



Appliance Administration Manual **v8.2.5**

Rev. 1.0.0

Table of Contents

Chapter 1 – Introduction.....	11
About this Manual.....	12
About the Appliance.....	12
Latest Version.....	12
Appliance Configuration Overview.....	14
Appliance Security.....	14
Deployment Guides.....	14
Related Documentation.....	15
Additional Information.....	15
Chapter 2 – Load Balancing Concepts.....	16
Load Balancing – the Basics.....	17
Supported Protocols.....	17
Layer 4 & Layer 7.....	17
Load Balancing Algorithms.....	17
Weighted Round Robin.....	17
Weighted Least Connection.....	17
Destination Hashing.....	17
Real Server Agent.....	17
Layer 4 vs Layer 7.....	18
Other Considerations.....	18
Does Your Application Cluster correctly Handle its own State?.....	18
Replication Solutions for Shared Data.....	19
Solutions for Session Data.....	19
Persistence (aka Affinity).....	19
What do You do if Your Application is not Stateless?.....	19
Loadbalancer.org Persistence Options.....	20
Loadbalancer.org Terminology.....	20
What is a Virtual IP Address?.....	21
What is a Floating IP Address?.....	22
What are Your Objectives?.....	22
Chapter 3 – Topologies & Load Balancing Methods.....	23
One-Arm and Two-Arm Topologies.....	24
Supported Load Balancing Methods.....	25
Direct Routing (DR).....	26
Network Address Translation (NAT).....	27
NAT Mode Packet re-Writing.....	28
Source Network Address Translation (SNAT)	29
Which Load Balancing Method should I Use?.....	30
Mode Summary.....	30
Our Recommendation.....	30

Chapter 4 – Appliance Fundamentals.....	31
The Hardware Appliance – Unpacking and Connecting.....	32
The Virtual Appliance – Hypervisor Deployment.....	33
Supported Hypervisors.....	33
Host Requirements.....	33
Downloading the Appliance.....	34
VMware Deployment.....	34
Vmware Tools.....	34
Hyper-V Deployment.....	34
Linux Integration Services.....	35
KVM Deployment.....	35
XEN Deployment.....	35
Initial Network Interface Configuration.....	36
Using the Network Setup Wizard.....	36
Using Linux Commands.....	38
Appliance Access & Configuration Methods.....	38
Local Methods.....	38
Console Access.....	38
Appliance Configuration using Links.....	39
Keyboard Layout.....	39
Remote Methods.....	39
Accessing the WebUI.....	39
Configuring the Appliance using the Wizard.....	41
Configuring the Appliance Manually.....	42
Chapter 5 – Appliance Management.....	43
Network Configuration.....	44
Physical Interfaces.....	44
Configuring IP Addresses.....	44
Configuring Bonding.....	46
Configuring VLANs.....	47
Configuring MTU Settings.....	48
Configuring Default Gateway & Static Routes.....	48
Configuring Hostname & DNS Configuration.....	50
System Date & Time and NTP Server Configuration.....	50
Auto Configuration using NTP Servers.....	50
Manual Configuration.....	51
Appliance Internet Access via Proxy.....	51
SMTP Relay Configuration.....	52
Syslog Server Configuration.....	52
SNMP Configuration.....	54
Installing License Keys.....	54
Running OS Level Commands.....	55
Restoring Manufacturer's Settings.....	55

Using the WebUI.....	55
Using the Console/SSH Session.....	56
Restarting & Reloading Services.....	56
Appliance Restart & Shutdown.....	58
Appliance Software Updates.....	58
Checking the Current Software Version.....	59
Online Update.....	59
Offline Update.....	61
Updating a Clustered Pair.....	61
Firewall Configuration.....	62
Manual Firewall Configuration.....	62
Firewall Lock-down Wizard.....	63
Conntrack Table Size.....	65
Users & Passwords.....	65
Appliance Security Lockdown Script.....	67
Web User Interface HTTPS Configuration.....	69
SSH Keys.....	70
Full Root Access.....	71
Appliance Configuration Files & Locations.....	71
Chapter 6 – Configuring Load Balanced Services.....	72
Layer 4 Services.....	73
The Basics.....	73
Creating Virtual Services (VIPs).....	73
Modifying a Virtual Service.....	74
Creating Real Servers (RIPs).....	77
Persistence Considerations.....	78
Persistence State Table Replication.....	78
DR Mode Considerations.....	79
The ARP Problem.....	79
Detecting the ARP Problem.....	80
Solving the ARP Problem for Linux.....	80
Method 1 (using iptables).....	80
Method 2 (using arp_ignore sysctl values).....	80
Solving the ARP Problem for Solaris.....	81
Solving the ARP Problem for Mac OS X/BSD.....	82
Solving the ARP Problem for Windows Servers.....	83
Windows Server 2000.....	83
Windows Server 2003.....	86
Windows Server 2008.....	89
Windows Server 2012 & 2016.....	93
Verifying Strong/Weak Host Settings for Windows 2008/2012/2016.....	97
Configuring Your Application to Respond to both the RIP and VIP.....	98
Windows Firewall Settings.....	100
NAT Mode Considerations.....	102

NAT Mode Potential Issues.....	102
Enabling Real Server Internet access using Auto-NAT.....	102
Enabling Access to non Load-Balanced Services.....	102
One-Arm (Single Subnet) NAT Mode.....	103
Route Configuration for Windows Servers.....	103
Route Configuration for Linux Servers.....	103
Firewall Marks.....	104
Firewall Marks – Auto Configuration.....	104
Firewall Marks – Manual Configuration.....	105
Layer 4 – Advanced Configuration.....	110
Layer 7 Services.....	111
The Basics.....	111
Creating Virtual Services (VIPs).....	112
Modifying a Virtual Service.....	113
URL Rewriting / Content Switching (ACL's).....	117
Configuring HTTP Headers.....	120
Creating Real Servers (RIPs).....	120
Persistence Considerations.....	121
Persistence State Table Replication.....	121
Layer 7 – Custom Configurations.....	121
Configuring Manual Virtual Services.....	121
Manual Config Ex. 1 – Simple HTTP Redirect.....	122
Manual Config Ex. 2 – Load Balancing with URL matching using ACL's.....	124
HAProxy Error Codes.....	126
Layer 7 – Advanced Configuration.....	126
SSL Termination.....	130
Concepts.....	130
SSL Termination on the Real Servers – aka SSL Pass-through.....	130
SSL Termination on the Load Balancer – aka SSL Offloading.....	131
Server Name Indicator (SNI).....	131
Certificates.....	131
Generating a CSR on the Load Balancer.....	132
Uploading a Certificate.....	133
Exporting PFX Certificates from Windows Servers.....	133
Creating a PEM file.....	134
Converting between certificate formats.....	134
Creating an STunnel SSL Virtual Service.....	135
Configuring SNI (Server Name Indicator) Rules.....	138
STunnel Cipher Settings and the BEAST Attack.....	138
Creating a Pound SSL Virtual Service.....	139
Modifying a Pound SSL Virtual Service.....	141
Pound Cipher Settings and the BEAST Attack.....	141
SSL Termination on the Load Balancer with Re-encryption - aka SSL Bridging.....	142
SSL – Advanced Configuration.....	143
Pound Global Settings.....	143
STunnel Global Settings.....	144

HTTP to HTTPS Redirection.....	144
SSL Termination on the Real Servers.....	145
SSL Termination on the Load Balancer.....	146
Using Transparent Proxy (TProxy).....	147
TProxy & HAProxy.....	147
TProxy, HAProxy & Pound.....	148
TProxy, HAProxy & STunnel.....	149
Proxy Protocol.....	150
Example Configuration.....	150
Floating IPs.....	150
Server Feedback Agent.....	151
Windows Agent.....	152
Linux/Unix Agent.....	154
Custom HTTP Agent.....	154
Configuration.....	154
Configuring the Appliance via CLI, API & direct Service Calls.....	155
Command Line Interface (CLI).....	155
Application Programming Interface (API).....	157
Using ipvsadm to configure Layer 4 Services.....	161
Using Linux socket commands to configure Layer 7 Services.....	162
Chapter 7 – Web Application Firewall (WAF).....	163
Introduction.....	164
Implementation Concepts.....	165
WAF Gateway Configuration.....	166
Initial Setup.....	166
WAF Gateway Timeout.....	168
Web Gateway Authentication.....	168
WAF Gateway Operating Mode.....	169
WAF Gateway Rule Configuration.....	169
Rule White-Listing.....	169
Browsing by IP Address.....	170
Anomaly Scoring.....	170
WAF Gateway Logging & Monitoring.....	170
Modifying Default Actions.....	171
Chapter 8 – Real Server Health Monitoring & Control.....	173
Configuring Health Checks.....	174
Health Checks for Layer 4 Services.....	174
Health Checks for Layer 7 Services.....	178
Testing External Health-Check Scripts at the Command Line.....	182
Simulating Health-Check Failures.....	183
Disabling Health-Checks.....	183
Fallback Server Settings.....	183
Configuring Email Alerts.....	185

Layer 4.....	185
Global Settings.....	185
VIP Level Settings.....	186
Layer 7.....	186
Real Server Monitoring & Control using System Overview.....	187
Real Server Monitoring.....	187
Real Server Control.....	188
Ordering of VIPs.....	189
Sort by Column.....	189
Drag & Drop.....	190
Real Server Control using the HAProxy Statistics Page.....	191
Chapter 9 – Appliance Clustering for HA.....	192
Introduction.....	193
Clustered Pair Considerations.....	193
Master/Slave Operation.....	193
Heartbeat.....	193
Master Slave Replication.....	193
Settings that are NOT Replicated to the Slave Appliance.....	193
High Availability Configuration.....	194
To Create an HA Pair (Add a slave).....	194
To Break an HA Pair (Remove a slave).....	195
Promoting a Slave to Master.....	197
Configuring Heartbeat.....	197
Clustered Pair Diagnostics.....	200
Heartbeat State Diagnostics.....	200
Split Brain Scenarios.....	201
Forcing Master/Slave Failover & Failback.....	201
Testing & Verifying Master/Slave Replication & Failover.....	202
Chapter 10 – Application Specific Settings.....	205
FTP.....	206
Layer 4 Virtual Services for FTP.....	206
FTP Layer 4 Negotiate Health Check.....	206
FTP Recommended Persistence Settings.....	207
Layer 7 Virtual Services for FTP.....	207
Active Mode.....	207
Windows 2008 Example.....	208
Passive Mode.....	209
Windows 2008 Example.....	210
Limiting Passive FTP Ports.....	211
For Windows 2008.....	211
For Windows 2003.....	212
For Windows 2000 (SP4 and later).....	212
For Linux.....	212
Terminal Services/Remote Desktop Services.....	213

Layer 4 – IP Persistence.....	213
Layer 7 – Microsoft Connection Broker/Session Directory.....	213
Layer 7 – RDP Cookies.....	213
Other Applications.....	214
Chapter 11 – Configuration Examples.....	215
Introduction.....	216
Initial Network Settings.....	216
1 – One-Arm DR Mode (Single Appliance).....	216
Configuration Overview.....	216
Network Settings.....	216
Virtual Service (VIP).....	217
Real Servers (RIPs).....	218
Physical Real Server Changes – Solve the ARP Problem.....	218
Basic Testing & Verification.....	219
2 – Two-Arm NAT Mode (Clustered Pair).....	219
Configuration Overview.....	219
Master Unit – Network Settings.....	219
Slave Unit – Network Settings.....	220
Creating the HA Clustered Pair.....	221
Checking the Status.....	222
Virtual Service (VIP).....	223
Real Servers (RIPs).....	223
Physical Real Server Changes – Set the Default Gateway.....	224
Verify the Slave Configuration.....	224
Basic Testing & Verification.....	224
3 – One-Arm SNAT Mode & SSL Termination (Single Appliance).....	224
Configuration Overview.....	225
Network Settings.....	225
Virtual Service (VIP).....	226
Real Servers (RIPs).....	227
SSL Termination.....	228
Basic Testing & Verification.....	229
Chapter 12 – Testing Load Balanced Services.....	230
Testing Load Balanced Services.....	231
Diagnosing VIP Connection Problems.....	231
Taking Real Servers Offline.....	232
Using Log Files.....	233
Using Reports.....	233
Chapter 13 – Appliance Monitoring.....	234
Appliance Log Files.....	235
Load Balancer.....	235
Layer 4.....	235
Layer 7.....	235
SSL Termination (Pound).....	235

SSL Termination (STunnel).....	235
Heartbeat.....	235
Apache Error Log.....	235
Apache User Log.....	235
WAF Logs.....	236
Appliance Reports.....	236
Layer 4 Status.....	236
Layer 4 Traffic Rate.....	237
Layer 4 traffic Counters.....	238
Layer 4 Current Connections.....	238
Layer 4 Current Connections (resolve hostnames).....	239
Layer 7 Status.....	239
Layer 7 Stick Table.....	240
Graphing.....	240
Graphs – Load Balanced Services.....	240
Graphs – Appliance Specific.....	243
Graph Options.....	244
SNMP Reporting.....	245
SNMP for Layer 4 Based Services.....	245
Monitoring Layer 4 VIPs & RIPs using SNMP.....	246
SNMP for Layer 7 Based Services.....	246
Monitoring Layer 7 RIPs using SNMP.....	247
Chapter 14 – Useful Tools & Utilities.....	248
Useful Diagnostics Tools.....	249
Netstat.....	249
Telnet.....	249
Tcpdump.....	249
Ethtool.....	250
Wireshark.....	250
Windows Specific Tools.....	250
Microsoft Network Monitor.....	250
WinSCP.....	251
PuTTY.....	251
Remote Support Tools.....	251
Chapter 15 – Backup & Restore and Disaster Recovery.....	252
Introduction.....	253
Backup & Restore.....	253
Restoring XML Files.....	254
Disaster Recovery.....	255
Being Prepared.....	255
Backing Up Configuration Files to a Remote Location.....	255
Using wget to Copy the Files.....	256
Backing up locally on the Load Balancer.....	256
Appliance Recovery using a USB Memory Stick.....	256

Disaster Recovery After Node (Master or Slave) Failure.....	259
Recovery Steps.....	259
Chapter 16 – Technical Support.....	261
Introduction.....	262
WebUI Support Options.....	262
Contact Us.....	262
Technical Support Download.....	263
Useful Links.....	264
Appendix.....	265
Front & Rear Panel Layouts.....	266
IPMI (Remote Management) Configuration for the Enterprise R20 & MAX.....	267
iDRAC (Remote Management) Configuration for the Enterprise 10G & Ultra.....	271
Appliance IPv4 Address Format (CIDR notation).....	272
Company Contact Information.....	273

Chapter 1 – Introduction

About this Manual

This document covers all required administration information for v8.2 Loadbalancer.org appliances.

About the Appliance

The Loadbalancer.org appliance runs the GNU/Linux operating system with a custom kernel configured for load balancing.

The core software is based on customized versions of Centos 6.x/RHEL 6.x, Linux 3.10.x, LVS, HA-Linux, HAProxy, Pound, STunnel & Ldirectord. Full root access is provided which enables complete control of all settings.

The appliance is available in the following formats: hardware, virtual (VMware, HyperV, KVM & XEN) and cloud based (Amazon & Azure).

Appliances can be deployed as single units or as a clustered pair.

Note:

Loadbalancer.org always recommend that clustered pairs should be used where possible for high availability and resilience, this avoids introducing a single point of failure to your network. For more information on configuring an HA pair please refer to page [192](#).

LATEST VERSION

The latest version of the appliance (v8.2.5) includes the following new features, updates and bug fixes:

Haproxy

- IPV6 RIPS fail when the firewall lockdown wizard is updated and enabled
- When mode selected the incorrect source address was being sent
- Advanced configuration settings were not getting saved
- HTTP Request timeout added

WAF

- Waf is not completely removed from the slave when deleted from the master
- Waf is not running on slave when HA failover occurs after a reboot
- Waf now using apache 2.4
- Google authentication is now available
- Improved performance
- Double Login for important information

Ldirectord

- Ldirectord hangs on deleted real server if SMTP server causes the email send to stall

- Deleting VIP on the master when the slave is down results in an exception
- Health checks can loop and cause flapping when used with Feedback agent

Stunnel

- Updated Stunnel to v5.41
- Problem inputting PFX certificate with password containing dollar sign
- Fixed typo in Stunnel cipher list
- Stunnel initscript now allows seamless restarts

Network

- Could only configure R20 appliances with 2 NICs

Other

- Improved options for Rsyslog
- IP Routing rules priority for TProxy

RPMs

- keyutils-libs-devel-1.4-5.el6.x86_64.rpm
- krb5-devel-1.10.3-65.el6.x86_64.rpm
- libcom_err-devel-1.41.12-23.el6.x86_64.rpm
- libkadm5-1.10.3-65.el6.x86_64.rpm
- libselinux-devel-2.0.94-7.el6.x86_64.rpm
- libsepol-devel-2.0.41-4.el6.x86_64.rpm
- zlib-devel-1.2.3-29.el6.x86_64.rpm
- e2fsprogs-1.41.12-23.el6.x86_64.rpm
- e2fsprogs-libs-1.41.12-23.el6.x86_64.rpm
- krb5-libs-1.10.3-65.el6.x86_64.rpm
- libcom_err-1.41.12-23.el6.x86_64.rpm
- libss-1.41.12-23.el6.x86_64.rpm
- cjpeg-0.4.1-1.el6.x86_64.rpm
- jansson-2.9-1.el6.x86_64.rpm
- jansson-devel-2.9-1.el6.x86_64.rpm
- openssl-devel-1.0.1e-57.el6.x86_64.rpm
- openssl-1.0.1e-57.el6.x86_64.rpm
- stress-1.0.4-4.el6.x86_64.rpm

Note:

For details of previous releases, please refer to our [support website](#).

Appliance Configuration Overview

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

Once the network is configured, the appliance can be configured manually or by using the Setup Wizard (for layer 7 services). The WebUI is accessible using HTTP on port **9080** and HTTPS on port **9443**. If you prefer that the WebUI is only accessible via HTTPS, please refer to page [69](#) for more information. It's also possible to configure the load balancer at the console using the text based Links browser, although using the WebUI is the recommended method.

For a clustered pair, we recommend that the master is fully configured first, then the slave should be added. For more information on configuring a clustered pair, please refer to page [192](#). Once a pair is configured, load balanced services must be configured & modified on the master appliance. The slave appliance will then be kept in sync automatically.

Appliance Security

The appliance includes a security lockdown command (**lbsecure**) that enables passwords to be set, network access to be locked down and SSH key regeneration in one simple step. This command can be run on a single appliance or an HA pair. For more details please refer to page [67](#).

As mentioned above, by default, the Web User Interface is accessible via HTTP and HTTPS and can be locked down to HTTPS only if preferred. Please refer to page [69](#) for more information.

Deployment Guides

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

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 Remote Desktop Services](#)
- [Load Balancing Microsoft Exchange 2016](#)
- [Load Balancing Microsoft Exchange 2013](#)
- [Load Balancing Microsoft Exchange 2010](#)
- [Load Balancing Microsoft Sharepoint](#)
- [Load Balancing Microsoft AD FS](#)
- [Load Balancing Microsoft DirectAccess](#)
- [Load Balancing VMware View](#)
- [Load Balancing Microsoft Skype For Business](#)
- [Load Balancing Microsoft Lync 2010/2013](#)
- [Load Balancing Microsoft OCS 2007 R2](#)
- [Load Balancing Web Proxies/Filters/Gateways \(generic guide\)](#)

- [Load Balancing Bloxx Web Filter](#)
- [Load Balancing McAfee Web Gateway](#)
- [Load Balancing Smoothwall Web Gateway](#)
- [Load Balancing Barracuda Web Filter](#)
- [Load Balancing Clearswift Web Gateway](#)
- [Load Balancing Trend Micro Web Gateway](#)
- [Load Balancing Sophos Web Gateway](#)
- [Load Balancing RSA Authentication Manager](#)
- [Load Balancing Oracle Application Server](#)
- [Load balancing Medical Imaging & Information Systems Protocols](#)
- [Load Balancing SIP](#)

AWS Specific Quick Guides:

- [Load Balancing Asterisk in AWS](#)
- [Load Balancing Web Servers and Deploying a WAF in AWS](#)
- [Load Balancing Microsoft Session Host in AWS](#)

Related Documentation

The following related documentation may also be useful:

- [Appliance Quick Start Guide \(Hardware\)](#)
- [Appliance Quick Start Guide \(Virtual\)](#)
- [Amazon AWS appliance \(Enterprise AWS\) Quick Start Guide](#)
- [Microsoft Azure appliance \(Enterprise Azure\) Quick Start Guide](#)

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 if you have any questions, please contact our support team: 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. etc.

LAYER 4 & LAYER 7

Load balancing at layer 4 and layer 7 is supported. LVS (*Linux Virtual Server*) 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 assigns jobs to servers through looking up a statically assigned hash table by their destination IP addresses (available for Layer 4 Virtual Services only).

REAL SERVER AGENT

To compliment the methods above, Loadbalancer.org appliances also support Real Server (i.e backend 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 or RAM. Without the agent, the load balancer has 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 [151](#).

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). 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 page [79](#) 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 in any way.

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 [147](#) and [116](#) respectively).

Other Considerations

DOES YOUR APPLICATION CLUSTER CORRECTLY HANDLE ITS OWN STATE?

Note:

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 backend 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, 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 backend 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

The following default persistence options are available:

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

Note:

It's also possible to define other custom persistence types if required using the manual configuration option available for layer 7 Virtual Services.

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 older versions of Internet Explorer, the session ID can be renegotiated frequently (every few seconds) which effectively breaks the persistence.

Loadbalancer.org Terminology

Load Balancer	An IP based traffic manager for server clusters
VIP	Virtual IP address – the address of the load balanced cluster of RIPs, the address presented to connecting clients
Floating IP	The Floating IP Address is automatically created whenever a VIP is configured, the FIP address is the same as the VIP address, it enables services to be moved between the master and slave appliance
RIP	The Real IP address of a backend server in the cluster
GW	The Default Gateway for a backend server in the cluster
WebUI / WUI	Web User Interface
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 Mode <i>(Layer 4)</i>	Direct Routing (aka DSR/Direct Server Return) is a standard load balancing technique that distributes packets by altering only the destination MAC address of the packet
NAT Mode <i>(Layer 4)</i>	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 Mode <i>(Layer 7, HAProxy)</i>	Source Network Address Translation – the 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 interfaces connected to two subnets - this can be achieved using two physical network cards or by assigning two addresses to one physical network card
Eth0	Usually the internal Ethernet interface, although this is optional. Also known as Gb0 on the Enterprise 10G and Enterprise Ultra
Eth1	Usually the external Ethernet interface, although this is optional. Also known as Gb1 on the Enterprise 10G and Enterprise Ultra
Eth2	Third Ethernet interface
Eth3	Fourth Ethernet interface
Eth4	Fifth Ethernet interface (Enterprise Ultra only)
Eth5	Sixth Ethernet interface (Enterprise Ultra only)

WHAT IS A VIRTUAL IP ADDRESS?

Most load balancer vendors use the term Virtual IP address (VIP) to describe the address that the cluster is accessed from. It's important to understand that the Virtual IP address (VIP) refers to both the physical IP

address and also to the logical load balancer configuration. Likewise the real IP (RIP) address refers to both the Real Servers physical IP address and its representation in the logical load balancer configuration.

Note:

It's not possible to configure a VIP on the same IP address as any of the network interfaces. This ensures services can 'float' (move) between master and slave appliances.

WHAT IS A FLOATING IP ADDRESS?

A floating IP address (FIP) is automatically created whenever a VIP is configured. The FIP address is the same as the VIP address. Since the FIP must be able to move between the master and slave appliance, it's not possible to configure a VIP/FIP on the same IP address as an interface as mentioned in the note above. This restriction also applies to single appliance (i.e. master only) deployments. FIPs can also be manually defined to provide a 'floating default gateway' for layer 4 NAT mode configurations. This allows the default gateway for the NAT mode Real Servers to be brought up on the slave should the master fail.

What are Your Objectives?

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

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

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

Performance	A load balancer can increase performance by allowing you to utilize several commodity servers to handle the workload of one application
Reliability	Running an application on one server gives you a single point of failure. Utilizing a load balancer moves the point of failure to the load balancer. At Loadbalancer.org we always advise that you deploy load balancers as clustered pairs to remove this single point of failure (for more details on configuring a clustered pair please refer to page 192)
Maintenance	Using the appliance, you can easily bring servers on and off line to perform maintenance tasks, without disrupting your users

Note:

In order to achieve all three objectives of performance, reliability & maintenance in a web based application, your application must handle persistence correctly (see page [18](#) for more details).

