source      [Static]
Station IP address                0.0.0.0
Subnet mask                      0.0.0.0
Station MAC address              00-00-00-00-00-00
Gateway IP address               0.0.0.0
```

IPMI BIOS Menu – Enterprise & Enterprise R16:



To set the address

select **Set LAN Configuration**

change **IP Address Source** to 'Static'

now set the IP address, mask etc. as required.

```
Channel Number          [01]
Channel Number Status:Channel number is OK
IP Address Source       [Static]
IP Address               [192.168.075.111]
Subnet Mask              [255.255.192.000]
Gateway Address         [192.168.064.001]
MAC Address              [00.25.90.6F.39.DA]
```

Accessing the login page

Using a browser, connect to `http://<ip address>`

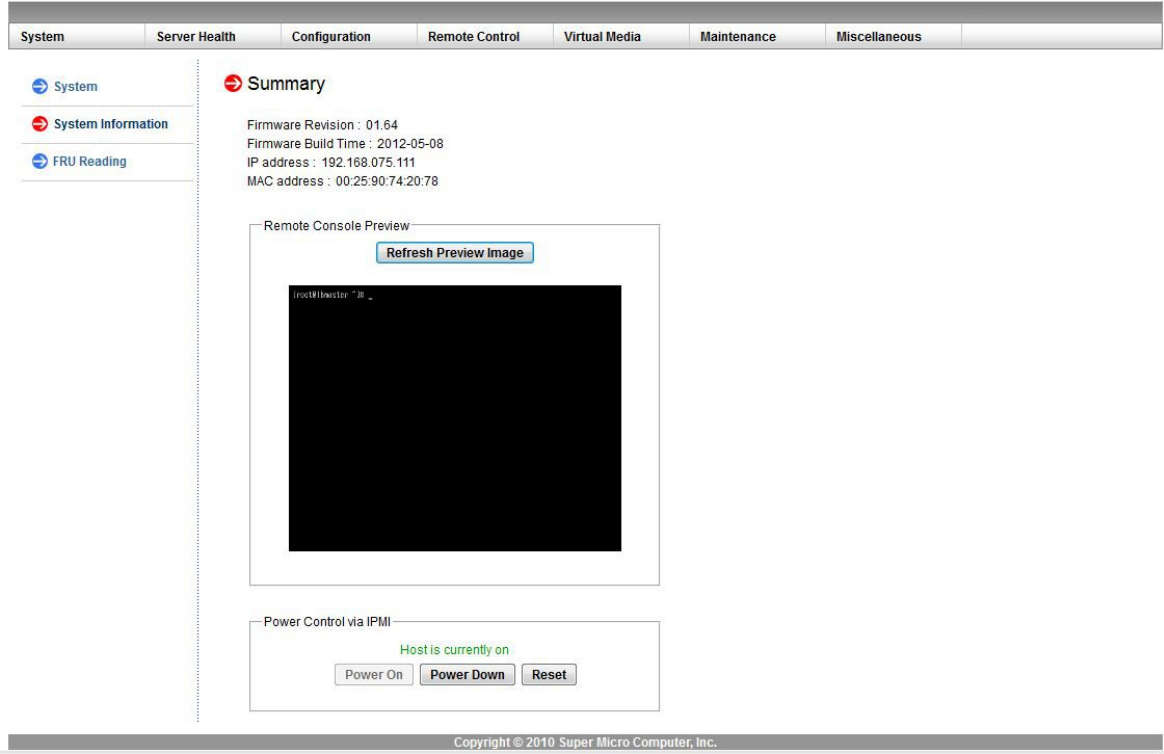
the following login prompt is displayed:

A login form with a light beige background. At the top, it says "Please Login". Below that are two input fields: "Username" and "Password". At the bottom center is a "login" button.

username: ADMIN

default password: ADMIN

Once logged in, the following screen is displayed:



IPMI Interface

As mentioned above IPMI can be accessed via the dedicated interface or via one of the standard on-board NICs. This can be configured in the IPMI interface using: *Configuration > Network > LAN Interface*

Dedicate – use the dedicated interface only

Share – run in bridge mode using one of the standard NICs

Failover – allows either connection method to be used (the default)

N.B. This option is not available on the Enterprise & Enterprise R16 since there is no dedicated IPMI port.

Remote Control

To access the systems console, simply click on the Remote Console Preview. A new window will open with access to the console of the appliance.