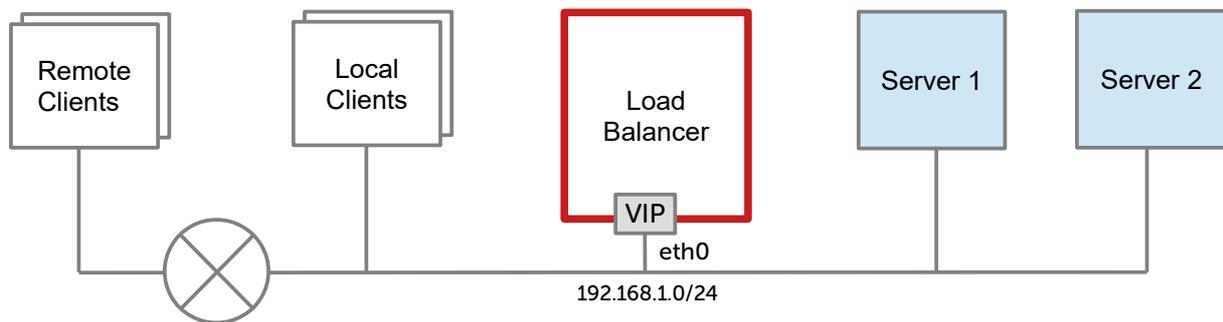
Chapter 3 – Topologies & Load Balancing Methods

One-Arm and Two-Arm Topologies

The number of 'arms' is a descriptive term for how many interfaces are used to connect a device to a network. It's 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.

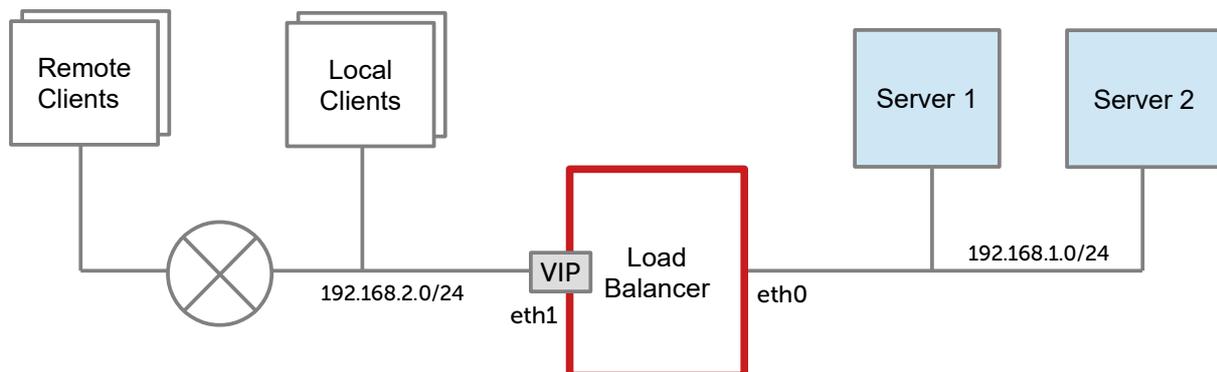
One Arm

In this mode, the VIP and the load balanced servers are located in a single subnet. The load balancer requires a single network adapter – eth0 in the diagram below.



Two Arm

In this mode, 2 subnets are used. The VIP is located in one subnet and the load balanced servers are located in the other subnet. The load balancer requires 2 interfaces – eth0 and eth1 in the diagram below. Note that this can be achieved by using two network adapters, or by creating VLANs on a single adapter.



Supported Load Balancing Methods

The Loadbalancer.org appliance is one of the most flexible load balancers on the market. The design 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 - Requires solving the 'ARP problem' on the Real Servers - please refer to page 79 for more details	One-Arm
Layer 4	NAT (Network Address Translation)	Fast Layer 4 load balancing - The appliance must be the default gateway for the Real Servers	One or Two-Arm
Layer 4	TUN	Similar to DR but works across IP encapsulated tunnels	One-Arm
Layer 7	SSL Termination (Pound & STunnel)	Usually required in order to process cookie persistence in HTTPS streams on the load balancer - SSL Termination is processor intensive	One or Two-Arm
Layer 7	SNAT (Source Network Address Translation using HAProxy)	Layer 7 allows great flexibility including full SNAT and remote server load balancing, cookie insertion and URL switching - Very simple to implement - Requires no Real Server changes - Not as fast as Layer 4 methods	One or Two-Arm

Key

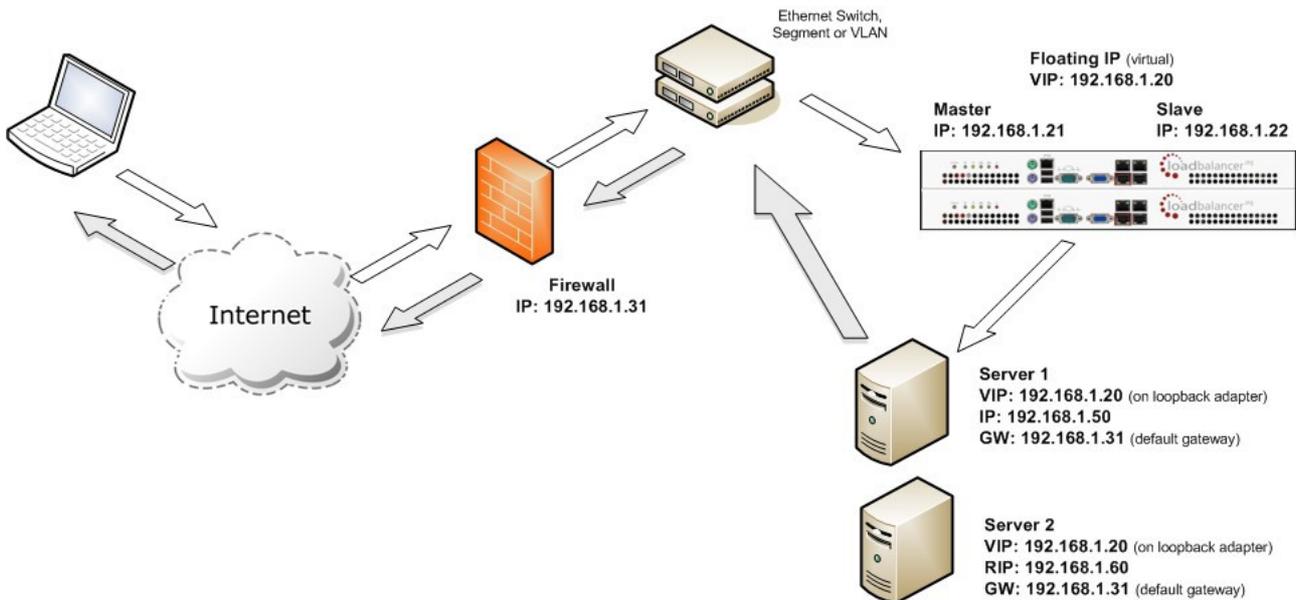
- Recommended for high performance fully transparent and scalable solutions
- Recommended if HTTP cookie persistence is required, also used for several Microsoft applications such as Exchange, Sharepoint & Remote Desktop Services and for overall deployment simplicity since real servers can be on any accessible subnet and no Real-Server changes are required
- Only required for Direct Routing implementation across routed networks (rarely used)

DIRECT ROUTING (DR)

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

Note:

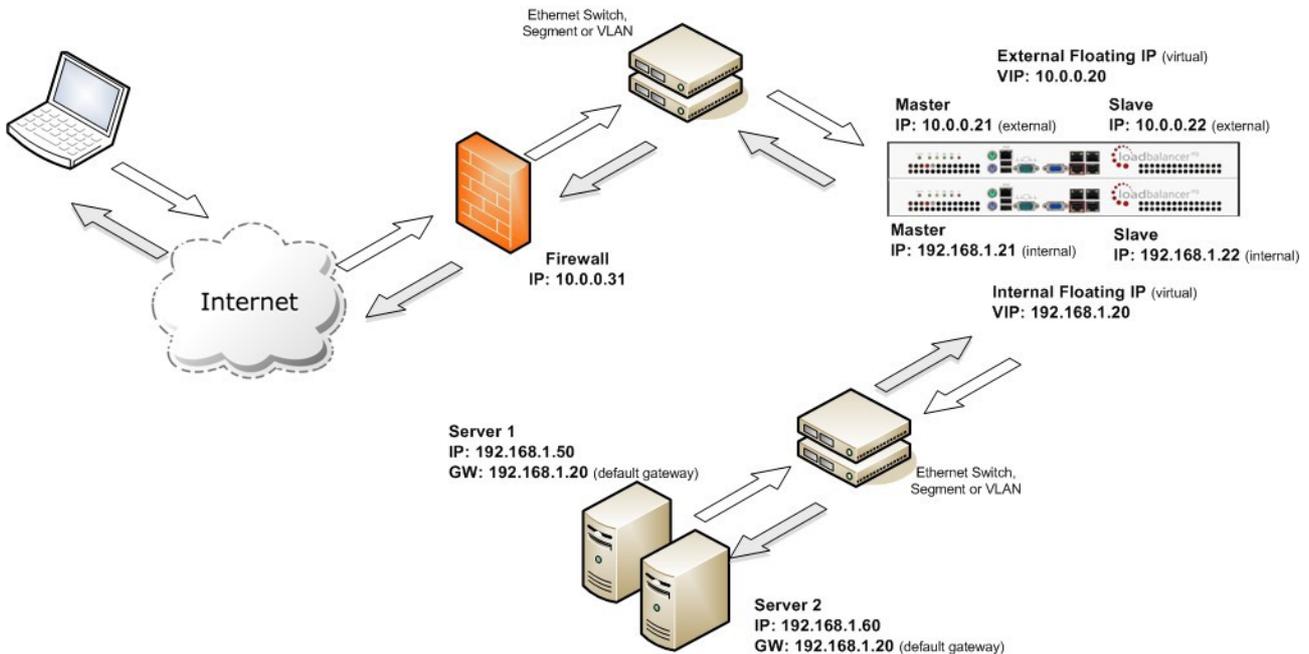
Kemp, Brocade, Barracuda & A10 Networks call this *Direct Server Return* and F5 call it *N-Path*.



- DR mode works by changing the destination MAC address of the incoming packet to match the selected Real Server on the fly which is very fast
- When the packet reaches the Real Server it expects the Real Server to own the Virtual Services IP address (VIP). This means that you need to ensure that the Real Server (and the load balanced application) respond to both the Real Servers own IP address and the VIP
- The Real Servers should not respond to ARP requests for the VIP. Only the load balancer should do this. Configuring the Real Servers in this way is referred to as *Solving the ARP Problem*, please refer to page [79](#) onward for more details
- On average, DR mode is 8 times quicker than NAT for HTTP, 50 times quicker for Terminal Services and much, much faster for streaming media or FTP
- The load balancer must have an Interface in the same subnet as the Real Servers to ensure layer 2 connectivity required for DR mode to work
- The VIP can be brought up on the same subnet as the Real Servers, or on a different subnet provided that the load balancer has an interface in that subnet
- Port translation is not possible in DR mode i.e. having a different RIP port than the VIP port
- DR mode is transparent, i.e. the Real Server will see the source IP address of the client

NETWORK ADDRESS TRANSLATION (NAT)

Layer 4 NAT mode is also a high performance solution, although not as fast as layer 4 DR mode. This is because real server responses must flow back to the client via the load balancer rather than directly as with DR mode.



- The 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 WebUI option: *Cluster Configuration > Layer 4 – Advanced Configuration*, the external interface should be selected
- NAT mode can be deployed in the following ways:
 - 2-arm (using 2 Interfaces), 2 subnets** (as shown above) - One interface on the load balancer is connected to subnet1 and the second interface and Real Servers are connected to subnet2. The VIP is brought up in subnet1. The default gateway on the Real Servers is set to be an IP address in subnet2 on the load balancer. Clients can be located in subnet1 or any remote subnet provided they can route to the VIP
 - 2-arm (using 1 Interface), 2 subnets** - same as above except that a single interface on the load balancer is allocated 2 IP addresses, one in each subnet
 - 1-arm (using 1 Interface), 1 subnet** - Here, the VIP is brought up in the same subnet as the Real Servers. For clients located in remote networks the default gateway on the Real Servers must be set to be an IP address on the load balancer. For clients located on the same subnet, return traffic would normally be sent directly to the client bypassing the load balancer which would break NAT mode. To address this, the routing table on the Real Servers must be modified to force return traffic to go via the load balancer - For more details on 'One-Arm NAT Mode' refer to page [103](#)
- 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 or add additional VIPs for this - please refer to page [102](#) for more details
- NAT mode is transparent, i.e. the Real Server will see the source IP address of the client
- Port translation is possible in NAT mode, i.e. VIP:80 --> RIP8080 is possible

NAT MODE PACKET RE-WRITING

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

The following table shows an example NAT mode setup:

Protocol	VIP	Port	RIP	Port
TCP	10.0.0.20	80	192.168.1.50	80

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

Packet rewriting works as follows:

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

SOURCE	x.x.x.x:34567	DEST	10.0.0.20:80

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

SOURCE	x.x.x.x:34567	DEST	192.168.1.50:80

3) Replies return to the load balancer as:

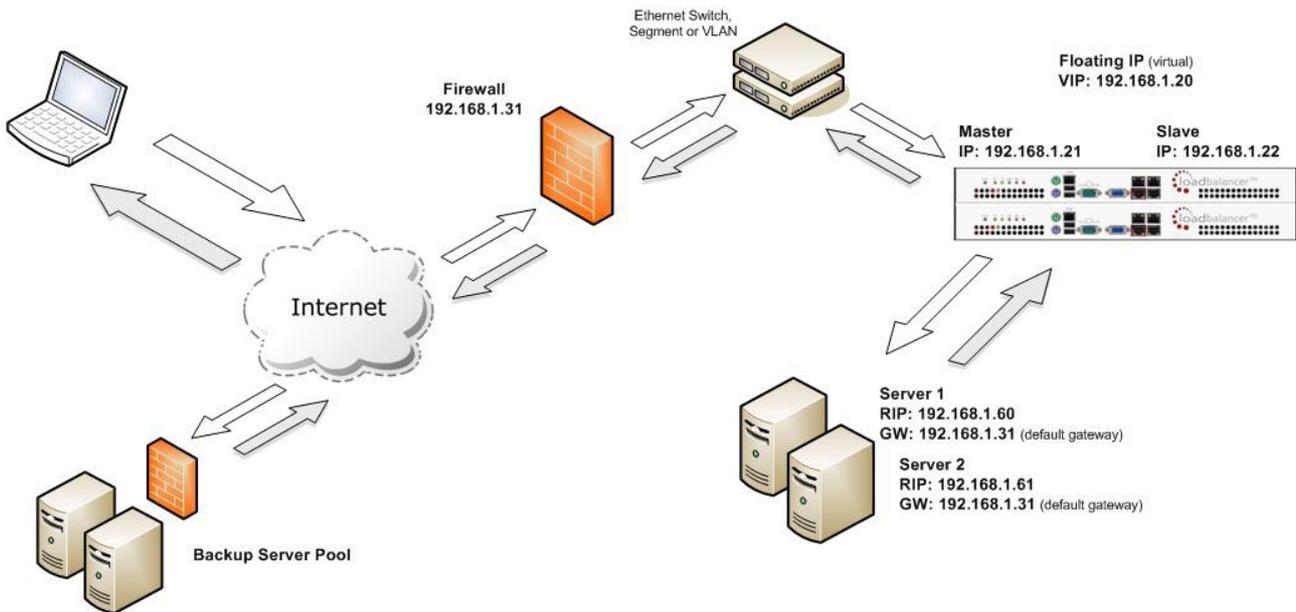
SOURCE	192.168.1.50:80	DEST	x.x.x.x:34567

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:34567

SOURCE NETWORK ADDRESS TRANSLATION (SNAT)

Layer 7 SNAT mode uses a proxy (HAProxy) at the application layer. Inbound requests are terminated on the load balancer, and HAProxy generates a new request to the chosen Real Server. As a result, Layer 7 is a slower technique than DR or NAT mode at Layer 4. Layer 7 is typically chosen when either enhanced options such as SSL termination, cookie based persistence, URL rewriting, header insertion/deletion etc. are required, or when the network topology prohibits the use of the layer 4 methods.



This mode can be deployed in a one-arm or two-arm configuration and does not require any changes to the Real Servers. However, since the load balancer is acting as a full proxy it doesn't have the same raw throughput as the layer 4 methods.

The load balancer proxies the application traffic to the servers so that the source of all traffic becomes the load balancer.

- SNAT mode is a full proxy and therefore load balanced Real Servers do not need to be changed in any way
- Because SNAT mode is a full proxy any server in the cluster can be on any accessible subnet including across the Internet or WAN
- SNAT mode is not transparent by default, i.e. the Real Servers will not see the source IP address of the client, they will see the load balancers own IP address by default, or any other local appliance IP address if preferred (e.g. the VIP address), this can be configured per layer 7 VIP. If required, the clients IP address can be passed through either by enabling TProxy on the load balancer, or for HTTP, using X-forwarded-For headers. Please see pages [147](#) and [116](#) respectively for more details.
- SNAT mode can be deployed using either a 1-arm or 2-arm configuration

Note:

For detailed configuration examples using various modes, please refer to chapter 11 starting on page [215](#).

Which Load Balancing Method should I Use?

MODE SUMMARY

Layer 4 DR Mode offers the best performance and requires limited physical Real Server changes. The load balanced application must be able to bind to the Real Servers own IP address and the VIP at the same time. This mode requires the “ARP Problem” to be solved as described starting on page [79](#).

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, Connection Broker integration and works very well with either Pound or STunnel when SSL termination is required. It does not require any changes to the Real 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 mode offers the best possible performance since replies go directly from the Real Servers to the client, not via the load balancer. It's also relatively simple to implement. Ultimately, the final choice does depend on your specific requirements and infrastructure.

Note:

If you are using Microsoft Windows Real Servers (i.e. the backend 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

1. Remove all packaging and rack mount the appliance if required
2. Connect the power lead from the power socket to the mains or UPS

Note:

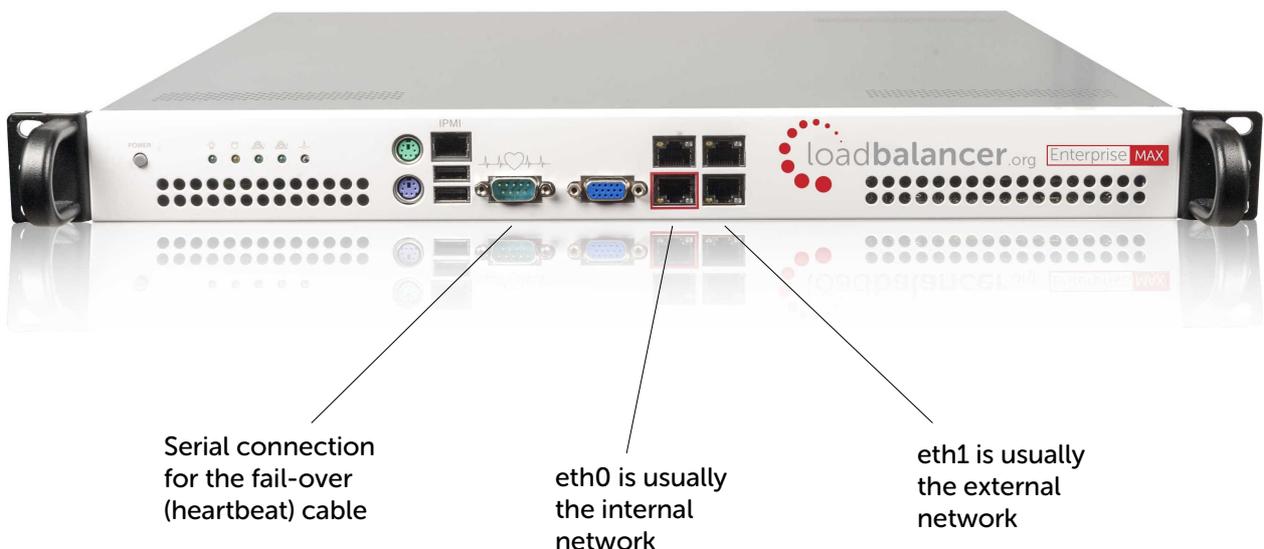
the power supply is an auto sensing unit (100v to 240v).

3. Connect a network cable from your 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 again, this is not mandatory
4. For a clustered hardware pair, the units must be able to communicate either via network (ucast), via serial cable or both. By default, ucast only is used. If serial is preferred or you want to use both methods, connect a serial cable between the two appliances

Note:

If a serial cable is used, Heartbeat must be configured for this using the WebUI option: *Cluster Configuration > Heartbeat Configuration* and enabling 'Serial'

5. Attach a monitor to the VGA port and keyboard to one of the USB ports
6. Check mains power is on and press the power switch to start the appliance. The fans should start & front panel LED's should light



Note:

The above image shows the Enterprise MAX, for other models please refer to page [266](#) in the Appendix.

The Virtual Appliance – Hypervisor Deployment

SUPPORTED HYPERVISORS

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

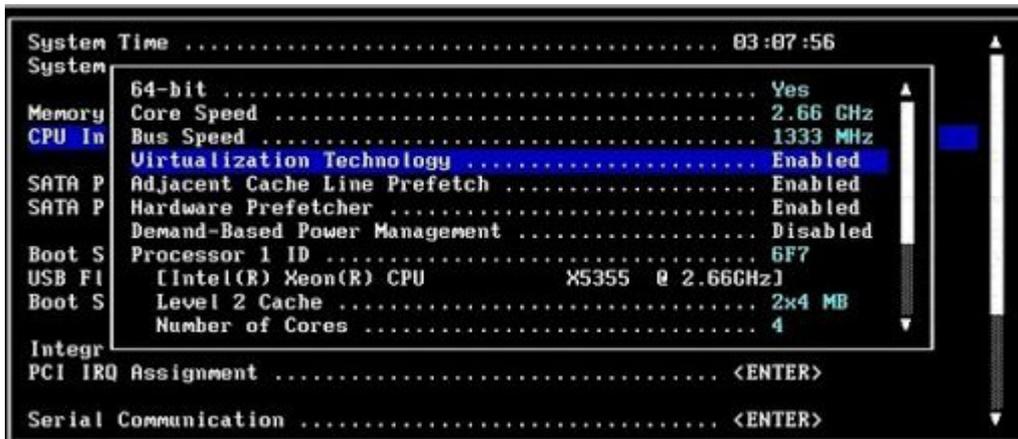
- VMware ESXi : v4.0 & later
- Virtual Box : v4.0 & later
- Microsoft Hyper-V : v2012 & later
- KVM : Kernel version v2.6.20 & later
- XEN : v6.0 & later

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.

Once deployed, the VA is allocated the following resources by default:

- 1 vCPU
- 2GB RAM
- 8GB disk

Typically these allocations don't need changing. However, the number of vCPU's and the allocated RAM can easily be increased if needed. Please contact support@loadbalancer.org if you need assistance assessing if this would be advantageous for your deployment.

DOWNLOADING THE APPLIANCE

All downloads are accessible from the following location: <http://www.loadbalancer.org/resources/free-trial>

To access the downloads, enter your name, email address and phone number (optional), specify the application that you'll be load balancing (optional) and select the Hypervisor type. Once the details have been entered, click **Download Now**. The various download links will then be presented on screen and we'll also send you an email containing the same links.

Once downloaded, extract the archive using your preferred utility. The download also includes a quickstart guide which covers the VA deployment process in more detail.

Note:

All information provided is 100% confidential. We may follow up with an email or phone call to see how you're getting on with the trial and offer assistance, but under no circumstances will Loadbalancer.org share your details with a third party.

VMWARE DEPLOYMENT

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

- For vSphere Client use: **File > Deploy ovf Template**
- For VMware Workstation use: **File > Open**
- For VMware Player use: **Player > File > Open**

VMWARE TOOLS

VMware tools are pre-installed on the appliance which enables basic console control functions such as power on/off etc. The installed version of the various kernel modules and drivers is controlled by Loadbalancer.org at build time to ensure that only stable, fully tested versions are deployed. If the tools are later upgraded, these drivers and modules may be over-written. Therefore we do not recommend a full tool re-installation. If you do want to update the basic tool functionality (i.e. without affecting the installed drivers and modules) please follow the steps listed in [this Loadbalancer.org blog](#).

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-Vv8
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* pane
- Once one of the above is done, repeat steps 1-6 to create the second virtual machine.

Windows 2012 and Later

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-V3v8
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.

LINUX INTEGRATION SERVICES

Linux Integration Services are pre-installed by default. Therefore manual installation is not required.

KVM DEPLOYMENT

The following steps should be followed on the KVM host:

1. Extract the archive to /var/lib/libvirt/images/
2. virsh define Loadbalancer*.xml
3. virsh start Loadbalancer*

Note:

Network cards are set to NAT by default so adjust as needed before powering on. Also, please refer to the included XML file for additional configuration notes.

XEN DEPLOYMENT

The following steps should be followed on the XEN host:

1. Extract the archive
2. Import the **xva** file into XEN

Note:

For details of the cloud based products, please refer to the relevant quick start guide available in the [documentation library](#).

Initial Network Interface Configuration

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

192.168.2.21/24 (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

Note:

For the VA, four NICs are included but only eth0 is connected by default at power on. If the other NICs are required, these should be connected using the network configuration screen within the Hypervisor.

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

This will overwrite the current configuration.
If you do not wish to proceed please enter CTRL + c.

Static IP address (eg. 192.168.0.26) : 192.168.111.80/18

Default gateway (eg. 192.168.0.1) : 192.168.64.1

DNS Servers

Primary (eg. 192.168.0.250) : 8.8.8.8

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 WebUI as shown below:

Summary of settings

Static IP address: 192.168.111.80/18

Default gateway: 192.168.64.1

DNS servers: 8.8.8.8

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

`http://192.168.111.80:9080/lbadmin/`

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 WebUI menu option: *Local Configuration > Network Interface Configuration* (to access the WebUI please refer to pages [39](#) and [42](#)) or by using Linux commands as explained in the following section.

At this stage you will also be asked if you're recovering from node (i.e. master or slave) failure as shown below:

Are you recovering from node failure?

Only use this facility if your master or slave appliance has failed and you'd like this new appliance to be a replacement. The configuration will be recovered from the remaining node and the HA clustered pair will be restored without disrupting running services

(If you are simply deploying a new appliance, hit N)

Do you want to continue? [y/N]

-

As mentioned in the text, if you're simply deploying a new appliance, click "N"

Note:

For more details on node recovery using this option, please refer to chapter 15 , “Backup & Restore and Disaster Recovery” starting on page [252](#).

USING LINUX COMMANDS

at the console or via an SSH session login 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
```

Note:

Setting the IP address in this way is temporary, the IP address **MUST** be set via the WebUI 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

For a hardware appliance, 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 the default heartbeat configuration is used - i.e. heartbeat is configured to communicate over the network only.

Log in to the console:

Username: root
Password: loadbalancer

APPLIANCE CONFIGURATION USING LINKS

Once logged into the console, the text based 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
```

Log in to Links:

Username: loadbalancer

Password: loadbalancer

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

Note:

The preferred configuration method is the WebUI which can be accessed via a browser as detailed on page [39](#).

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. The Enterprise R20 and Enterprise MAX include IPMI support, iDRAC is included on the Enterprise 10G & Ultra. For details on configuring both IPMI & iDRAC please refer to pages [267](#) & [271](#) in the Appendix.

The appliance can be remotely accessed using the following tools:

- HTTP/HTTPS Web Browser → Web User Interface (WebUI)
- OpenSSH (Linux hosts) or [PuTTY](#) (Windows hosts) → Secure Shell Access
- OpenSCP (Linux hosts) or [WinSCP](#) (Windows hosts) → Secure File Transfer

ACCESSING THE WEBUI

The WebUI 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 WebUI menu option: *Maintenance > Passwords*.

Note:

By default, the WebUI is accessible via HTTP and HTTPS. For details on limiting this to HTTPS

only and configuring HTTPS settings, please refer to page [69](#).

- Using a web browser, access the WebUI using the following URL:

<http://192.168.2.21:9080/lbadmin/>

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

If you prefer you can use the HTTPS administration address:

<https://192.168.2.21:9443/lbadmin/>

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

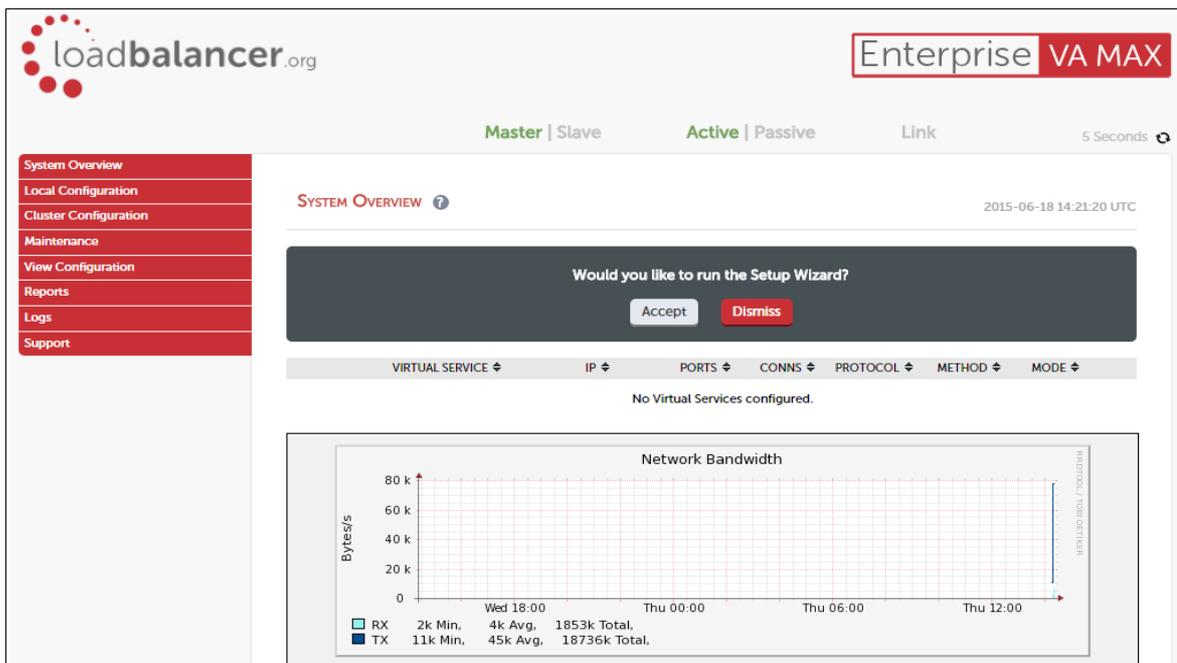
- Login to the WebUI:

Username: loadbalancer

Password: loadbalancer

Note:

To change the password, use the WebUI menu option: *Maintenance > Passwords*.



- Once logged in, you'll be asked if you want to run the web based setup wizard. If you click **Accept** the Layer 7 Virtual Service configuration wizard will start. If you prefer to configure the appliance manually, simply click **Dismiss**.

Note:

A number of compatibility issues have been found with various versions of Internet Explorer. The WebUI has been tested and verified using both Firefox & Chrome.

CONFIGURING THE APPLIANCE USING THE WIZARD

The wizard can be used to setup one or more Layer 7 Virtual Services and associated Real Servers. Layer 4 services must be configured manually.

To run the wizard:

1. First, set the IP address using one of the methods described on page [36](#)
2. Open the WebUI and start the wizard by clicking the **Accept** button shown above, or by using the WebUI menu option: *Cluster Configuration > Setup Wizard* and clicking **General Layer 7 Virtual Service**
3. Define the required Virtual Service settings as shown in the example below:

SETUP WIZARD - GENERAL LAYER 7 VIRTUAL SERVICE

Load balancer configuration

	Master	Slave
Hostname	lbmaster	<i>Not configured</i>
Static IP Addresses	eth0	192.168.111.222/18
Floating IP Addresses		

Create a new Layer 7 Virtual Service

Label

Virtual Service	IP Address	<input style="width: 150px;" type="text" value="192.168.111.225"/>
	Ports	<input style="width: 150px;" type="text" value="80"/>

Layer 7 Protocol

[Create Virtual Service](#)

Select the Layer 7 protocol to be handled by this Virtual Service.
Advanced options may be set by editing this Virtual Service once it has been created.

4. Click **Create Virtual Service**
5. Now continue and add the associated Real Servers as shown below:

Attach Real Servers

Label	IP Address	Port	Weight	
<input type="text" value="RIP1"/>	<input type="text" value="192.168.111.226"/>	<input type="text" value="80"/>	<input type="text" value="100"/>	<input type="button" value="✕"/>
<input type="text" value="RIP2"/>	<input type="text" value="192.168.111.227"/>	<input type="text" value="80"/>	<input type="text" value="100"/>	<input type="button" value="✕"/>

- Use the **Add Real Server** button to define additional Real Servers, once all are defined click **Attach Real Servers**
- Finally reload HAProxy using the **Reload HAProxy** button in the blue box at the top of the screen or by using the WebUI menu option: *Maintenance > Restart Services* and clicking **Reload HAProxy**

Note:

Running the wizard again will permit additional Layer 7 VIPs and associated RIPs to be defined.

Note:

To restore manufacturer's settings use the WebUI menu option: *Maintenance > Backup & Restore > Restore Manufacturer's Defaults*. This will reset the IP address to 192.168.2.21/24.

CONFIGURING THE APPLIANCE MANUALLY

Please refer to Chapter 6 "Configuring Load Balanced Services" starting on page [72](#) for more information.

Chapter 5 – Appliance Management

Network Configuration

PHYSICAL INTERFACES

The Enterprise R20, Enterprise MAX, Enterprise 10G and all virtual models have 4 network interfaces. The Enterprise Ultra has 6 network interfaces. For the VA, only the first interface is connected by default, the other interfaces can be connected when required using the Hypervisor's management interface. If multiple logical interfaces are required, these can be added simply by specifying multiple IP addresses as shown on the following page. If multiple cables must be connected, an external switch can be used.

Typically, the main reason for using all 4 or 6 interfaces is when bonding (e.g. 802.3ad) is required in a two-arm NAT mode (layer 4) or two-arm SNAT mode (layer 7) highly available configuration.

CONFIGURING IP ADDRESSES

As mentioned in the previous chapter, initial network settings can be easily configured using the Network Setup Wizard. For more details please refer to page [36](#).

IP addresses can also be configured using the WebUI menu option: *Local Configuration > Network Interface Configuration*. If a single interface is required, *eth0* is typically used. If 2 interfaces are required, *eth0* is typically 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 simple one-arm configuration, you would just need to configure the IP address and subnet mask for one interface, e.g. *eth0* and if there are remote clients, the relevant default gateway. Both IPv4 and IPv6 addresses can be configured.

CIDR notation is used to specify IP addresses and subnet masks. For example, to specify an IP address of 192.168.2.100 with a subnet mask of 255.255.255.0, then 192.168.2.100/24 would be entered in the relevant interface field as shown in the example below:

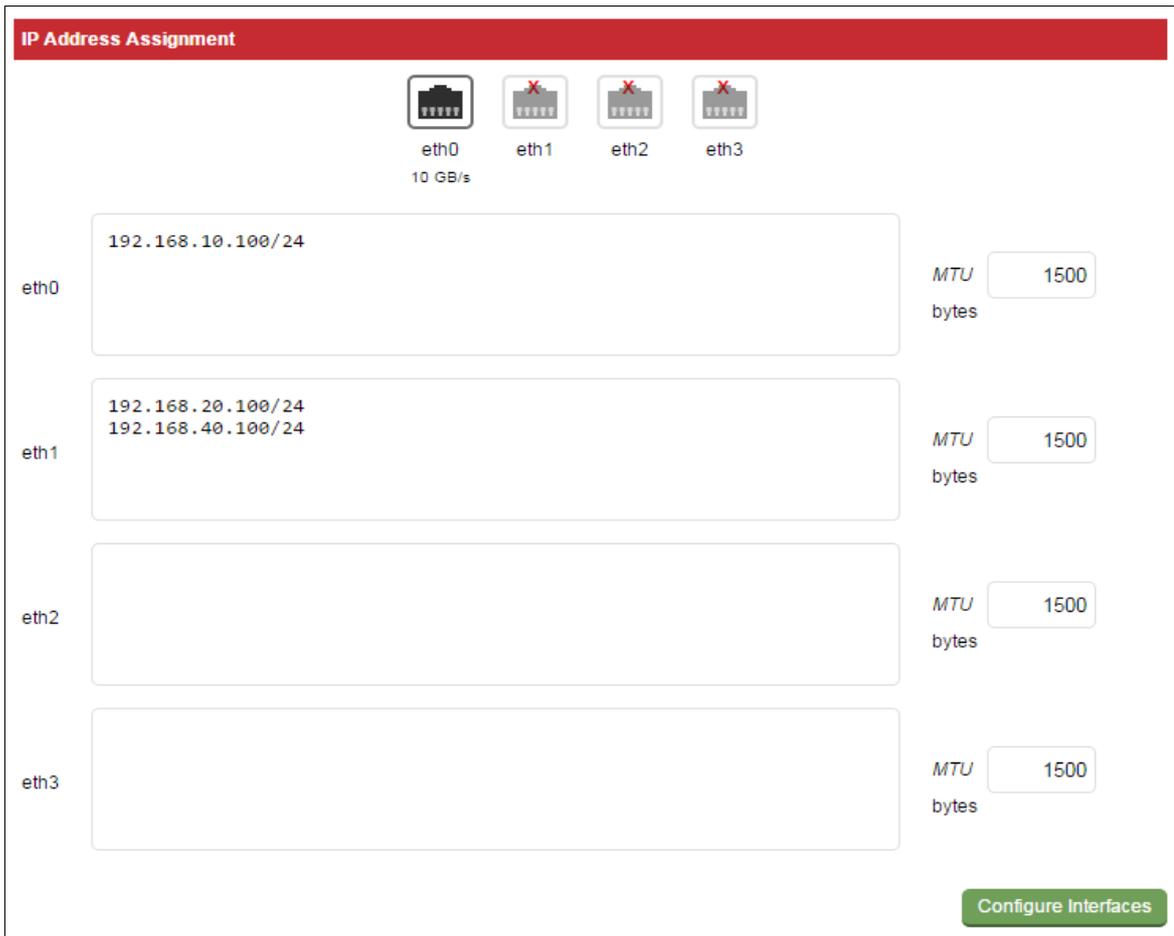
The image shows a screenshot of a network configuration interface. On the left, the interface name 'eth0' is displayed. To its right, a text input field contains the IP address and subnet mask '192.168.2.100/24'.

Note:

Please refer to page [272](#) in the the appendix for more details on CIDR notation.

To set IP address(es):

1. Using the WebUI, navigate to: *Local Configuration > Network Interface Configuration*
2. Assign the required IP address/mask, multiple addresses can be assigned as shown below:



3. 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.

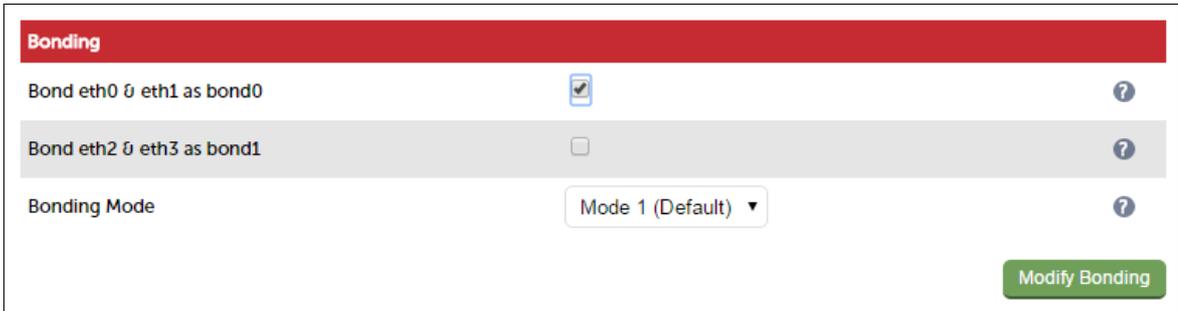
Note:
For the VA, four NICs are included but only eth0 is connected by default at power on. If the other NICs are required, these should be connected using the network configuration screen within the Hypervisor.

CONFIGURING BONDING

The appliance enables interfaces to be bonded together. The WebUI enables eth0/eth1 and eth2/eth3 to be bonded.

To Configure bonding of 2 interfaces:

1. Using the WebUI, navigate to: Local Configuration > Network Interface Configuration
2. If you want to bond eth0 and eth1, check the box named Bond eth0 & eth1 as bond0



3. Change the bonding mode if required

Bonding Modes Supported:

Mode 0 - Balance round robin. Transmits packets in a numerical order from the first available slave through to the last.

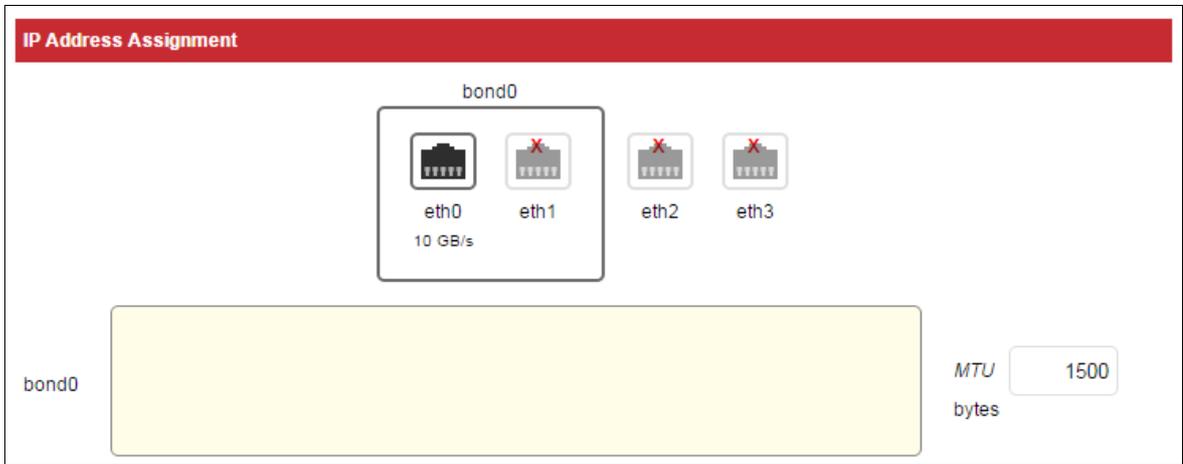
Mode 1 - Active Backup (default). This places one of the interfaces in a backup state and will only become active is the link is lost to the active interface. This mode provides fault tolerance.

Mode 4 - 802.3ad. Dynamic link aggregation mode. This mode requires a switch that supports IEEE 802.3ad.

Note:

After changing the bonding mode a restart of the appliance is required for the setting to take effect.

4. Click **Modify Bonding**
5. 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 and only the bond address will apply. If bonding is later disabled these addresses will be re-applied to the interfaces.

6. Enter the IP address for bond0 and click **Configure Interfaces**
7. If the bonding mode has been changed, restart the appliance used the WebUI menu option: *Maintenance > System Control* and clicking **Restart Load Balancer**

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 (master/slave communication) related issues.

Note:

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 802.1Q VLAN support can be enabled to load balance clusters on multiple VLANs.

In access mode, switch ports are 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:

1. Using the WebUI, navigate to: *Local Configuration > Network Configuration*
2. In the VLAN section select the required interface (e.g. eth0)
3. Enter the VLAN ID (e.g. 250)
4. Click **Add VLA**
5. An extra IP Address Assignment field named eth0.250 will be created as shown below, the required IP address should be entered in this field

IP Address Assignment

eth0 10 GB/s
eth0.250 10 GB/s
eth1
eth2
eth3

eth0
192.168.111.222/18
MTU 1500 bytes

eth0.250
MTU 1500 bytes
Delete eth0.250

6. Click **Configure Interfaces**
7. To delete the VLAN definition, click the appropriate **Delete** button

Note:

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 MTU SETTINGS

To set the MTU setting for an interface:

1. Using the WebUI, navigate to: *Local Configuration > Network Configuration*

eth0
192.168.10.100/24
MTU 1500 bytes

2. Enter the required MTU setting
3. Click **Configure Interfaces**

CONFIGURING DEFAULT GATEWAY & STATIC ROUTES

To set the Default Gateway for IPv4 and Ipv6:

1. Using the WebUI, navigate to: *Local Configuration > Routing*
2. In the Default Gateway section define the default gateway as shown in the example below:

Default Gateway			
IP v4	<input type="text" value="192.168.64.1"/>	via interface	<input type="text" value="auto"/> ?
IP v6	<input type="text"/>	via interface	<input type="text" value="auto"/> ?

3. Click **Configure Routing**

To configure Static Routes:

1. Using the WebUI, navigate to: *Local Configuration > Routing*
2. In the Static Routes section configure the subnets & gateway addresses shown in the example below:

Static Routes			
Subnet	<input type="text" value="10.12.0.0/16"/>	via gateway	<input type="text" value="192.168.64.250"/>
Subnet	<input type="text" value="10.20.0.0/16"/>	via gateway	<input type="text" value="192.168.64.251"/>
Subnet	<input type="text"/>	via gateway	<input type="text"/>

3. Click **Configure Routing**

Note:

Unlimited static routes can be defined, additional blank rows will be added to the WebUI screen as they're used.

Note:

If you require a custom gateway for a particular VIP, this can be achieved using Policy Based Routing. To configure a VIP to return it's traffic via a custom gateway rather than via the default gateway, create a simple configuration file in `/etc/pbr.d/` with a `.conf` extension for that VIP, e.g. `/etc/pbr.d/VIP1.conf`. The file must contain the desired gateway and the VIP address using the following format:

```
GW=172.16.200.1
VIP=172.16.200.37
```

Optional options:

ROUTES= Adding this variable with the option "local" as below forces the script to only copy the link local route for the VIP specified, not all link local routes.

FROM= Allows you to provide additional FROM rules, either a single address/subnet or multiple addresses/subnets which would need to be space separated.

Examples:

```
ROUTES="local"
```

FROM="10.10.10.10/32 12.0.0.0/8"

Please contact our support team: support@loadbalancer.org if you need any assistance.

CONFIGURING HOSTNAME & DNS CONFIGURATION

To set the *Hostname, Domain & DNS servers*:

- Using the WebUI, navigate to: *Local Configuration > Hostname & DNS*

HOSTNAME & DNS			
Hostname		<input type="text" value="lbmaster"/>	?
Domain Name		<input type="text" value="localhost"/>	?
Domain Name Server	Primary	<input type="text" value="8.8.8.8"/>	?
	Secondary	<input type="text"/>	?
	Tertiary	<input type="text"/>	?
			<input type="button" value="Update"/>

- Specify the required *Hostname*, by default this is set to **lbmaster**
- Specify the Domain name, by default this is set to **localhost**
- Specify the required DNS servers
- Click **Update**

System Date & Time and NTP Server Configuration

AUTO CONFIGURATION USING NTP SERVERS

To configure NTP:

- Using the WebUI, navigate to: *Local Configuration > System Date & Time*

SYSTEM DATE & TIME

Current system time 2015-03-19 12:24:18 UTC

System Timezone UTC ▼

NTP Servers

Set Timezone & NTP

Date 2015 ▼ – Mar ▼ – 19 ▼

Time 12 : 24

Set Date & Time

2. Select the required *System Timezone*
3. Define your NTP servers using the *NTP Servers* fields
4. Click **Set Timezone & NTP**

MANUAL CONFIGURATION

To manually set the date & time:

1. Set the data & time using the *Date & time* fields
2. Click **Set Date & Time**

Note:

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 date and time on 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:

1. Using the WebUI, navigate to: *Local Configuration > Physical Advanced Configuration*

2. Enter an appropriate IP address in the *IP Address* field
3. Enter an appropriate port in the *Port* field
4. Click **Update**

Note:

For a clustered pair, this setting must also be manually configured on the slave.

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:

1. Using the WebUI, navigate to: *Local Configuration > Physical Advanced Configuration*

2. Enter an appropriate IP address or hostname in the *Smart Host* field
3. Click **Update**

Note:

For a clustered pair, this setting must also be manually configured on the slave.

Syslog Server Configuration

The appliance supports the ability to write all logs either locally, to an external Syslog Server or both. The Syslog server may be specified by IP address or hostname.

To configure a Syslog server:

1. Using the WebUI, navigate to: *Local Configuration > Physical Advanced Configuration* and scroll to the *Logging* section

Logging

Rate limit interval

Rate limit Burst limit

Log Destination

 Local Files

 Remote syslog Server

 Both

Remote syslog Server IP

Remote syslog Server Port

Remote syslog Server Protocol TCP ▾

Remote syslog Server Template

2. Define whether logs should be written to *Local Files*, a *Remote Syslog Server* or *Both*
3. If *Remote Syslog Server* or *Both* is selected, the following options also apply:

Option	Description
Rate Limit	The Syslog Rate Limit Interval determines the amount of time that is being measured for rate limiting. By default this is set to 5 seconds.
Rate Limit Burst Limit	The Syslog Rate Limit Burst defines the amount of messages, that have to occur in the time limit of Syslog Rate Limit Interval, to trigger rate limiting. Here, the default is 200 messages.
Remote Syslog Server IP	The server may be specified by IP address or hostname. If you use a hostname, make sure DNS is correctly configured on the loadbalancer.
Remote Syslog Server Port	Specify the Remote Syslog Server port.
Remote Syslog Server Protocol	Select the communications protocol, either TCP or UDP.
Remote Syslog Server Template	Specify a Remote Syslog Server template (string format).

4. Click **Update**

Note:

For a clustered pair, this setting must also be manually configured on the slave.

SNMP Configuration

The appliance supports SNMP. Typical SNMP settings can be configured using the WebUI.

To Configure SNMP:

1. Using the WebUI, navigate to: *Local Configuration > SNMP Configuration*

The screenshot shows the 'SNMP CONFIGURATION' page in a web interface. It features three input fields: 'SNMP community string', 'SNMP location', and 'SNMP contact'. Each field has a question mark icon to its right. An 'Update' button is located at the bottom right of the form.

2. Set the required settings (If you leave the fields blank, default values will be applied)
3. Click **Update**

Note:

For a clustered pair, this setting must also be manually configured on the slave.

Note:

Please refer to page [245](#) for details of the various OIDs and associated MIBs for the appliance.

Installing License Keys

License keys are required for all appliances. At initial power up, the VA will display the following message:

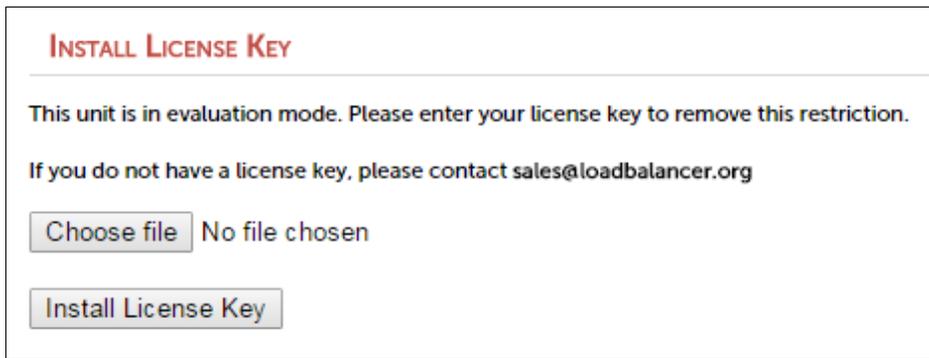
WARNING:
This evaluation version will expire in 30 days. **Please enter your license key** to remove this restriction.

The hardware appliance will display the following message:

WARNING:
This appliance is unregistered. **Please enter your license key** within 30 days to activate your appliance.
If you do not have your license key please contact sales@loadbalancer.org

To install the license:

1. Using the WebUI, navigate to: *Local Configuration > License Key*



INSTALL LICENSE KEY

This unit is in evaluation mode. Please enter your license key to remove this restriction.

If you do not have a license key, please contact sales@loadbalancer.org

Choose file No file chosen

Install License Key

2. Browse to the license file provided when the appliance was purchased
3. Click **Install License Key**

Running OS Level Commands

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

To run an OS level command:

1. Using the WebUI, navigate to: *Local Configuration > Execute Shell Command*



EXECUTE SHELL COMMAND

Execute shell command

2. Enter the relevant command in the field
3. Click **Execute Shell Command**
4. The results of the command as well as 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, RIPs and other settings will be removed and the IP address configured for eth0 will be set to 192.168.2.21/24.

USING THE WEBUI

To restore settings:

1. Using the WebUI, navigate to: *Maintenance > Backup & Restore > Restore Tab*
2. Click **Restore Manufacturer's Defaults**

Once restored, restart the appliance to complete the process.

USING THE CONSOLE/SSH SESSION

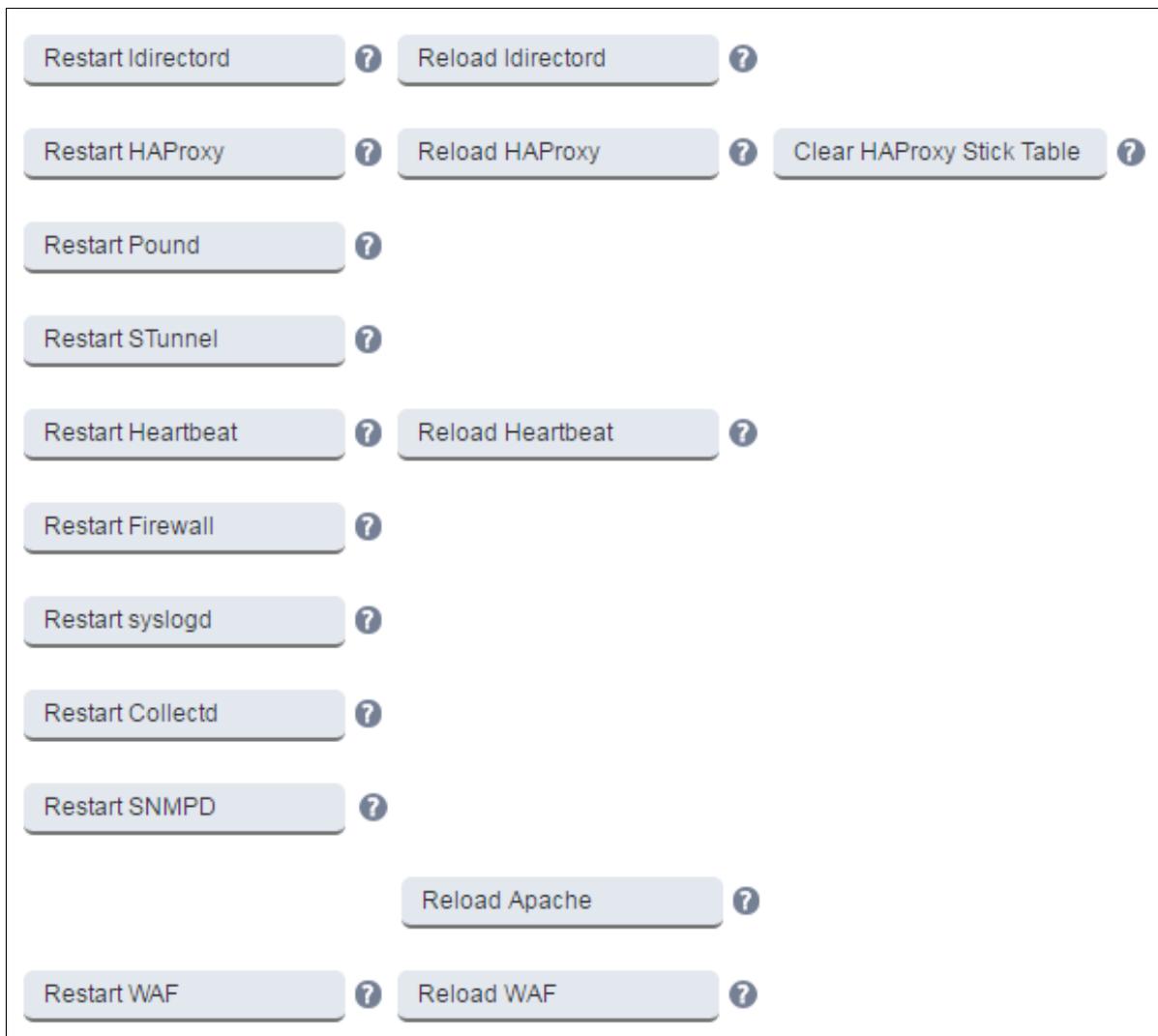
Run the following command:

```
lbrestore
```

Once restored, restart the appliance to complete the process.

Restarting & Reloading 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 and 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.

Note:

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 Service. Restarting Pound will result in a loss of SSL termination services during the restart.

Restart STunnel

Restart STunnel SSL Termination Service. 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

This option forces heartbeat to stop (for the active member of an HA pair the floating IP's will also be taken down, for a single unit they will be left up) then after an appropriate delay heartbeat is reloaded.

Restart Firewall

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

Restart Syslogd

Restart the syslog service.

Restart Collectd

Restart the graphs data collector service.

Restart SNMPD

Restart the SNMP service.

Reload Apache

Reload the Apache service.

Restart WAF

Restart the WAF. This will drop all current connections are re-read the config.

Reload WAF

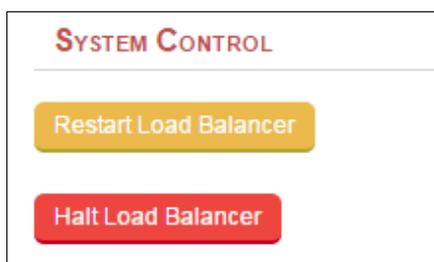
Reload the WAF and re-read the config.

Appliance Restart & Shutdown

The appliance can be restarted or shutdown using the WebUI.

To restart or shutdown the appliance:

1. Using the WebUI, navigate to: *Maintenance > System Control*



2. Select the required option:

Restart Load Balancer – *Shutdown and restart the appliance*

Halt Load Balancer – *Shutdown and halt the appliance*

Appliance Software Updates

Loadbalancer.org continually develop and add new and improved features to the appliance. To ensure that customers can benefit from this and can also receive bug fixes 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. A security updates only option is also available for customers that don't require the benefits of our complete support package.

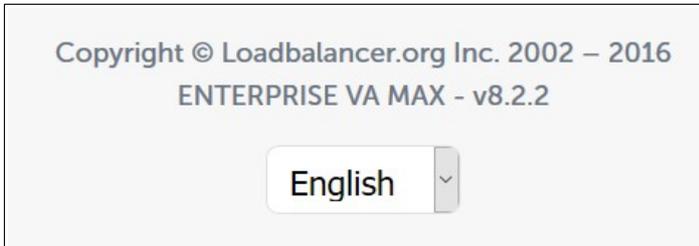
Note:

Since services may be restarted during the update process we recommend performing the update during a maintenance window.

For some updates a full appliance restart is required. In these cases a restart notification message will be displayed after the update is complete.

CHECKING THE CURRENT SOFTWARE VERSION

The software version is displayed at the bottom of the WebUI as shown in the example below:



ONLINE UPDATE

To perform an online update:

1. Using the WebUI, navigate to: *Maintenance > Software Update*
2. Select **Online Update**
3. If the latest version is already installed, the following message will be displayed:

Information: Version v8.1 is the current release. No updates are available.

4. If an update is available, Information similar to the following will be displayed (shows a truncated version of the actual text for the v8.1.1 to v8.2 upgrade):

Online Update

Information: You are about to update a loadbalancer appliance running on: vmware. If this is incorrect do not continue and contact support@loadbalancer.org

Online updates are only available if your organisation has a valid authorisation key.
An authorisation key may be obtained from 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 v8.1.1 to v8.2

Changes in this release:

HAProxy

- HAProxy updated to 1.7
- ACL to redirect IP block to specific backend.
- Syn block was delaying stick table synchronisation.
- Its now clearer which real servers have re-encryption enabled.

STunnel

- SNI Matching now available.
- Disable TLSv1.

Heartbeat

- Email Addresses with sub domains now allowed for alerting.

WARNING:HAProxy Will be stopped as part of this update. Causing an interruption in service.

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

Note that this is a large update, and may take several minutes to complete. Please do not stop the web browser, or move to another page, whilst the Loadbalancer.org logo is spinning. When the update is complete, the message *Update completed successfully* will be displayed.

5. Click **Online Update**
6. Once complete (the update can take several minutes depending on download speed and upgrade version) the following message is displayed:

Information: Update completed successfully.

7. If there are any specific post upgrade requirements such as a service restart these will be displayed on the screen after the installation completes.

Notes:

- As indicated in the WebUI, we recommend that you should backup your XML configuration and firewall script (if changes have been made) 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 a valid DNS server is specified (*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:

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

SOFTWARE UPDATE

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: No file selected.

Checksum: No file selected.

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

UPDATING A CLUSTERED PAIR

Note:

Since services may be restarted during the update process, we recommend performing the update during a maintenance window.

To update a clustered pair:

1. Perform the update on the slave first. The updates are incremental, so we recommend installing each update in turn, ignoring calls to restart services or reboot the appliance until all available updates have been installed and the appliance is fully up to date.
2. Next, restart services or reboot the appliance as directed.

- Now update the master unit in the same way.

Note:

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 [202](#).

Firewall Configuration

Note:

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:

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

MANUAL FIREWALL CONFIGURATION

The firewall can be configured manually using the WebUI 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 [104](#)).

To configure custom firewall rules:

- Using the WebUI, navigate to: *Maintenance > Firewall Script*
- The following screen will be displayed:

FIREWALL SCRIPT

```

1  #!/bin/sh
2  # $Id: rc.firewall 4933 2014-10-23 11:27:11Z mark $
3
4  #
5  # User firewall script for Loadbalancer.org appliance.
6  #
7
8
9
10 # Please note:
11 #     Most configurations will not require any changes to be made to
12 #     this script.
13 #
14 #     Administrators will only need to modify this script if their
15 #     needs are not met by the lock-down wizard, auto-NAT, and
16 #     automatic firewall mark functions of the web interface.
17
18
19
20 ##### One-arm NAT Mode #####
21 # For one-arm NAT, ICMP re-directs will need to be disabled.
22 # (1 = on, 0 = off)
23 #echo "0" >/proc/sys/net/ipv4/conf/all/send_redirects
24 #echo "0" >/proc/sys/net/ipv4/conf/default/send_redirects
25
26
27
28 ##### Manual Firewall Marks #####
29
30 # Example: Associate HTTP and HTTPS with Firewall Mark 1:
31 #VIP1="10.0.0.66"
32 #iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1

```

[Update](#)

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

```

echo "Firewall Activated"
exit 0;

```

- Click **Update**

Note:

For a clustered pair, firewall script changes must also be manually configured on the slave.

Note:

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 WebUI menu 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 WebUI 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 uses two files:

- rc.lockdownwizard* contains the script that can be changed

- `rc.lockdownwizard.conf` contains a set of variable definitions that is written automatically when **Update firewall lock down** is clicked. The file depends on the `rc.lockdownwizard` script and the load balancers configuration. This file should not be changed manually.

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 writes the definitions `rc.lockdownwizard.conf`.

You can modify `rc.lockdownwizard` via ssh or from the web interface using the **Modify the firewall lock down wizard script** button. Apart from this link there is no other influence from the WebUI.

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 the master and slave, and the admin related 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. Re-running the firewall lockdown wizard will adapt the `rc.lockdownwizard.conf` definitions file automatically – any changes made to the script `rc.lockdownwizard` will remain when you re-run the firewall lockdown wizard.

To run the lock-down wizard:

1. Using the WebUI, navigate to: *Maintenance > Firewall Lock Down Wizard*
2. The following screen will be displayed:

3. Define your administration subnet/host in the *Administration subnet* field

Note:

Make sure that the subnet mask is correct – by default a /24 mask is used

To lock down access to a single host use <IP address>/32, e.g. 192.168.2.1/32.

4. Click **Update firewall lock down**

Note:

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

To disable the lock-down script:

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

Note:

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 modify this setting:

1. Using the WebUI, navigate to: *Local Configuration > Physical – Advanced Configuration*
2. Use the following section:

The screenshot shows a configuration page for the Firewall. A red header bar contains the word "Firewall". Below it, the "Connection Tracking table size" field is set to "524288" with the unit "connections" to its right. A help icon (question mark) is visible on the right side of the field.

3. Set the required value using the *Connection Tracking table size* field
4. Click **Update**

Note:

For a clustered pair, this setting must also be manually configured on the slave.

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	Description (<i>see also the group table below</i>)
loadbalancer	loadbalancer	config *	appliance administration account
reportuser	reportuser	report	viewing the appliance configuration, reports & logs
maintuser	maintuser	maint	same as reportuser plus can also take servers on/off line & create the support download archive file

* It's not possible to change the default group for user 'loadbalancer'

Note:

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

The permissions for each group are shown below:

Group	Menu/Permissions							
	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:

- Using the WebUI, navigate to: *Maintenance > Passwords*
- In the following section, click the **Modify** button next to the relevant user

The screenshot shows a web interface titled "PASSWORDS". It contains a list of three users, each with a "Modify" button and, for some, a "Delete" button:

User	Modify	Delete
loadbalancer	Modify	
reportuser	Modify	Delete
maintuser	Modify	Delete

- Now change the password for the selected user:

The screenshot shows a form for changing the password for the selected user. The fields are:

- Username:** loadbalancer
- Password *:** [Empty text input field]
- Re-enter Password *:** [Empty text input field]

Note:

Passwords cannot contain the double quotation mark (").

4. Click **Edit User**

Adding New Users

To add new users:

1. Using the WebUI, navigate to: *Maintenance > Passwords*
2. Use the following section:

The screenshot shows a web form titled "ADD NEW USER". It contains three input fields: "Username", "Password *", and "Re-enter Password *". A green button labeled "Add New User" is positioned at the bottom right of the form area.

3. Enter the required *Username & Password* and click **Add New User**

Note:

Passwords cannot contain the double quotation mark (").

4. 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>
```

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 an SSH session. When run on the master of a correctly configured clustered pair, both appliance's will be updated. The script locks down the following:

- the password for the 'loadbalancer' Web User Interface account

- the password for the Linux 'root' account
- which subnet/host is permitted to access 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
```

The following image illustrates how the script works for a single appliance:

```
[root@lbmaster ~]# lbsecure

Loadbalancer.org security lock-down

This script enhances the security of a single or high-availability pair of
load balancers.

You will be asked to provide new passwords for the web interface and the
console root account, plus an IP subnet that should be allowed remote access
to the load balancer's web interface and ssh console.

Please enter a new password for the web interface 'loadbalancer' user. The
password will not be displayed as you type.
  New web interface password:
  Confirm password:

Please enter a new password for the console 'root' user. The password will not
be displayed as you type.
This password will also be used for the console 'setup' user.
  New console password:
  Confirm password:

Please enter an IP subnet that should be allowed remote access to the web
interface and ssh console.
Note that any host outside of this subnet will immediately lose access to the
load balancer. If you are running this script remotely, that includes the
current console.
  Administration subnet: 192.168.64.0/18

Working...
  Generating new SSH keys...
  SSH keys replaced.

  Setting web interface password...
  Setting console root password on local machine...
  Setting console 'setup' password on local machine...
  Passwords set.

  Setting up firewall...
  Firewall enabled.

Security enhancement complete.
```

Once the script has finished, the “**Security enhancement complete**” message is displayed as shown above.

Note:

If `lbsecure` is run on the master of a correctly configured HA pair, the passwords, firewall rules and SSH keys will also be updated on the slave appliance.

You should run `lbsecure` **AFTER** configuring the HA pair to ensure the correct HA related ports are configured in the firewall rules.

To reverse the action of `lbsecure`, the command `ibinsecure` can be used. For a clustered pair, run `ibinsecure` on both master and slave to completely reverse the configuration applied by running `lbsecure`.

Web User Interface HTTPS Configuration

By default the WebUI is accessible on HTTP port 9080 and HTTPS port 9443. This menu option enables the HTTPS configuration to be customized.

To view/manage WebUI HTTPS options:

1. Using the WebUI, navigate to: *Maintenance > Web Interface Settings*

APPLIANCE SECURITY OPTIONS.			
Web User Interface via HTTPS only	<input checked="" type="checkbox"/>		?
HTTPS Port for Web User Interface	<input type="text" value="9443"/>		?
Web Interface SSL Certificate	No SSL Certificates Installed! Defaulting to Self Signed.		?
Ciphers to use	<input type="text" value="ECDHE-RSA-RC4-SHA:ECI"/>		?
			<input type="button" value="Update"/>

2. To allow access to the WebUI via HTTPS only, check (tick) the option *Web User Interface via HTTPS only*
3. Define the HTTPS port to use for the WebUI in the *HTTPS Port for Web User Interface* field, the default is 9443
4. Select the required SSL certificate. Certificates can be created/uploaded using the WebUI menu option: *Cluster Configuration > SSL Certificate*. If no certificates are available, the default self-signed certificate will be used
5. Specify the required cipher, the default is:

```
ECDHE-RSA-RC4-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA:AES256-SHA:HIGH:
RC4:!MD5:!aNULL:!EDH
```

SSH Keys

This menu option enables SSH keys to be managed.

Note:

Normally this menu option will not be used because the keys are managed by the Loadbalancer.org appliance and under normal circumstances do not require user intervention.

To view/manage SSH keys:

- Using the WebUI, navigate to: *Local Configuration > SSH Keys*

SECURITY

SSH Keys | SSH Authentication

Host Keys ?

Create new key pair | Upload key pair

Type	Length (bits)	Date		
DSA	1024	2015-07-14 09:08	Delete	Download public key
RSA	2048	2015-07-14 09:08	Delete	Download public key

User Keys ?

Create new key pair | Upload key pair

Username	Type	Length (bits)	Date		
root	RSA	2048	2015-07-14 09:08	Delete	Download public key

Synchronise keys with peer

- The first tab (SSH Keys) enables the following keys to be viewed & managed:
 - Host Keys** - the host identification key(s) of the local host
 - User Keys** - the public key(s) of the user presented to remote hosts
- The second tab (SSH Authentication) enables the following keys to be viewed & managed:
 - Host Keys (known_hosts)** - the known key(s) of hosts that have been previously connected to or have been pre-configured. In an HA pair you will see the peer appliance keys.
 - User Keys (authorized_keys)** - the public key(s) of remote hosts that can log in as the specified user. In an HA pair you will see the peer appliance keys.

Full Root Access

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

APPLIANCE CONFIGURATION FILES & LOCATIONS

Network 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
Heartbeat configuration:	/etc/ha.d/ha.cf

Note:

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

Chapter 6 – Configuring Load Balanced Services

Layer 4 Services

THE BASICS

Layer 4 services are based on LVS (*Linux Virtual Server*). 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 backend server for both HTTP & HTTPS traffic and also prevent the client having to renegotiate the SSL connection.

Note:

It's not possible to configure a VIP on the same IP address as any of the network interfaces. this ensures that services can 'float' (move) between master and slave appliances.

CREATING VIRTUAL SERVICES (VIPS)

Each Virtual Service can have an unlimited number of Real Servers (except the Enterprise R20 which is limited to 5 x VIPs each with up to 4 RIPs). Typically you'll need one Virtual Service for each distinct cluster (group of load balanced servers). Multiple ports can also be specified.

To add a new layer 4 VIP:

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Virtual Services*
2. Click **Add a new Virtual Service**

Label	VIP Name	?
Virtual Service	IP Address	10.0.0.20
	Ports	80
Protocol	TCP	?
Forwarding Method	Direct Routing	?

3. Enter an appropriate *Label* (name) for the new Virtual Service
4. Enter the required IP address in the *Virtual Service IP address* field
5. Enter the required port(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** (WebUI – HTTP), **9443** (WebUI – HTTPS), **7777** (HAProxy statistics page), **7778** (HAProxy persistence table replication) and **9081** (nginx fallback page).

6. Set the *Protocol* as required:
7. **TCP** – Transmission Control Protocol is the default and most common option
8. **UDP** – User Datagram Protocol – used for DNS, SIP, etc.
9. **One Packet Scheduling** - used for UDP SIP connections
10. **Firewall Marks** – For use when traffic has been tagged in the firewall script using the MARK target
11. Select the required *Forwarding Method*:

Direct Routing (DR) - This is the default one-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 two-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.

12. Click **Update**
13. Now proceed to define the RIPs (Real Servers) as detailed on page [77](#)

MODIFYING A VIRTUAL SERVICE

When first adding a Virtual Service, only certain settings 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.
Persistent		<p>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.</p> <p>Note: If <i>Protocol</i> for the Virtual Service is set to 'One Packet Scheduling', persistence will be based on SIP Call-ID.</p> <p>Note: If your real servers cannot keep session state persistence themselves, then you will obtain performance but not reliability benefits from a load balancer.</p>
	Timeout	<p>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.</p>
	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>
Health Checks		Specify the type of health check to be performed on the Real Servers.
	Check Type	<p>Specify the type of health check to be performed on the Real Servers.</p> <p>Negotiate – 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>Ping Server – Use a simple ICMP ping to perform health checks.</p> <p>External script – Use a custom file for the health check. Specify the script path in the Check Command field.</p> <p>No checks, always off – all Real Servers are marked</p>

		<p>offline.</p> <p>No checks, always on – all Real Servers are marked online.</p> <p>5 Connects, 1 Negotiate – Repeating pattern of 5 Connect checks followed by 1 Negotiate check.</p> <p>10 Connects, 1 Negotiate – Repeating pattern of 10 Connect checks followed by 1 Negotiate check.</p> <p>Note: For full details on the various options for each health check type, please refer to Chapter 8 – Real Server Health Monitoring & Control (Layer 4 services) on page 174.</p>
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>
Fallback Server		<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).</p> <p>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.</p> <p>For example you have one server in the cluster and one fallback they will act as a master/slave pair.</p>
	IP Address	Set the fallback server IP Address.
	Port	Set the fallback server port, for DR mode leave this blank as it must be the same as the VIP.
	MASQ Fallback	Masquerade fallback. When enables, this enables the fallback server to be set as a Layer 7 Virtual Service. This is especially useful in WAN/DR site

		environments.
Email Alert Destination Address		Destination email address for server health-check notifications.

Note:

For more details on configuring health-checks please refer to chapter 8 starting on page [173](#).

Note:

If you require a custom gateway for a particular VIP, this can be achieved using Policy Based Routing. To configure a VIP to return it's traffic via a custom gateway rather than via the default gateway, create a simple configuration file in `/etc/pbr.d/` with a `.conf` extension for that VIP, e.g. `/etc/pbr.d/VIP1.conf`. The file must contain the desired gateway and the VIP address using the following format:

```
GW=172.16.200.1
VIP=172.16.200.37
```

Optional options:

ROUTES= Adding this variable with the option "local" as below forces the script to only copy the link local route for the VIP specified, not all link local routes.

FROM= Allows you to provide additional FROM rules, either a single address/subnet or multiple addresses/subnets which would need to be space separated.

Examples:

```
ROUTES="local"
FROM="10.10.10.10/32 12.0.0.0/8"
```

Please contact our support team: support@loadbalancer.org if you need any assistance.

CREATING REAL SERVERS (RIPS)

You can add an unlimited number of Real Servers to each Virtual Service (except the Enterprise R20 which is limited to 5 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:

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Real Servers*
2. 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="100"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?

3. Enter an appropriate *Label* (name) for the new Real Server
4. Enter the required IP address in the *Real Server IP Address* field
5. Enter the required port in the *Real Server Port* field. This only applies to NAT mode, in DR mode port redirection is not possible so by default the port is the same as defined in the VIP
6. Specify the required *Weight*, this is an integer specifying the capacity of a server relative to the others in the pool, valid values are 0 to 65535, the default is 100. The higher the value, the more connections the server will receive. If the weight is set to 0, the server will effectively be placed in drain mode
7. 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
8. 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
9. 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 active appliance (typically the master) swaps over to the passive appliance (typically the slave appliance) then you can start the synchronization daemons on each load balancer to replicate the data in real time as detailed below.

First login to the master appliance using SSH or at 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 at the console, then as root run the following commands:

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

To ensure that these sync daemons are started on each reboot put these commands in the rc.firewall. This can be done using the WebUI menu option: *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 viewing the output from:

```
ipvsadm -Lc
```

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

When run on the active device the output will show all connections including those in state 'NONE' i.e. the persistence entries. When run on the passive device, the output will only include connections in state 'NONE' since only these are being replicated.

To stop the replication, run the following commands on both appliance's:

```
ipvsadm --stop-daemon master
ipvsadm --stop-daemon backup
```

Note:

Setting this option can generate a high level traffic between the master and slave appliances.

Note:

Once configured, you'll see multicast traffic from the active appliance on IP address 224.0.0.81 , port 8848.

DR MODE CONSIDERATIONS

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. To enable DR mode to operate:

- Each Real Server must be configured to accept packets destined for both the VIP address **and** the Real Servers IP address (RIP). This is because in DR mode the destination address of load balanced packets is the VIP address, whilst for other traffic such as health-checks, administration traffic etc. it's the Real Server's own IP address (the RIP). The service/process (e.g. IIS) must also respond to both addresses.
- Each Real Server must be configured so that it does not respond to ARP requests for the VIP address – only the load balancer should do this.

Configuring the Real Servers in this way is referred to as '**Solving the ARP problem**'. The steps required depend on the OS used as detailed in the following sections.

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.

LAYER 4 CURRENT CONNECTIONS				
Check Status				
IPVS connection entries				
pro	expire	state	source	virtual destination
TCP	00:26	SYN_RECV	192.168.64.7:20415	192.168.111.232:80 192.168.110.240:80
TCP	00:26	SYN_RECV	192.168.64.7:20414	192.168.111.232:80 192.168.110.240:80
TCP	04:18	NONE	192.168.64.7:0	192.168.111.232:80 192.168.110.240:80

SOLVING THE ARP PROBLEM 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.

Note:

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 1-4 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: re-configure DAD on the Real Servers (this step can be skipped for IPv4 Virtual Services)

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

Step 3: apply these settings

Either reboot the Real Server or run the following command to apply these settings:

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

Step 4: 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
```

Note:

Steps 1, 2 & 3 can be replaced by writing directly to the required files using the following commands, this is 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
echo 0 > /proc/sys/net/ipv6/conf/lo/dad_transmits
echo 0 > /proc/sys/net/ipv6/conf/lo/accept_dad
```

SOLVING THE ARP PROBLEM 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(s).

SOLVING THE ARP PROBLEM 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(s).

Note:

Don't forget that the service on the Real Servers needs to listen on both the RIP address and VIP address as mentioned previously.

Note:

Failure to correctly configure the Real Servers to handle the ARP problem is the most common mistake in DR mode configurations.

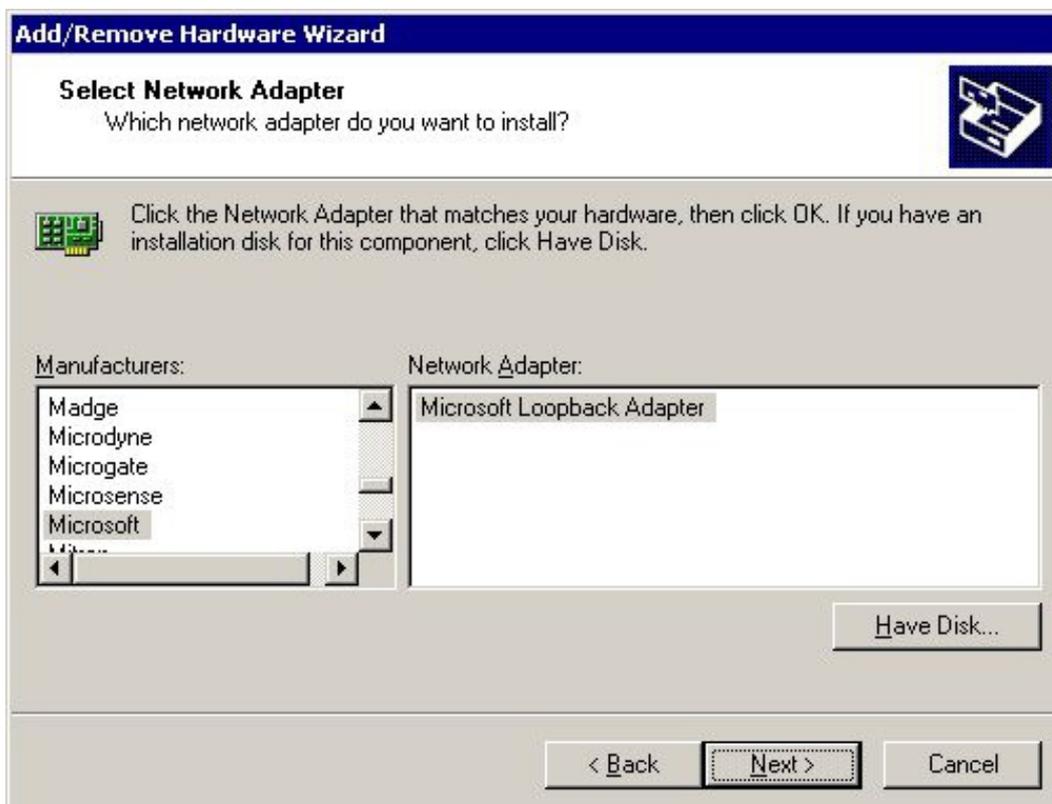
SOLVING THE ARP PROBLEM FOR WINDOWS SERVERS

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

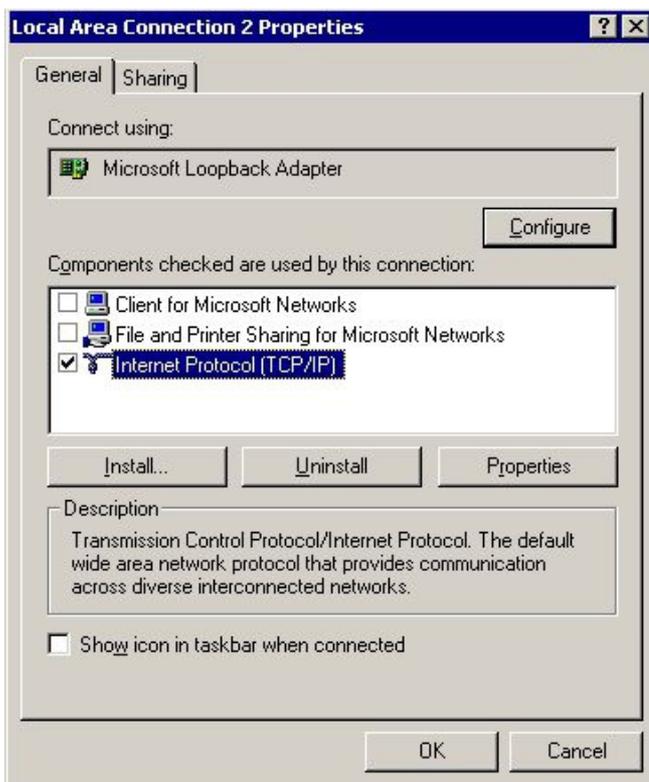
1. Open the Control Panel and double-click **Add/Remove Hardware**
2. Once the Hardware Wizard opens, click **Next**
3. Select **Add/Troubleshoot a device**, click **Next**
4. Once the device list appears, select **Add a new device** at the top of the list, click **Next**
5. Select **No, I want to select 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:



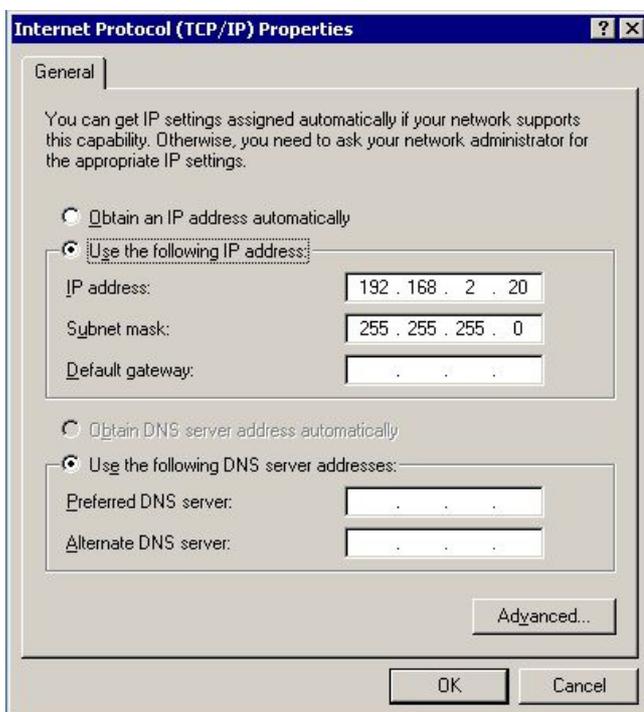
8. Click **Next** to start the installation, when complete click **Finish**

Step 2: Configure the Loopback Adapter

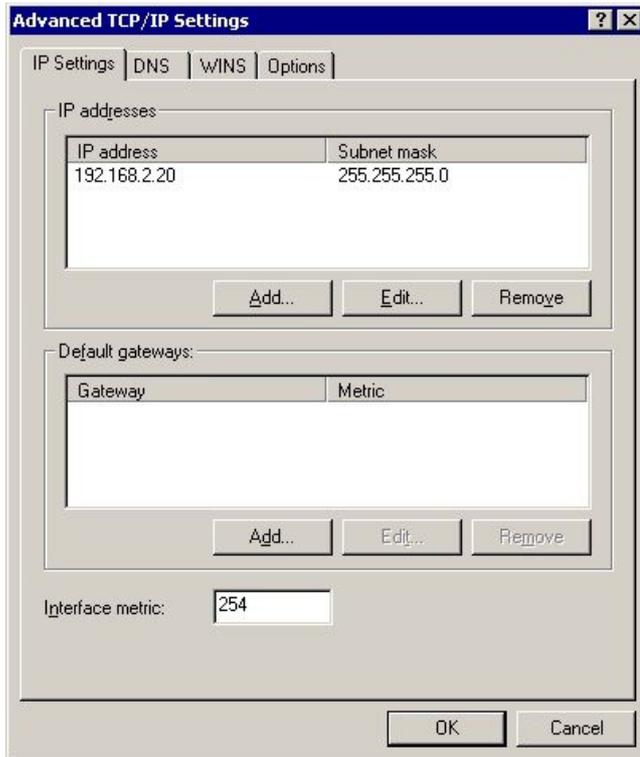
1. Open the Control Panel and double-click **Network and Dial-up Connections**
2. Right-click the new Loopback Adapter and select **Properties**
3. Un-check all items except **Internet Protocol (TCP/IP)** as shown below:



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

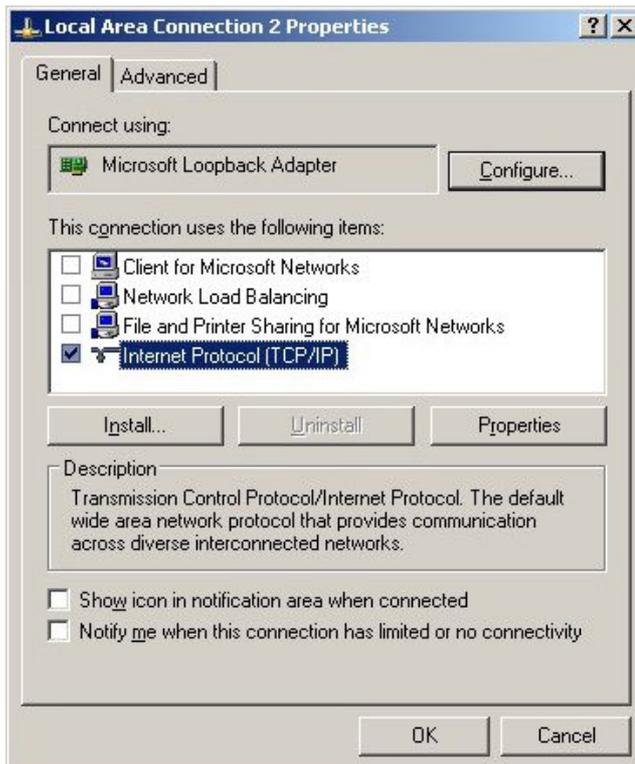
1. Open the Control Panel and double-click **Add Hardware**
2. Once the Hardware Wizard opens, click **Next**
3. Select **Yes, I have already connected the hardware**, click **Next**
4. Scroll to the bottom of the list, select **Add a new hardware device**, click **Next**
5. Select **Install the hardware 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:



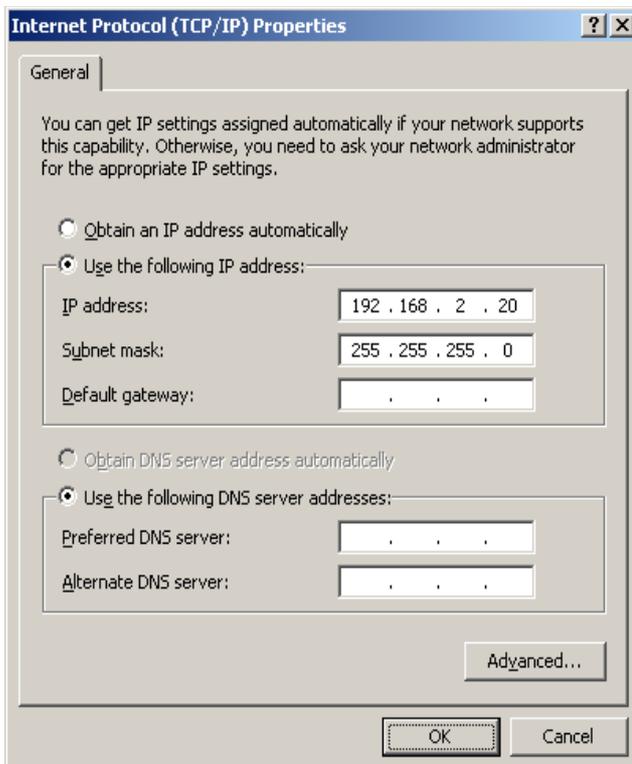
8. Click **Next** to start the installation, when complete click **Finish**

Step 2: Configure the Loopback Adapter

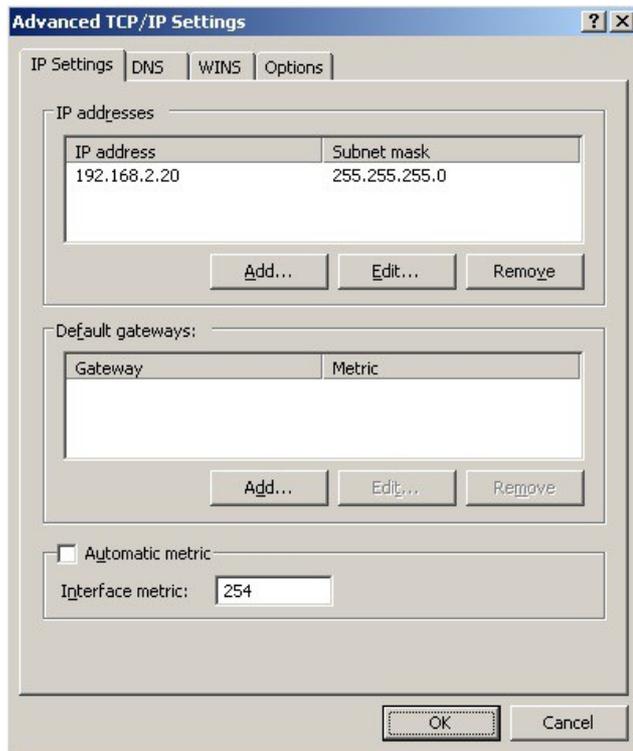
1. Open the Control Panel and double-click **Network Connections**
2. Right-click the new Loopback Adapter and select **Properties**
3. Un-check all items except **Internet Protocol (TCP/IP)** as shown below:



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:



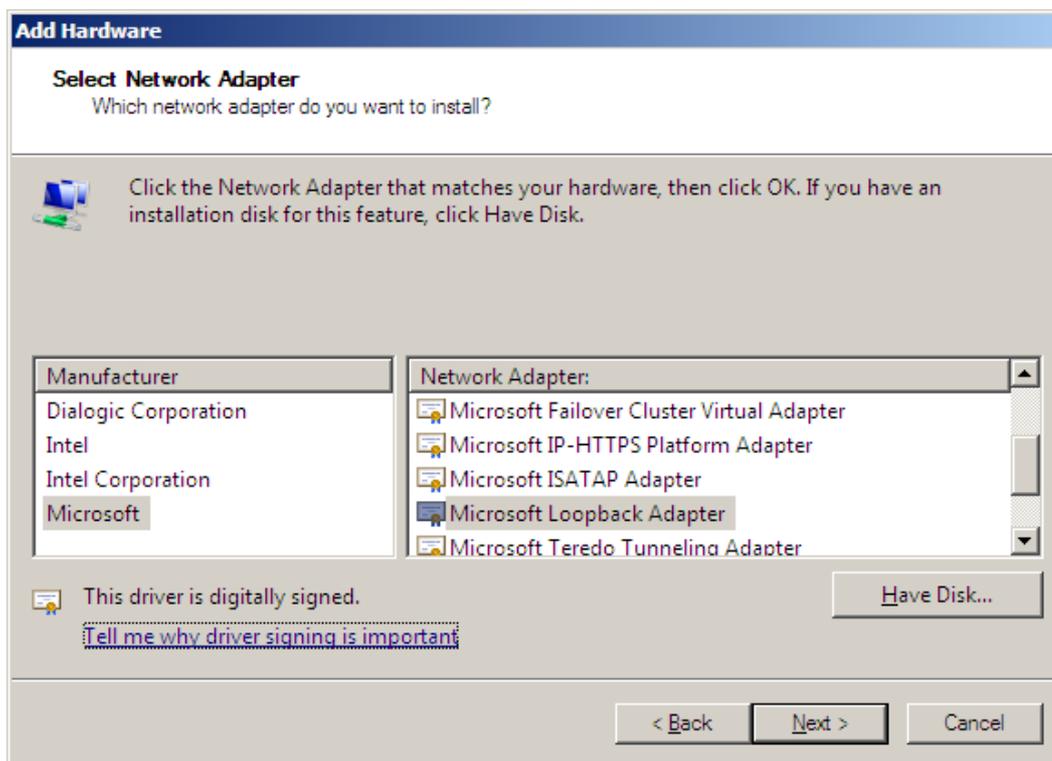
6. Click **OK** on Advanced Settings & TCP/IP Properties, then click **Close** on Connection Properties to save and apply the new settings
7. Now repeat the above process for all other Windows 2003 Real Servers

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

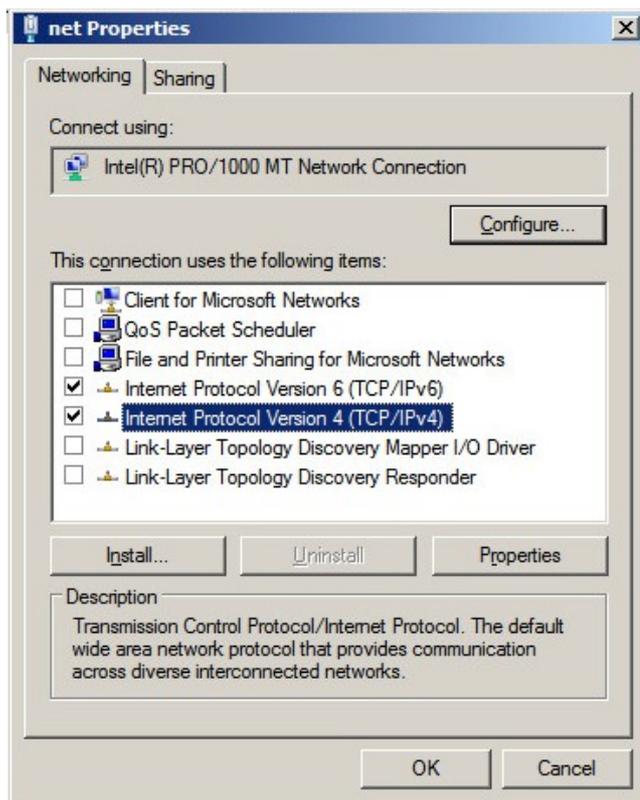
1. Click **Start**, then run **hdwwiz** to start the Hardware Installation Wizard
2. When the Wizard has started, click **Next**
3. Select **Install the hardware that I manually select from a list (Advanced)**, click **Next**
4. Select **Network adapters**, click **Next**
5. Select **Microsoft & Microsoft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



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

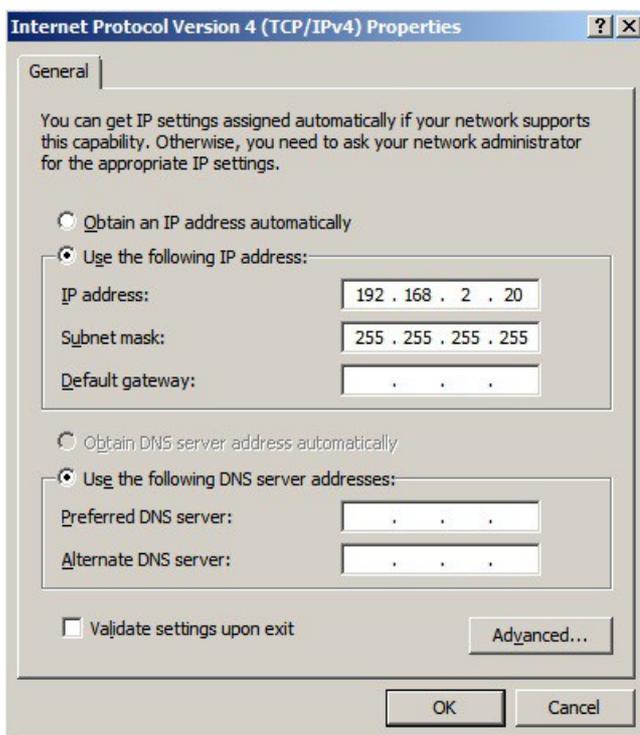
Step 2: Configure the Loopback Adapter

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)** and **Internet Protocol Version 6 (TCP/IPv6)** as shown below:

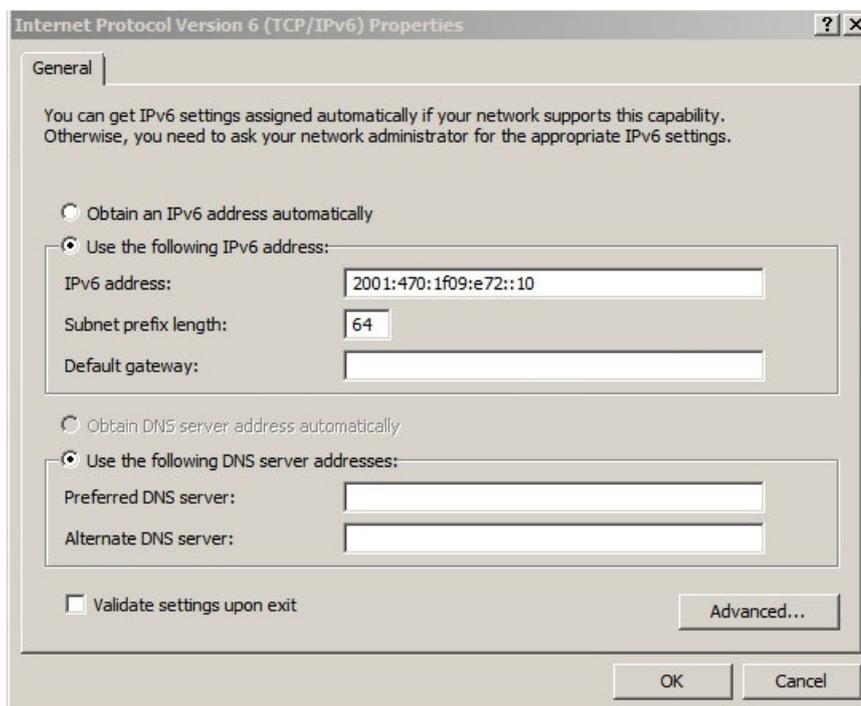
**Note:**

Leaving both checked ensures that both IPv4 and IPv6 are supported. Select one if preferred.

- If configuring IPv4 addresses 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:



- If configuring IPv6 addresses select **Internet Protocol Version (TCP/IPv6)**, click **Properties** and configure the IP address to be the same as the Virtual Service (VIP) and set the *Subnet Prefix Length* to be the same as your network setting , e.g. 2001:470:1f09:e72::15/64 as shown below:



- Click **OK**, then click **Close** to save and apply the new settings
- Now repeat the above process on the other Windows 2008 Real Servers

Note:

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 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:

For IPv4 addresses:

```
netsh interface ipv4 set interface "net" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostsends=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:

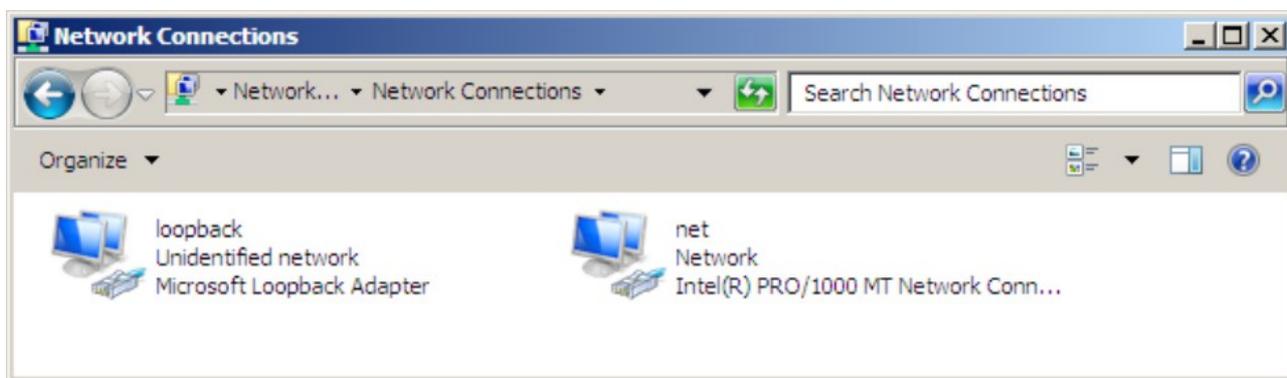
```
netsh interface ipv4 set interface "LAN" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostsends=enabled
```

For IPv6 addresses:

```
netsh interface ipv6 set interface "net" weakhostreceive=enabled
netsh interface ipv6 set interface "loopback" weakhostreceive=enabled
netsh interface ipv6 set interface "loopback" weakhostsend=enabled
netsh interface ipv6 set interface "loopback" dadtransmits=0
```

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 ipv6 set interface "LAN" weakhostreceive=enabled
netsh interface ipv6 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv6 set interface "LOOPBACK" weakhostsend=enabled
netsh interface ipv6 set interface "LOOPBACK" dadtransmits=0
```



Note:

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 appropriate netsh commands as shown in the example below:

```
Administrator: Windows PowerShell
PS C:\Users\administrator.ROBSTEST> netsh interface ipv6 set interface "net" weakhostreceive=enabled
Ok.
PS C:\Users\administrator.ROBSTEST> netsh interface ipv6 set interface "loopback" weakhostreceive=enabled
Ok.
PS C:\Users\administrator.ROBSTEST> netsh interface ipv6 set interface "loopback" weakhostsend=enabled
Ok.
PS C:\Users\administrator.ROBSTEST> netsh interface ipv6 set interface "loopback" dadtransmits=0
Ok.
PS C:\Users\administrator.ROBSTEST>
```

Note:

This shows an IPv6 example, use the IPv4 commands if you're using IPv4 addresses.

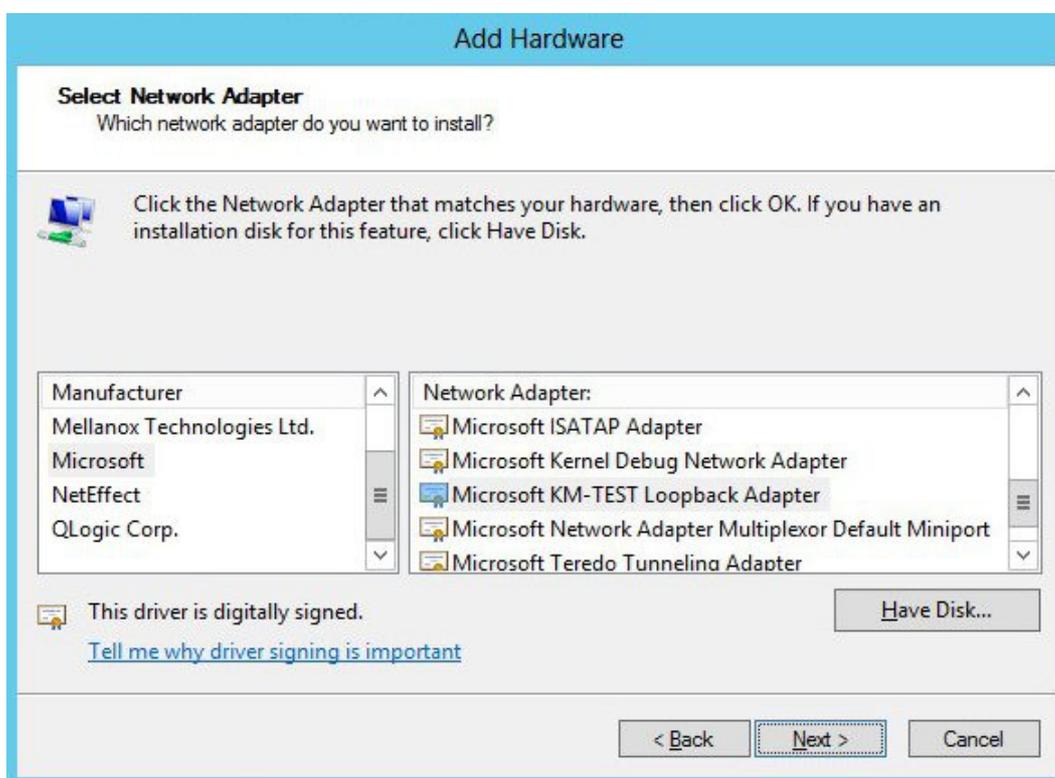
2. Now repeat these 4 commands on the other Windows 2008 Real Servers

Windows Server 2012 & 2016

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

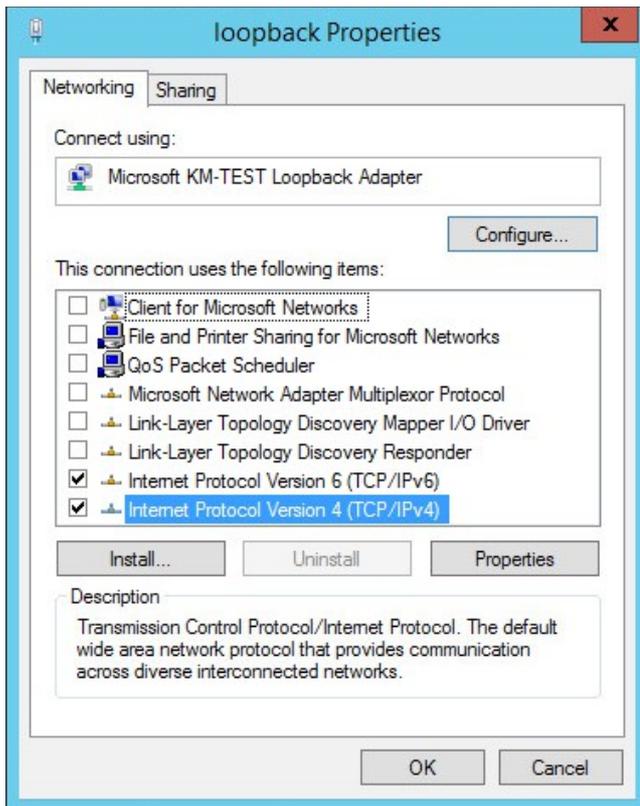
1. Click **Start**, then run **hdwwiz** to start the Hardware Installation Wizard
2. When the Wizard has started, click **Next**
3. Select **Install the hardware that I manually select from a list (Advanced)**, click **Next**
4. Select **Network adapters**, click **Next**
5. Select **Microsoft & Microsoft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



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

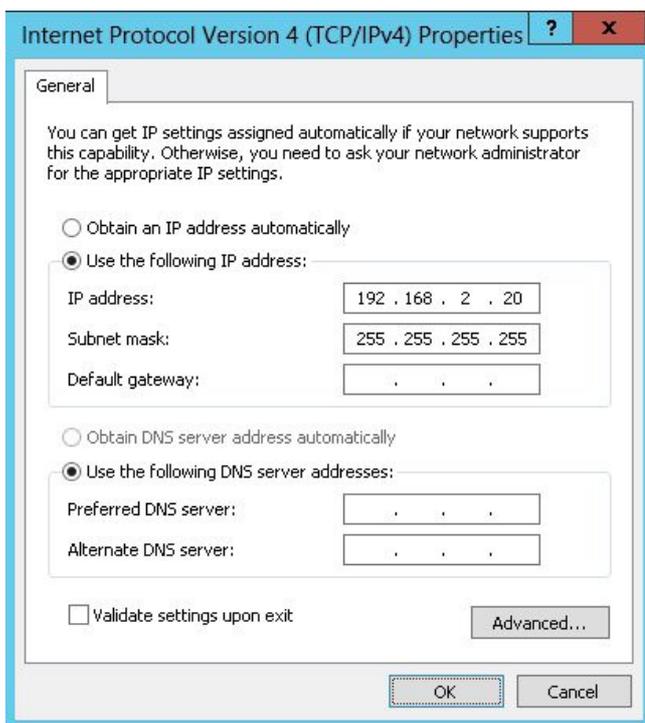
Step 2: Configure the Loopback Adapter

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**
4. Un-check all items except **Internet Protocol Version 4 (TCP/IPv4)** and **Internet Protocol Version 6 (TCP/IPv6)** as shown below:

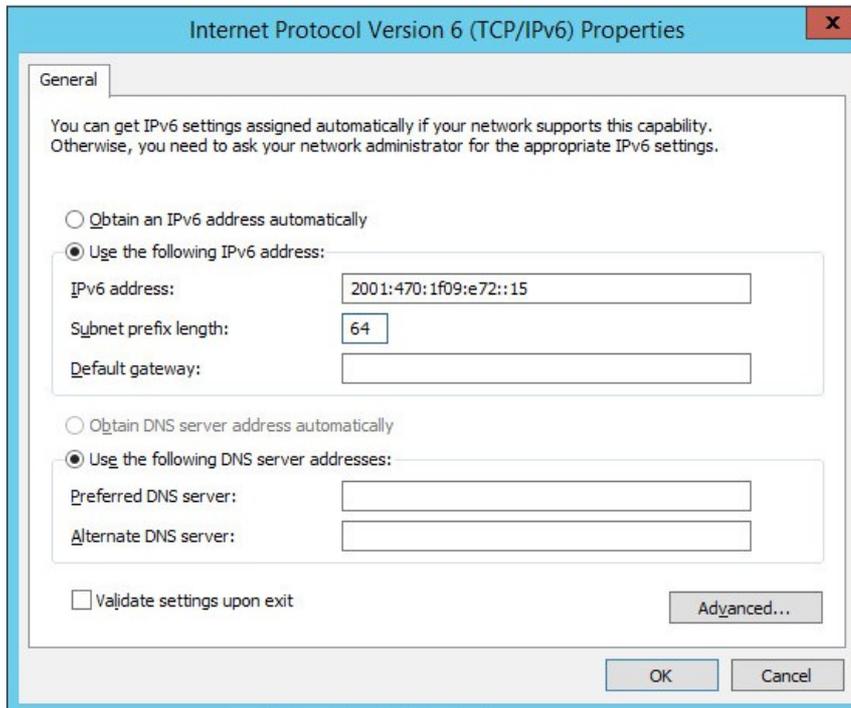
**Note:**

Leaving both checked ensures that both IPv4 and IPv6 are supported. Select one if preferred.

5. If configuring IPv4 addresses 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:



- If configuring IPv6 addresses select **Internet Protocol Version (TCP/IPv6)**, click **Properties** and configure the IP address to be the same as the Virtual Service (VIP) and set the *Subnet Prefix Length* to be the same as your network setting , e.g. 2001:470:1f09:e72::15/64 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 2012/2016 Real Servers

Note:

For Windows 2012/2016, 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 Windows 2012/2016 is running in the correct mode to be able to respond to the VIP, the following commands must be run on each Real Server:

For IPv4 addresses:

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

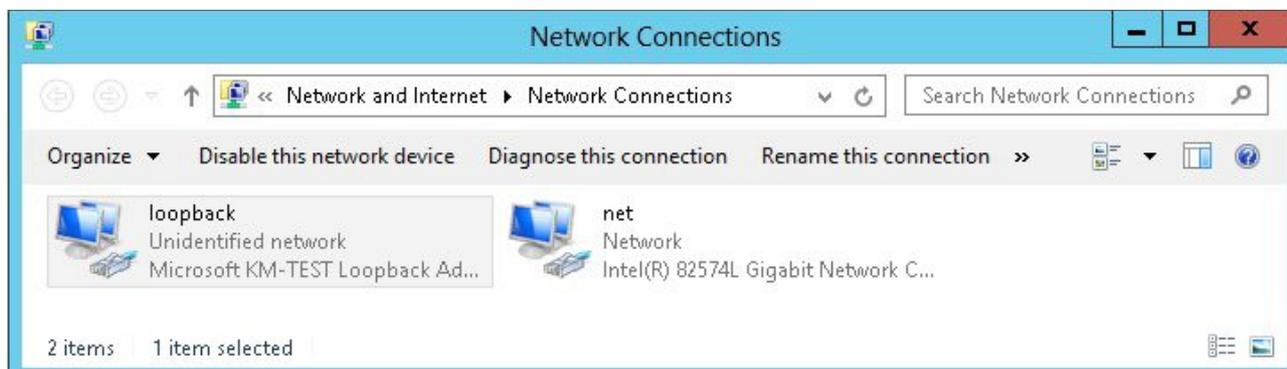
```
netsh interface ipv4 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostsend=enabled
```

For IPv6 addresses:

```
netsh interface ipv6 set interface "net" weakhostreceive=enabled
netsh interface ipv6 set interface "loopback" weakhostreceive=enabled
netsh interface ipv6 set interface "loopback" weakhostsend=enabled
netsh interface ipv6 set interface "loopback" dadtransmits=0
```

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 ipv6 set interface "LAN" weakhostreceive=enabled
netsh interface ipv6 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv6 set interface "LOOPBACK" weakhostsend=enabled
netsh interface ipv6 set interface "LOOPBACK" dadtransmits=0
```



Note:

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 appropriate netsh commands as shown in the example below:

```
Administrator: Windows PowerShell
PS C:\Users\administrator.ROBTEST> netsh interface ipv6 set interface "net" weakhostreceive=enabled
Ok.
PS C:\Users\administrator.ROBTEST> netsh interface ipv6 set interface "loopback" weakhostreceive=enabled
Ok.
PS C:\Users\administrator.ROBTEST> netsh interface ipv6 set interface "loopback" weakhostsend=enabled
Ok.
PS C:\Users\administrator.ROBTEST> netsh interface ipv6 set interface "loopback" dadtransmits=0
Ok.
PS C:\Users\administrator.ROBTEST>
```

Note:

This shows an IPv6 example, use the IPv4 commands if you're using IPv4 addresses.

2. Now repeat these 4 commands on the other Windows 2012 Real Servers

For Windows 2012/2016 you can also use the latest PowerShell Cmdlets:

The following example configures both IPv4 and IPv6 at the same time:

```
Set-NetIPInterface -InterfaceAlias loopback -WeakHostRecieve enabled -WeakHostSend enabled
-DadTransmits 0
```

```
Set-NetIPInterface -InterfaceAlias net -WeakHostRecieve enabled
```

To configure just IPv4:

```
Set-NetIPInterface -InterfaceAlias loopback -WeakHostRecieve enabled -WeakHostSend enabled
-DadTransmits 0 -AddressFamily IPv4
```

```
Set-NetIPInterface -InterfaceAlias net -WeakHostRecieve enabled -AddressFamily IPv4
```

To configure just IPv6:

```
Set-NetIPInterface -InterfaceAlias loopback -WeakHostRecieve enabled -WeakHostSend enabled
-DadTransmits 0 -AddressFamily IPv6
```

```
Set-NetIPInterface -InterfaceAlias net -WeakHostRecieve enabled -AddressFamily IPv6
```

Verifying Strong/Weak Host Settings for Windows 2008/2012/2016

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**

```
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>
```

Note:

For IPv6, simply replace 'ipv4' with 'ipv6' in the above commands.

For Windows 2012/2016 you can also use the following PowerShell Cmdlets to verify the settings:

To view both IPv4 and IPv6:

```
Get-NetIPInterface -InterfaceAlias loopback | FL
```

for IPv4 only:

```
Get-NetIPInterface -InterfaceAlias loopback -AddressFamily IPv4 | FL
```

for IPv6 only:

```
Get-NetIPInterface -InterfaceAlias loopback -AddressFamily IPv6 | FL
```

Note:

For Windows server 2008/2012/2016, 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.

Note:

Failure to correctly configure the Real Servers to handle the ARP problem is the most common problem in DR configurations.

CONFIGURING YOUR APPLICATION TO RESPOND TO BOTH THE RIP AND VIP

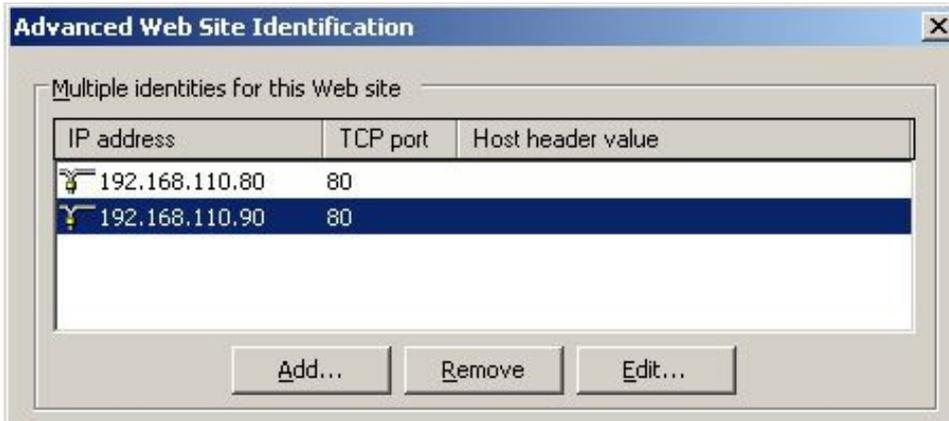
For DR & TUN modes, it's also important to make sure that your application (IIS in this case) responds to both the VIP and RIP.

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'.



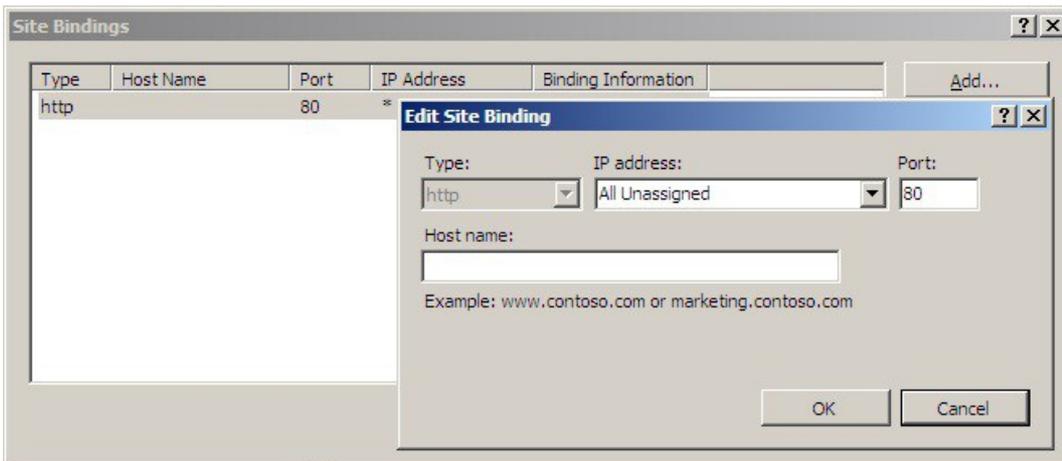
If the default configuration is left, no further IIS configuration is required. If you do change the IP address in the bindings from 'All Unassigned' to a specific IP address, then you need to make sure that you also add a binding for the Virtual Service IP address (VIP) as shown in the example below:

**Note:**

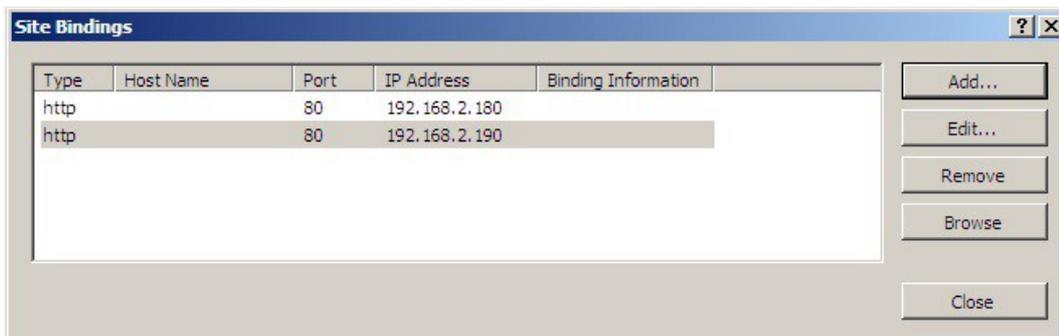
These examples illustrate how IIS must be configured to ensure that it's listening on both the RIP and VIP address. It's important to remember that this applies equally to all applications when running in DR mode.

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:

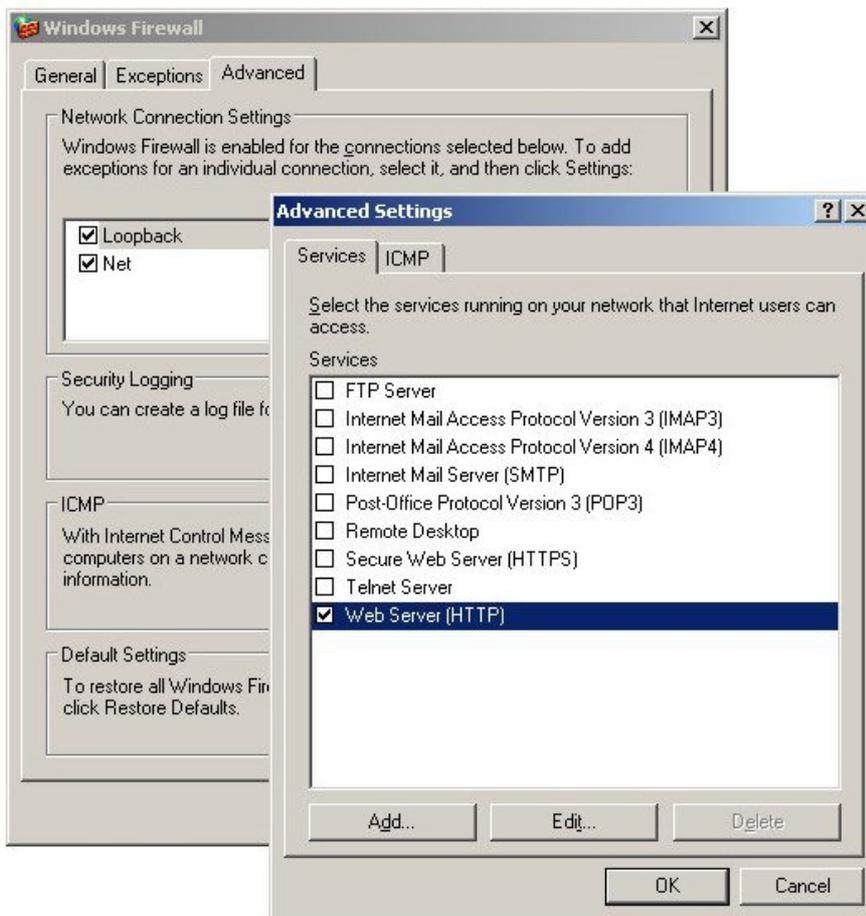


Note:

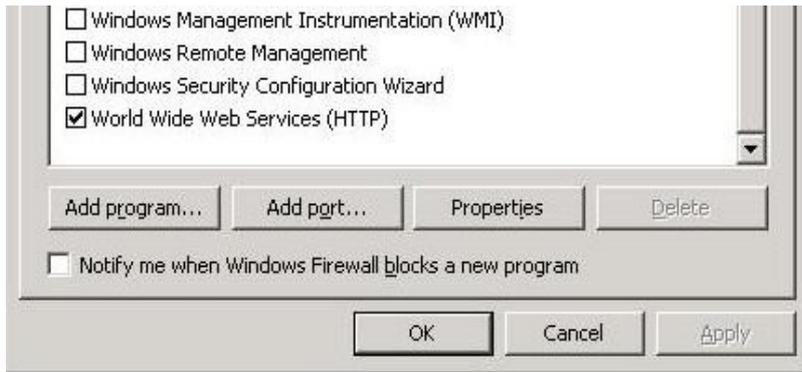
These examples illustrates how IIS must be configured to ensure that its listening on both the RIP and VIP address. It's important to remember that this applies equally to all applications when running in DR mode.

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:



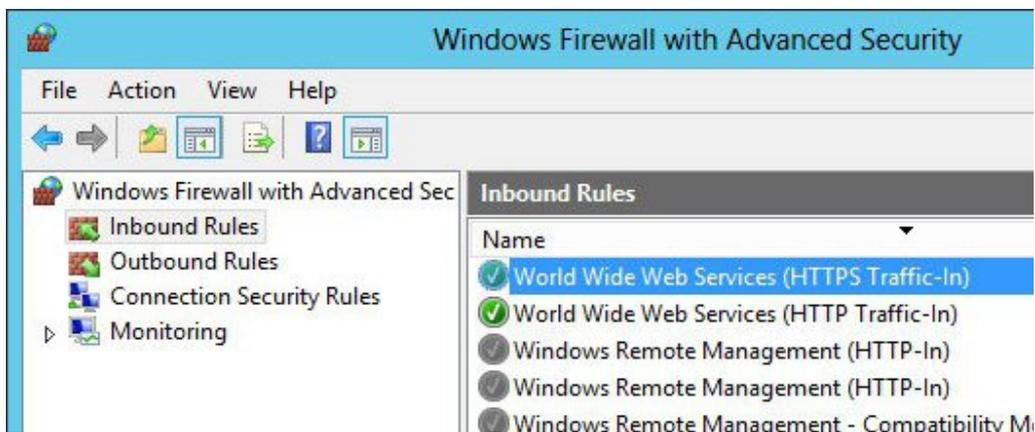
Windows 2008 R2 Firewall Settings

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. By default 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

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Advanced Configuration*
2. Change Auto-NAT from **off** to the external interface being used – typically **eth1**
3. 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 WebUI menu 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
```

Note:

If Autonat is already enabled, only the DNAT rule (i.e. not the SNAT rule) will be required.

Note:

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 is located on a different subnet to the Real Servers. However, it is possible to perform NAT mode load balancing on a single subnet. Here, the VIP is brought up in the same subnet as the Real Servers. For clients located on this subnet, return traffic would normally be sent directly to the client bypassing the load balancer which would break NAT mode. To address this, the routing table on the Real Servers must be modified to force return traffic to go via the load balancer. The sections below explain how routing must be modified for Windows hosts and Linux hosts.

Route Configuration for Windows Servers

To rectify this issue for Windows servers, a route must be added to each Real Server that takes priority over the default Windows routing rules. This is a simple case of deleting the default On-link route and adding a permanent route via the load balancer using the following commands on each real server:

```
netsh interface ipv4 delete route 192.168.2.0/24 "LAN"  
netsh interface ipv4 add route 192.168.2.0/24 "LAN" 192.168.2.21
```

Note:

Ensure you specify your local subnet address.
Replace 192.168.2.21 with the IP address of your load balancer.
Replace "LAN" with the name of your Interface.

Note:

After running the above commands, reboot the server and check if the updated routing rules have remained. Depending on the specific version of Windows, it may be necessary to add the commands to a startup script. This is because under certain circumstances routing rules for on-link, directly accessible addresses can get reset to defaults after a reboot.

Verify routing rules using the following command:

```
netsh interface ipv4 show route
```

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 192.168.2.0 netmask 255.255.255.0 dev eth0  
route add -net 192.168.2.0 netmask 255.255.255.0 metric 2000 dev eth0
```

Note:

Ensure you specify your local subnet address.
 Replace 192.168.2.21 with the IP address of your load balancer.

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

```
route add -net 192.168.2.0 netmask 255.255.255.0 gateway 192.168.2.21 metric 0 dev eth0
```

Note:

Replace 192.168.2.0 & 255.255.255.0 with your local subnet address.
 Replace 192.168.2.21 with the IP address of your load balancer.

Any local traffic (same subnet) is then handled by the 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 and/or multiple IP addresses 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 remain on the same Real Server.

FIREWALL MARKS – AUTO CONFIGURATION

When defining a layer 4 VIP with multiple ports, firewall marks are used automatically in the background to enable this functionality. For example, to configure an HTTP & HTTPS NAT mode Virtual Service, port 80 & 443 must be specified separated by a comma in the 'Virtual Service Ports' field as shown below:

Label	<input type="text" value="HTTP-Cluster"/>		?
Virtual Service	IP Address	<input type="text" value="192.168.115.100"/>	?
	Ports	<input type="text" value="80,443"/>	?
Protocol	<input type="text" value="TCP"/>		?
Forwarding Method	<input type="text" value="NAT"/>		?
			<input type="button" value="Cancel"/> <input type="button" value="Update"/>

This will automatically configure the load balancer for firewall marks.

Note:

Persistence will be enabled automatically.

For NAT mode VIPs, leave the *Real Server Port* field blank as shown below:

Label	<input type="text" value="IIS1"/>	?
Real Server IP Address	<input type="text" value="192.168.30.22"/>	?
Real Server Port	<input type="text"/>	?
Weight	<input type="text" value="100"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?

Packets will then be forwarded to the Real Servers on the same port as received by the VIP.

Note:

For Layer 4 DR mode VIPs, there is no Real Server Port field since port translation is not possible in this mode and packets will be forwarded to the same port as specified for the VIP.

Note:

To create an auto firewall mark VIP that listens on **all ports**, simply specify * in the ports field rather than a specific port number.

Note:

The health check port is automatically set to be the first port in the list, e.g. if ports 80 & 443 are defined for the VIP, the check port is automatically set to port 80. This can be changed if required using the *Check Port* field.

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

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Virtual Services*
2. Click **Add a new Virtual Service**

Label	<input type="text" value="Cluster-1"/>	?
Virtual Service	IP Address <input type="text" value="1"/>	?
	Ports <input type="text" value="80"/>	?
Protocol	<input type="text" value="Firewall Marks"/>	?
Forwarding Method	<input type="text" value="Direct Routing"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

3. Define the required *Label* (name) for the VIP, e.g. **Cluster-1**
4. Instead of entering an IP address, enter a numeric value, e.g. **1** – this is the numeric reference for the Firewall Mark, this reference is used in step 5 below when defining the firewall rules
5. The *Virtual Service Ports* field does not need to be set as it is not relevant in this case - the actual port(s) used are defined in the firewall script in step 5 below
6. Set *Protocol* to **Firewall Marks** – at this point the *Virtual Service Ports* field will be greyed out
7. Click **Update**

Note:

Persistence will be enabled automatically.

Step 2: Define a Health-Check Port

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Virtual Services*
2. Click **Modify** next to the new Virtual Service
3. Enter the appropriate value in the *Check Port* field
4. Click **Update**

Step 3: Add the Real Servers

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Real Servers*
2. Click **Add a new Real Server**
3. Enter the required details as shown below

Label	<input type="text" value="Server1"/>	?
Real Server IP Address	<input type="text" value="192.168.111.241"/>	?
Weight	<input type="text" value="100"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

4. Click **Update**

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

1. Using the WebUI, navigate to: *Cluster Configuration > Floating IPs*
2. Add a floating IP that corresponds to the required VIP, in this example **192.168.111.240**

New Floating IP

[Add Floating IP](#)

3. Click **Add Floating IP**

Step 5: Modify the Firewall Script

1. Using the WebUI, navigate to: *Maintenance > Firewall Script*
2. Scroll down to the Manual Firewall Marks section and add the following lines as shown below:

```
VIP1="192.168.111.240"
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 8025 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -p udp -d $VIP1 --dport 8025 -j MARK --set-mark 1
```

FIREWALL SCRIPT

```

27 ##### Manual Firewall Marks #####
28
29
30 # Example: Associate HTTP and HTTPS with Firewall Mark 1:
31 #VIP1="10.0.0.66"
32 #iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1
33 #iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 443 -j MARK --set-mark 1
34
35 # A Virtual Service may then be created in the web interface, using 1 as the
36 # service address.
37
38 #It is also possible to bind TCP and UDP protocols together with a firewall mark.
39 #VIP1="192.168.64.27"
40 #iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1
41 #iptables -t mangle -A PREROUTING -p udp -d $VIP1 --dport 300 -j MARK --set-mark 1
42
43 VIP1="192.168.111.240"
44 iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 8025 -j MARK --set-mark 1
45 iptables -t mangle -A PREROUTING -p udp -d $VIP1 --dport 8025 -j MARK --set-mark 1
46
47 ##### Packet Filtering #####
48
49 # You should always use a network perimeter firewall to lock down all
50 # external access to the load balancer except the required Virtual Services
51 # and the required services from your admin machine / network (SSH & HTTPS)
52
53 # Allow unlimited traffic on the loopback interface:
54 #iptables -A INPUT -i lo -j ACCEPT
55 #iptables -A OUTPUT -o lo -j ACCEPT
56
57
58 #Do not delete the following 2 lines.
59 echo "Firewall Activated"

```

[Update](#)

3. Click **Update**
4. For a clustered pair, make the same changes to the firewall script on the slave unit

**** The VIP is now configured and will be accessible on 192.168.111.240 , TCP & UDP port 8025 ****

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

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

Step 1: Modify the Existing Virtual Service

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Virtual Services*
2. Click **Modify** next to the relevant VIP

Label	<input type="text" value="Cluster-2"/>	?
Virtual Service	IP Address	<input type="text" value="2"/>
	Ports	<input type="text"/>
Protocol	<input type="text" value="Firewall Marks"/>	?

3. Change the IP address to the chosen 'mark' value as shown above, e.g. **2**
4. Clear the *Virtual Service Ports* field
5. change the *Protocol* field to Firewall Marks
6. Click **Update**

Step 2: Define a Health-Check Port

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Virtual Services*
2. Click **Modify** next to the new Virtual Service
3. Enter the appropriate value in the *Check Port* field
4. Click **Update**

Step 3: Modify the Firewall Script

1. Using the WebUI, navigate to: *Maintenance > Firewall Script*
2. Enter the rules to configure the Firewall Mark as shown in the example below:

```
VIP1="192.168.111.240"
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 2
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 443 -j MARK --set-mark 2
```

```

FIREWALL SCRIPT
27
28 ##### Manual Firewall Marks #####
29
30 # Example: Associate HTTP and HTTPS with Firewall Mark 1:
31 #VIP1="10.0.0.66"
32 #iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1
33 #iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 443 -j MARK --set-mark 1
34
35 # A Virtual Service may then be created in the web interface, using 1 as the
36 # service address.
37
38 #It is also possible to bind TCP and UDP protocols together with a firewall mark.
39 #VIP1="192.168.64.27"
40 #iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1
41 #iptables -t mangle -A PREROUTING -p udp -d $VIP1 --dport 300 -j MARK --set-mark 1
42
43 VIP1="192.168.111.240"
44 iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 2
45 iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 443 -j MARK --set-mark 2
46
47 ##### Packet Filtering #####
48
49 # You should always use a network perimeter firewall to lock down all
50 # external access to the load balancer except the required Virtual Services
51 # and the required services from your admin machine / network (SSH & HTTPS)
52
53 # Allow unlimited traffic on the loopback interface:
54 #iptables -A INPUT -i lo -j ACCEPT
55 #iptables -A OUTPUT -o lo -j ACCEPT
56
57
58 #Do not delete the following 2 lines.
  echo "Firewall Activated"
  
```

[Update](#)

3. Click **Update**
4. For a clustered pair, make the same changes to the firewall script on the slave unit

**** The VIP is now configured and will be accessible on 192.168.111.240 , TCP ports 80 & 443 ****

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 - ADVANCED CONFIGURATION		
Lock Ldirectord Configuration	<input type="checkbox"/>	?
Check Interval	<input type="text" value="5"/>	?
Check Timeout	<input type="text" value="3"/>	?
Negotiate Timeout	<input type="text" value="5"/>	?
Failure Count	<input type="text" value="2"/>	?
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"/>	?

[Update](#)

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 WebUI. 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 be 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 WebUI 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.

Note:

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 Alert Source Address – Specify the global source address of the email alerts. When an email alert is sent, the system will use this address as the 'From' field.

Email Alert Destination Address – Specify the global destination email alert address. This 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 balancer's own IP address.

Layer 7 supports a number of persistence methods including source IP address, HTTP cookie (both application based and inserted), Connection Broker, 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 secondary IP addresses under the relevant interface.

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 backend 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:8080 is supported

Manual configuration of layer 7 services is possible using the WebUI menu option: *Cluster Configuration > Layer 7 – Manual Configuration*

Note:

It's not possible to configure a VIP on the same IP address as any of the network interfaces. This ensures services can 'float' (move) between master and slave appliances.

CREATING VIRTUAL SERVICES (VIPS)

Each Virtual Service can have an unlimited number of Real Servers (except the Enterprise R20 which is limited to 5 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:

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Services*
2. Click **Add a new Virtual Service**

Label	VIP Name	?
Virtual Service	IP Address	10.0.0.20
	Ports	80
Layer 7 Protocol	HTTP Mode	?
Manual Configuration	<input type="checkbox"/>	?

3. Enter an appropriate *Label* (name) for the new Virtual Service
4. Enter the required IP address in the *Virtual Service IP Address* field
5. Enter the required port(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** (WebUI – HTTP), **9443** (WebUI – HTTPS), **7777** (HAProxy statistics page), **7778** (HAProxy persistence table replication) and **9081** (Nginx fallback page).

6. Select the Layer 7 protocol to be handled by this Virtual Service, either HTTP or TCP
7. **HTTP Mode** – 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.
8. **TCP Mode** – Required for non HTTP traffic such as HTTPS, RPC, RDP, FTP etc.
9. If the VIP will be configured manually, check (enable) the **Manual Configuration** check-box

Note:

Please refer to page [121](#) for more information on manually configuring layer 7 services.

10. Click **Update**

11. Now proceed to define the RIPs (Real Servers) as detailed on page [120](#)

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
HTTP Pipeline Mode	<p>Select how HAProxy should handle HTTP pipelining to client and server</p> <p>No change - Act as a transparent tunnel, allowing the client and server to negotiate pipelining.</p> <p>Close both client and server - Disable pipelining, always closing connections to both client and server using HTTP.</p> <p>Keep-alive client, close server - Allow client to negotiate pipelining, whilst closing the server connection using HTTP.</p> <p>Close client, force close server - Close the server connection at the TCP layer, as well as sending the Connection: close header. Also close the client connection using HTTP.</p>
Work around broken <i>Connection: close</i>	Work around Real Servers that do not correctly implement the HTTP <i>Connection:close</i> option. This does not take effect when HTTP pipeline mode is set to No change.
Configure Content Redirection	Enables ACL's to be configured. Please see page 117 for more details.
Configure HTTP Headers	Enables HTTP headers to be added, set or deleted. Please see page 120 for more details.
Balance Mode	The scheduler used to specify server rotation. Specify the scheduler to utilize when deciding the backend server to use for the next new connection.
Persistence Mode	<p>Select how the load balancer should track clients so as to direct each request to the same server.</p> <p>HTTP Cookie - The load balancer will set an HTTP Cookie to track each client.</p> <p>Application Cookie - Where an existing HTTP Cookie is set by the web application on the Real Servers, use this to track each client.</p> <p>SSL Session ID - Read the Session ID from the SSL connection and use this to track each client.</p> <p>MS Session Broker - Use the server-set msts RDP Cookie to track clients connecting to a Microsoft Terminal Server. The Session Broker service must be enabled on the real servers.</p>

	<p>RDP Client Cookie - Use the client-set mstshash RDP Cookie to track clients connecting to a Microsoft Terminal Server. If the cookie is missing, source IP persistence will be used instead.</p> <p>Source IP - Make sure the same source IP always hits the same server.</p> <p>HTTP Cookie and Source IP - As HTTP Cookie, falling back to Source IP if the cookie is missing from the HTTP request.</p> <p>X-Forwarded-For and Source IP – Use X-Forwarded-For, falling back to Source IP if the X-Forwarded-For header is missing from the request.</p> <p>Note: You cannot use the set X-Forwarded-For header option with this method of persistence. It will be disabled.</p> <p>None - No persistence. The allocation of clients to Real Servers will be determined solely by the Balance Mode.</p>
Persistence Options	<p>The persistence options depend on which mode is selected. The following list details all available options.</p> <p>HTTP Cookie name – set the name of the HTTP cookie</p> <p>HTTP Cookie Max Idle Duration – set the max idle time of the cookie.</p> <p>HTTP Cookie Max Life Duration – set the max lifetime of the cookie.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Persistence timeout - The time-out period before an idle connection is removed from the connection table. The source IP address will be removed from memory when it has been idle for longer than the persistence timeout. The default units are minutes.</p> <p>Persistence 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</p>

	<p>information is stored to allow a session to return to the same server within the timeout period. The default units are in KB.</p>
Feedback Method	<p>Select whether HAProxy should query each Real Server for its load level.</p> <p>Agent - The Real Server is queried every health check interval for the real server's percent CPU idle. This is used to set each Real Server's weight to a value proportional to its initial weight. For example, if the initial weight is 100 and the percentage cpu idle is 34, the weight will be set to 34. Remember lower numbers mean lower priority for traffic, when compared with other real servers in the pool.</p> <p>None - HAProxy will not modify the Real Server's weight.</p>
Fallback Server	<p>Configure fallback server settings. i.e. where to direct requests if all RIPs are down.</p> <p>IP Address – define the server's IP address</p> <p>Port - define the server's IP address</p> <p>Fallback 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.</p>
Health Checks	<p>Specify the type of health check to be performed on the real servers.</p> <p>Negotiate HTTP - Scan the page specified in <i>Request to Send</i>, and check the returned data for the <i>Response Expected</i> string.</p> <p>Negotiate HTTPS - Scan the page specified in <i>Request to Send</i>, and check the returned data for the <i>Response Expected</i> string.</p> <p>Note: If a Negotiate check is selected and Response Expected is left blank, the appliance will check the location specified in Request to Send (if blank the root will be checked) and look for a 200 OK response from the real server.</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 name & path in the <i>Check Script</i> field.</p> <p>MySQL - The check consists of sending two MySQL packets, one Client Authentication packet, and one QUIT packet, to correctly close the MySQL session.</p> <p>No checks, Always on – No health checks, all real servers are marked</p>

	<p>online.</p> <p>Note: For full details on the various options for each health check type, please refer to Chapter 8 – Real Server Health Monitoring & Control (Layer 7 services) on page 178.</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.
Timeout	Use this option to override the default client & server timeouts in the Layer 7 advanced section.
Set X-Forwarded-For Header	Instruct HAProxy to add an X-Forwarded-For (XFF) 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.
Force to HTTPS	<p>If set to 'Yes' any HTTP connections that are made on this VIP will be forced to reconnect using HTTPS.</p> <p>This will keep any entered URL. If you are terminating the SSL on the Loadbalancer you should use the same VIP address for both the SSL Termination and Layer7 configurations.</p>
HTTPS Redirect Code (available when Force to HTTPS is enabled)	<p>Indicates which type of HTTP redirection is desired. Codes 301, 302, 303, 307 and 308 are supported, with 302 used by default if no code is specified.</p> <p>301 means "Moved permanently", and a browser may cache the Location.</p> <p>302 means "Moved permanently" and means that the browser should not cache the redirection.</p> <p>303 is equivalent to 302 except that the browser will fetch the location with a GET method.</p> <p>307 is just like 302 but makes it clear that the same method must be reused.</p> <p>308 replaces 301 if the same method must be used.</p>
Accept Proxy Protocol	Enable Accept 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 'Enable Proxy Protocol' is enabled in your STunnel VIP.
Send Proxy Protocol	<p>Enable Proxy Protocol to the backend servers. This option allows the backend servers to see the client's IP address. It should only be enabled if your server supports Proxy Protocol and is configured to use it.</p> <p>Send V1: This uses the first version of the Proxy Protocol and send the headers in a human readable format</p> <p>Send V2: This is the newer version of the Protocol and sends the headers in binary</p> <p>Send V2 SSL: This is used to show the client was connected over SSL/TLS</p> <p>Send V2 SSL CN: This is the same as V2 SSL but also provides the Common Name from the client certificate if set</p>
Enable Compression	Enable gzip HTTP compression. The following MIME types will be

	compressed when this is enabled: text/html , text/plain , text/css , text/xml , text/javascript , application/javascript , application/xml
Source Address	Allows the setting of the source IP address that your backend server will see the traffic coming from. This is useful when you wish to only allow a known IP Address to access your Real Servers or need to allow access through a public gateway.
Enable Backend Encryption	Enabling this option will enable by default the use of HTTPS for all new Backend Servers. This options can then be disabled per backend server under the Real Server settings.

Note:

For more details on configuring health-checks please refer to Chapter 8 starting on page [173](#).

Note:

If you require a custom gateway for a particular VIP, this can be achieved using Policy Based Routing. To configure a VIP to return it's traffic via a custom gateway rather than via the default gateway, create a simple configuration file in `/etc/pbr.d/` with a `.conf` extension for that VIP, e.g. `/etc/pbr.d/VIP1.conf`. The file must contain the desired gateway and the VIP address using the following format:

```
GW=172.16.200.1
VIP=172.16.200.37
```

Optional options:

ROUTES= Adding this variable with the option "local" as below forces the script to only copy the link local route for the VIP specified, not all link local routes.

FROM= Allows you to provide additional FROM rules, either a single address/subnet or multiple addresses/subnets which would need to be space separated.

Examples:

```
ROUTES="local"
FROM="10.10.10.10/32 12.0.0.0/8"
```

Please contact our support team: support@loadbalancer.org if you need any assistance.

URL REWRITING / CONTENT SWITCHING (ACL'S)

The WebUI supports the ability to create ACL's which can be used to control and direct HTTP traffic based on the rules defined. This option can be accessed by clicking the **Edit ACL Rules** button when modifying a VIP.

HAProxy ACL List.

Add Rule Cancel Save

Rule Type	Boolean	URL Text/IP	Rule Action Type	Redirect Location
path_beg	Equals	/example	URL Location	http://www.example.com

Add

HT Close both client and server
 Keep-alive client, close server

- Multiple rules can be defined using the **Add** button
- Once all rules have been defined, click **Save** to save the rules, then click **Update** to update the VIP, then click **Reload HAProxy** at the top of the page to apply the new settings

In the example above, requests are redirected to the URL location **http://www.example.com** if the path begins with **/example**

e.g. if the requested URL is: **http://www.domain.com/example**
 the request is redirected to: **http://www.example.com**

Other Examples:

HAProxy ACL List.

Add Rule Cancel Save

Rule Type	Boolean	URL Text/IP	Rule Action Type	Redirect Location
hdr_host	Equals	www.domain1.com	URL Prefix	www.domain3.com

Add

ACL Rules

In the example above, requests are redirected to the URL prefix **http://www.domain3.com** if the host header value is **www.domain1.com**

e.g. if the requested URL is: **http://www.domain1.com/contract**
 the request is redirected to: **http://www.domain3.com/contract**

In the example above, requests are forwarded to the backend called **Blog** if the path begins with **/blog**

e.g. if the requested URL is: **http://www.domain1.com/blog**
the request is forwarded to the manually defined backend called Blog

Requests to **http://www.domain1.com/<other locations>** are forwarded to the Real Servers that were defined using the WebUI menu option: *Cluster Configuration > Layer 7 – Real Servers*

To define the backend, use the WebUI menu option: *Cluster Configuration > Layer 7 – Manual Configuration*

The following example shows how the backend is defined:

```
backend Blog
  mode http
  balance roundrobin
  option forwardfor
  server rip3 192.168.110.242:80 weight 1 check
  server rip4 192.168.110.243:80 weight 1 check
```

Note:

When defining ACL's that have their *Rule Action Type* set to **Backend** or **Use Server**, the relevant Backend / Real Server must exist before HAProxy can be successfully restarted.

Note:

For more details on configuring ACL's please also refer to the HAProxy online documentation available at the following URL:

<https://cbonte.github.io/haproxy-dconv/1.7/configuration.html#7>

CONFIGURING HTTP HEADERS

The appliance enables HTTP headers to be added, set and deleted as described below. This option can be accessed by clicking the **Edit HTTP Headers** button when modifying a VIP.

Action	Description
Add	Allows you to append a HTTP header whose name is controlled by the 'Header Name' input box. The value of the header is controlled by the 'Header Value' Box.
set	Does the same as add but the header is removed/replaced if it already exists.
Delete	Removes all HTTP header fields that match the header name specified in the Header Name Box.

- Multiple headers can be defined using the **Add** button
- Once all headers have been defined, click **Save** to save the headers, then click **Update** to update the VIP, then click **Reload HAProxy** at the top of the page to apply the new settings

In the example above, the 3 header config rows result in the following headers being added to the requests sent from the appliance to the Web Servers:

[HTTP_X_CLIENT_DEST_PORT] , i.e. the port that the client connected to

[HTTP_X_CLIENT_DEST] , i.e. the IP address that the client connected to

[HTTP_X_SOURCE] , i.e. the clients source IP address

CREATING REAL SERVERS (RIPS)

You can add an unlimited number of Real Servers to each Virtual Service (except the Enterprise R20 which is limited to 5 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:

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Real Servers*
2. 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"/>	?
Real Server Port	<input type="text"/>	?
Re-Encrypt to Backend	<input type="checkbox"/>	?
Weight	<input type="text" value="100"/>	?

3. Enter an appropriate *Label* (name) for the new Real Server
4. Enter the required settings in the *Real Server IP Address* field and *Real Server Port* field
5. Enable *Re-encrypt to backend* if required (see page [142](#) for more details)
6. Specify the required *Weight*, this is an integer specifying the capacity of a server relative to the others in the pool, valid values are 0 to 256, the default is 100. The higher the value, the more connections the server will receive. If the weight is set to 0, the server will effectively be placed in drain mode

Note:

The configuration options *Minimum Connections* and *Maximum Connections* are available when the Real Server is modified using **Modify** after the RIP has been created.

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 WebUI. Enabling this option will replicate persistence tables for all relevant layer 7 VIPs to the peer load balancer.

To enable persistence state table replication:

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Advanced Configuration*
2. Enable the *Persistence Table Replication*
3. Click **Update**

LAYER 7 – CUSTOM CONFIGURATIONS

Custom Layer 7 services can be configured via the WebUI.

CONFIGURING MANUAL VIRTUAL SERVICES

Step 1

Create a new layer 7 Virtual Service using the WebUI menu option: *Cluster Configuration > Layer 7 - Virtual Services* ensuring that the **Manual Configuration** check-box is ticked. Enabling this option stops the

HAProxy configuration file being written for this virtual service, leaving the user to configure via the WebUI menu option: *Cluster Configuration > Layer 7 – Manual Configuration* instead.

Step 2

Define the required layer 7 Real Servers using the WebUI menu option: *Cluster Configuration > Layer 7 – Real Servers*.

Step 3

Use the WebUI menu option: *Cluster Configuration > Layer 7 - Manual Configuration* to manually define the Virtual Service and Real Servers using the same Names, IP Addresses and Ports used in steps 1 & 2.

Note:

Make sure you use the same Names, IP Addresses and Ports in Step 3 as you did in Step 1 & 2. This is required to ensure that the system overview is able to report the VIP & RIP status correctly. If different details were used, this would not be possible.

Note:

It's also possible to define ACL rules at layer 7 using the WebUI so depending on your requirements a manual configuration may not be required. Please refer to page [117](#) for more details on configuring ACL's.

Manual Config Ex. 1 – Simple HTTP Redirect

In this example, requests that start with `/staff/` or `/staff` will be redirected to <https://login.domain.com>

```
listen VIP1
bind 192.168.2.110:80
mode http
balance leastconn
acl ACL-1 path_beg /staff/                                     ← see note 1
acl ACL-2 path_beg /staff                                     ← see note 1
redirect location https://login.domain.com if ACL-1 or ACL-2   ← 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 192.168.110.111:80 weight 1 cookie rip1 check inter 2000 rise 2 fall 3 minconn 0 maxconn 0 on-
marked-down shutdown-sessions
server rip1 192.168.110.112:80 weight 1 cookie rip1 check inter 2000 rise 2 fall 3 minconn 0 maxconn 0 on-
marked-down shutdown-sessions
```

Configuration Steps:

1. Using the WebUI menu option: *Cluster Configuration > Layer 7 – Virtual Services* create a Layer 7

VIP with the required Label (name), IP Address and Port, and ensure that the **Manual Configuration** check-box is enabled, e.g.:

Label	<input type="text" value="VIP1"/>	?	
Virtual Service	IP Address	<input type="text" value="192.168.2.110"/>	?
	Ports	<input type="text" value="80"/>	?
Layer 7 Protocol	<input type="text" value="HTTP Mode"/>	?	
Manual Configuration	<input checked="" type="checkbox"/>	?	

- Using the WebUI menu option: *Cluster Configuration > Layer 7 – Real Servers* define the associated RIPs in the normal way, e.g.:

Label	<input type="text" value="rip1"/>	?
Real Server IP Address	<input type="text" value="192.168.110.111"/>	?
Real Server Port	<input type="text" value="80"/>	?
Weight	<input type="text" value="100"/>	?

- Select the WebUI menu option: *Cluster Configuration > Layer 7 – Manual Configuration* and define the required VIP/RIP settings in the text window using the same Names, IP Addresses and Ports used in the WebUI. e.g.:

```
listen VIP1
bind 192.168.2.110:80
mode http
balance leastconn
acl ACL-1 path_beg /staff/
acl ACL-2 path_beg /staff
redirect location https://login.domain.com if ACL-1 or ACL-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 192.168.110.111:80 weight 1 cookie rip1 check inter 2000 rise 2 fall 3 minconn 0
maxconn 0 on-marked-down shutdown-sessions
```

```
server rip1 192.168.110.112:80 weight 1 cookie rip1 check inter 2000 rise 2 fall 3 minconn 0
maxconn 0 on-marked-down shutdown-sessions
```

4. Click **Update**
5. Now reload HAProxy using the **Reload HAProxy** button in the blue *Commit Changes* box at the top of the screen or by using the WebUI menu option: *Maintenance > Restart Services*

Notes:

- These lines configure 2 ACL's named **ACL-1 & ACL-2** where the criteria for a match is that the URL starts with either **/staff/** or **/staff**
- This line causes a redirect to **https://login.domain.com** to occur when either acl is matched

Manual Config Ex. 2 – Load Balancing with URL matching using ACL's

To support URL matched load balancing the structure of the HAProxy configuration file must be changed to use the front-end/backend model as shown in the example below:

```
frontend f1
  bind 192.168.2.110:80
  acl ACL-1 path_beg /test1
  acl ACL-2 path_beg /test2
  use_backend b1 if ACL-1
  use_backend b2 if ACL-2
  default_backend b2
  option httpclose

backend b1
  cookie SERVERID insert nocache indirect
  server s1 192.168.2.111:80 weight 1 cookie s1 check
  server s2 192.168.2.112:80 weight 1 cookie s2 check

backend b2
  cookie SERVERID insert nocache indirect
  server s3 192.168.2.113:80 weight 1 cookie s3 check
  server s4 192.168.2.114:80 weight 1 cookie s4 check
```

Configuration Steps:

1. Using the WebUI menu option: *Cluster Configuration > Floating IPs*, add a floating IP for the new VIP, in this example 192.168.2.110 is added to match the IP address required:

The screenshot shows a web interface titled "FLOATING IPs". Below the title is a horizontal line. Underneath, there is a label "New Floating IP" followed by a text input field containing the IP address "192.168.2.110". To the right of the input field is a green button labeled "Add Floating IP".

2. Click **Add Floating IP**
3. Select the WebUI menu option: *Cluster Configuration > Layer 7 – Manual Configuration* and define the required VIP/RIP settings in the text window. .e.g.:

```
frontend F1
bind 192.168.2.110:80
acl ACL-1 path_beg /test1
acl ACL-2 path_beg /test2
use_backend B1 if ACL-1
use_backend B2 if ACL-2
default_backend B2
option httpclose

backend B1
cookie SERVERID insert nocache indirect
server s1 192.168.2.111:80 weight 1 cookie s1 check
server s2 192.168.2.112:80 weight 1 cookie s2 check

backend B2
cookie SERVERID insert nocache indirect
server s3 192.168.2.113:80 weight 1 cookie s3 check
server s3 192.168.2.114:80 weight 1 cookie s3 check
```

4. Click **Update**
5. Now reload HAProxy using the **Reload HAProxy** button in the blue *Commit Changes* box at the top of the screen or by using the WebUI menu option: *Maintenance > Restart Services*

Notes:

- **ACL-1 & ACL-2** are the names of the ACLs
- **path_beg** matches the beginning of the path to a certain value, in this case **/test1 & /test2** and then directs requests to the appropriate backend, either backend B1 or B2

Note:

This example uses the Frontend/Backend structure to define the Layer 7 Virtual Service. When using this structure, the related Virtual Service cannot be displayed in the System Overview so there is no need to define a matching VIP in this case.

These are fairly simple examples to show the principle of using ACLs. For much more information please refer to the HAProxy manual at the following link:

<https://cbonte.github.io/haproxy-dconv/1.7/configuration.html#7>

Note:

Don't hesitate to contact support@loadbalancer.org to discuss any specific ACL or other custom

configuration requirements you may have.

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 links:

<http://www.haproxy.org/download/1.7/doc/configuration.txt>

<http://cbonte.github.io/haproxy-dconv/1.7/configuration.html>

LAYER 7 – ADVANCED CONFIGURATION

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

Lock HAProxy Configuration (Deprecated)	<input type="checkbox"/>	?
Logging	Off ▾	?
Redispatch	<input checked="" type="checkbox"/>	?
Connection Timeout	4000 ms	?
Client Timeout	42000 ms	?
Real Server Timeout	43000 ms	?
Maximum Connections	40000	?
ulimit	<input type="text"/>	?
Abort on Close	<input checked="" type="checkbox"/>	?
Transparent Proxy	<input type="checkbox"/>	?
Disable On Start	<input type="checkbox"/>	?
Interval	4000 ms	?
Rise	2 checks	?
Fall	2 checks	?
Feedback Agent Interval	2000 ms	?
HAProxy Statistics Page	Password	<input type="text"/>
	Port	7777
	Advanced Stats	<input type="checkbox"/>
	Enable SSL	<input type="checkbox"/>
Request buffer length	16384 bytes	?
Header buffer length	1024 bytes	?
Persistence Table Replication	<input type="checkbox"/>	?
	Replication port	7778
eMail Alert From	<input type="text"/>	?
eMail Alert To	<input type="text"/>	?
eMail Server Address	<input type="text"/>	?
eMail Server Port	25	?

Lock HAProxy Configuration – Prevent the WebUI 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 be disabled on the [Advanced Configuration](#) page.

Note:

This Feature is now deprecated. It's now possible to configure each virtual service as read-only. The manual configuration can then be created using the WebUI menu option: *Layer 7 - Manual Configuration*.

Logging – Set the required logging level for layer 7 services. Logs are written to /var/log/haproxy.log.

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 auto-configured internally based on other system parameters and does not need to be set here.

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. Can be used on its own or in combination with Pound Tproxy.

Note:

For more details on using TProxy, please refer to page [147](#).

Note:

Since the load balancer must be in a NAT configuration (i.e. VIPs & RIPs in different subnets and default gateway on the real servers set as an IP on the load balancer) to utilize TProxy, it's 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.

Note:

For details on how to enable X-Forwarded-For support, please refer to page [116](#). For details on how to enable X-Forwarded-For support with Apache and IIS, please refer to the following blog links:

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

IIS - <http://www.loadbalancer.org/blog/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.

Feedback Agent Interval - The time in milliseconds between each feedback agent check from HAProxy to the feedback agent.

HAProxy Statistics Page Password – Set the password used to access *Reports > Layer 7 Status*.

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

Advanced Stats - Enable/disable additional actions available on the HAProxy stats page.

Enable SSL - Once enabled the HAProxy statistics page will be forced to use a HTTPS/SSL connection. You will still need to use the login details and the port set in the other HAProxy Statistics Page settings.

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

Note:

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 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. The default port is TCP 7778.

eMail Alert From – Set the 'from address' for email alerts

eMail Alert To – Set the 'to address' for email alerts

eMail Server Address – Set the email server address as either an IP address or FQDN

eMail Server Port – Set the email server TCP port

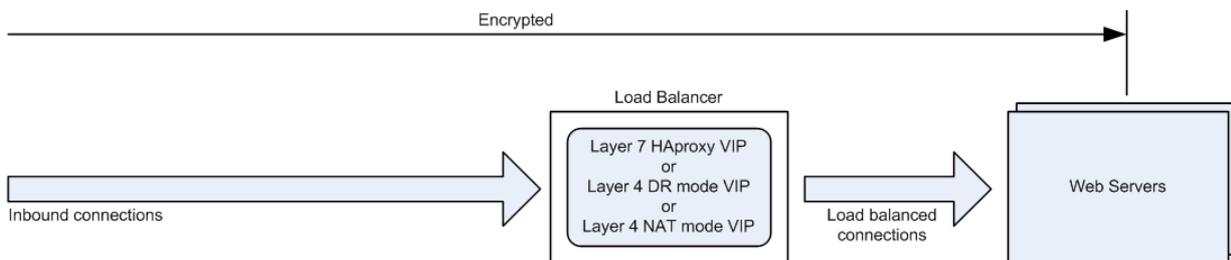
SSL Termination

CONCEPTS

SSL termination can be handled in the following ways:

1. On the Real Servers (recommended) – aka *SSL Pass-through*
2. On the load balancer – aka *SSL Offloading*
3. On the load balancer with re-encryption to the backend servers – aka *SSL Bridging*

SSL TERMINATION ON THE REAL SERVERS – AKA SSL PASS-THROUGH



Notes:

- Data is encrypted from client to server. This provides full end-to-end data encryption as shown in the diagram above
- 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
- This is our recommended solution. SSL termination on the load balancer (SSL Offload) can be very CPU intensive and in most cases, for a scalable solution, terminating SSL on the Real Servers is the best option

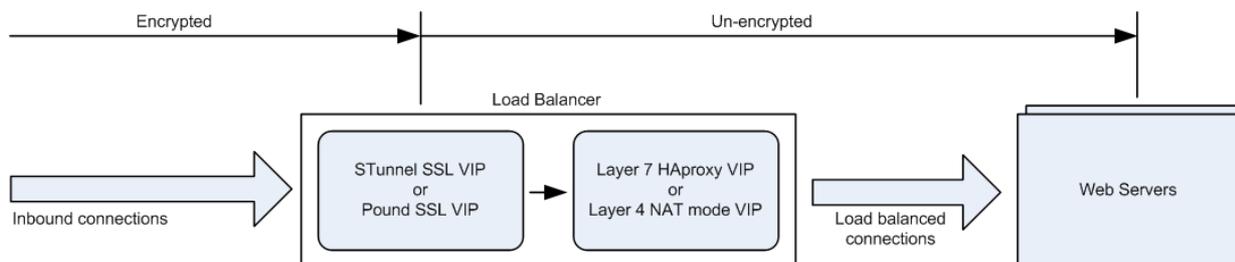
In this case SSL certificates are installed on each Real Server in the normal way. The load balancer is then configured with a VIP that listens on HTTPS port 443 and distributes inbound requests to the Real Servers on port 443 as shown in the layer 4 DR mode example below:

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 – AKA SSL OFFLOADING



Notes:

- Since SSL is terminated on the load balancer, by default, 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. If it is, use SSL bridging mode as described below
- It's possible to use HTTP cookie based persistence
- A Pound or STunnel SSL VIP is used to terminate SSL. The backend for the VIP can be either a Layer 4 NAT mode VIP or a Layer 7 HAProxy VIP. Layer 4 DR mode **cannot** be used since Pound & STunnel act as a proxy, and the real servers see requests with a source IP address of the VIP. However, since the real servers believe that they own the VIP (due to the loopback adapter configured to handle to ARP problem) they are unable to reply to Pound

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.

In this case an SSL certificate 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 an SSL Virtual Service must be defined that specifies an IP address and port to listen for inbound HTTPS connections and a backend IP address/port where to forward the corresponding un-encrypted HTTP connection.

If SSL bridging is used, the backend connections will also be encrypted. For more information please refer to page [142](#).

By default a self-signed certificate is used for the new Pound/STunnel VIP which is ideal for testing but needs to be replaced for production deployments. The self-signed certificate can be regenerated if needed using the WebUI menu option: *SSL Certificate* and clicking the **Regenerate Local SSL Certificate** button.

SERVER NAME INDICATOR (SNI)

From v8.2 SNI is supported directly in the WebUI. SNI rules can be associated with an Stunnel VIP only, not Pound. The rules can be configured using the **Modify** option for an existing STunnel VIP. For more details please refer to page [138](#).

CERTIFICATES

If you already have a full SSL certificate file in either PFX or PEM file format, this can be uploaded to the Load balancer using the certificate upload option. Alternatively, you can create a Certificate Signing Request (CSR) and send this to your Certificate Issuer to create a new certificate.

Generating a CSR on the Load Balancer

CSRs can be created on the load balancer to apply for a certificate from your CA.

To generate a CSR:

1. Using the WebUI, navigate to: *Cluster Configuration > SSL Certificates*
2. Click **Add a new SSL Certificate** & select *Create a New SSL Certificate (CSR)*

I would like to: Upload prepared PEM/PFX file ?
 Create A New SSL Certificate (CSR) ?

Label ?

Domain (CN) ?

3. Enter a suitable label (name) for the certificate
4. Enter the required Domain (CN)
5. Additional fields will appear as each one is filled in, populate these according to your requirements

I would like to: Upload prepared PEM/PFX file ?
 Create A New SSL Certificate (CSR) ?

Label ?

Domain (CN) ?

Organisation (O) ?

Organisation unit (OU) ?

City (L) ?

State or Province (ST) ?

Country code (C) ?

Email address ?

CSR Key Length ?

Signature Algorithm ?

Add Certificate

6. Once all fields are complete click **Add Certificate**

7. To view the CSR click **Modify**, then expand the Certificate Signing Request (CSR) section
8. Copy the CSR and send this to your chosen CA
9. Once received, copy/paste your signed certificate into the *Your Certificate* section
10. Intermediate and root certificates can be copied/pasted into the *Intermediate Certificate* and *Root Certificate* sections as required

Uploading a Certificate

Certificates in either PEM or PFX formats can be uploaded to the load balancer.

To upload a Certificate:

1. Using the WebUI, navigate to: *Cluster Configuration > SSL Certificates*
2. Click **Add a new SSL Certificate** & select *Upload prepared PEM/PFX file*

3. Enter a suitable *Label* (name) for the certificate
4. Browse to and select the certificate file to upload (PEM or PFX format)
5. Enter the password , if applicable
6. Click **Add Certificate**, if successful, a message similar to the following will be displayed:

Information: cert1 SSL Certificate uploaded successfully.

Note:

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

Note:

It's important to backup all your certificates. This can be done via the WebUI from *Maintenance > Backup & Restore > Download SSL Certificates*.

Exporting PFX Certificates from Windows Servers

When exporting certificates from Windows servers, make sure that *Yes, export the private key* is selected, this will enable the output format to be PFX. Also make sure that *Include all certificates in the certification path if possible* is selected.

Creating a PEM file

1. 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**, the **Private Key** and any additional **Intermediate Certificates** into the file as shown below (truncated versions are shown):

```
-----BEGIN CERTIFICATE-----
MIICsDCCAhmGAWIBAgIJAL98jHEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgN
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajiLSfE
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAABKQCCcPYkYHm8gYwlm3HyoVxjrymusOelFgZlWyuaeblrreCplo+iydRf
E89UJCG2nMW5JVBNkyHYbQTVU8MeR3ilhe2fw+qVE2pgxWYWaGm8QwTsxQKgbx
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIICsDCCAhmGAWIBAgIJAL98jHEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgN
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajiLSfE
-----END CERTIFICATE-----
```

2. Save the file

Converting between certificate formats

In some circumstances it may be required to manually convert certificates between formats. In these cases OpenSSL can be used. 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 as detailed below.

Converting PFX certificates to PEM format

Using OpenSSL on 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 OpenSSL on Windows:

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

e.g

```
openssl x509 -in cert.cer -inform DER -out 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
```

CREATING AN STUNNEL SSL VIRTUAL SERVICE

To add an STunnel SSL VIP:

1. Using the WebUI, navigate to: *Cluster Configuration > SSL Termination*
2. Click **Add a new Virtual Service**

Label	<input type="text" value="VIP Name"/>	?
SSL Certificate	No SSL Certificates Installed! Defaulting to Self Signed.	?
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" value="ECDHE-RSA-AES256-SHA384:ECDHE-"/>	?
SSL Terminator	<input type="radio"/> Pound <input checked="" type="radio"/> STunnel	?
Do not insert empty fragments	<input checked="" type="checkbox"/>	?
Delay DNS Lookups	<input checked="" type="checkbox"/>	?
Disable SSLv2 Ciphers	<input checked="" type="checkbox"/>	?
Disable SSLv3 Ciphers	<input checked="" type="checkbox"/>	?
Disable TLSv1 Ciphers	<input checked="" type="checkbox"/>	?
Client Cipher Renegotiation	<input type="checkbox"/>	?
Disable SSL Renegotiation	<input checked="" type="checkbox"/>	?
Time To Close	<input type="text" value="0"/>	?
Enable Proxy Protocol	<input type="checkbox"/>	?

3. Enter an appropriate *Label* (name) for the new Virtual Service
4. Select the required certificate from the drop down list

Note:

If you have not added any certificates at this point, a self signed cert will be used.

5. Enter the required IP address in the *Virtual Service IP address* field
6. Enter the required port in the *Virtual Service Port* field – typically 443
7. Enter the required IP address in the *Backend Virtual Service IP Address* field
8. 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.

Note:

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.

9. Enter the required port in the *backend Virtual Service Port* field
10. Define the list of accepted ciphers using the *Ciphers to use* field

By default the cipher is set to:

```
ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA:AE
S128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:AES128-SHA:DHE-RSA-
AES256-SHA256:RC4:!aNULL:!EDH:!3DES
```

This can be modified as required, or the field can be cleared (blank) to allow all available ciphers (not recommended)

11. 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.
12. Ensure *SSL Terminator* is set to **STunnel**
13. Configure *Do not Insert Empty Fragments*
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.
14. 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).
15. Configure *Disable SSLv2 Ciphers*
When ticked this option disables all SSLv2 Ciphers by using the OpenSSL 'SSL_OP_NO_SSLv2' option.
16. Configure *Disable SSLv3 Ciphers*
When ticked this option disables all SSLv3 Ciphers by using the OpenSSL 'SSL_OP_NO_SSLv3' option. Please note that entering '!SSLv3' into a Cipher list can have an adverse effect and we recommend using this option and not entering '!SSLv3'.
17. Configure *Disable TLSv1 Ciphers*
When ticked this option disables all TLSv1 Ciphers by using the OpenSSL 'SSL_OP_NO_TLSv1' option.
18. Configure *Client Cipher Renegotiation*
Sets whether the client is allowed to renegotiate the cipher order. This option needs to be disabled (unchecked) to mitigate the BEAST attack.
19. 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.
20. Configure *Time to Close*
Configure the global client response timeout in seconds. This setting should not require changing.
21. Configure *Enable Proxy Protocol*
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 page [147](#))
22. Click **Update** to create the SSL VIP

Configuring SNI (Server Name Indicator) Rules

Server Name Indication (SNI) matching allows you to send traffic to different backend Virtual Services based on the FQDN. SNI matching rules must be configured after the STunnel VIP is created.

To configure SNI rules:

1. Using the WebUI, navigate to: *Cluster Configuration > SSL Termination*
2. Click **Modify** next to the relevant STunnel VIP
3. Scroll to the bottom of the screen and set *Use Server Name Indicator (SNI)* to **Yes**, once set, the following additional options can be configured:

CURRENT SNI RULES

SNI Name	SNI to Match	SSL Certificate	Backend Virtual Service IP Address	Backend Virtual Service Port	Remove
<input type="button" value="Cancel"/> <input type="button" value="Update"/>					

ADD AN SNI RULE

Friendly Name	<input type="text" value="Friendly Name"/>	?
SNI to match	<input type="text" value="FQDN to match"/>	?
SSL Certificate	No SSL Certificates Installed! Defaulting to Self Signed.	?
Backend Virtual Service IP Address	<input type="text" value="Backend IP"/>	?
Backend Virtual Service Port	<input type="text" value="Backend Port"/>	?
<input type="button" value="Add Rule"/>		

4. Define new SNI rules and add them using the **Add Rule** button
5. Add all required SNI rules in the same way
6. Once the rules are added, they are displayed as shown in the example below:

SNI Name	SNI to Match	SSL Certificate	Backend Virtual Service IP Address	Backend Virtual Service Port	Remove
rule1	www.loadbalancer.org	cert1	192.168.110.240	80	<input type="checkbox"/>
rule2	www.lbtestdom.com	cert2	192.168.110.241	80	<input type="checkbox"/>

7. To save the changes click **Update**
8. To apply the new settings, restart STunnel using the **Restart STunnel** button at the top of the screen

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

Client Cipher 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:

1. Using the WebUI, navigate to: *Cluster Configuration > SSL Termination*
2. Click **Add a new Virtual Service**

Label	<input type="text" value="VIP Name"/>	?
SSL Certificate	Default Self Signed Certificate ▾	?
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	ECDH+AESGCM:DH+AESGCM:ECDH	?
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 checked="" type="checkbox"/>	?
Allow Client Renegotiation	No Client Renegotiation ▾	?
Disable SSLv2 Ciphers	<input checked="" type="checkbox"/>	?
Disable SSLv3 Ciphers	<input checked="" type="checkbox"/>	?
Disable TLSv1 Ciphers	<input checked="" type="checkbox"/>	?

3. Enter an appropriate *Label* (name) for the new Virtual Service
4. Select the required certificate from the drop down list

Note:

If you have not added any certificates at this point, a self signed cert will be used.

5. Enter the required IP address in the *Virtual Service IP address* field

6. Enter the required port in the *Virtual Service Port* field – typically 443
7. Enter the required IP address in the backend *Virtual Service IP address* field
8. 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.

Note:

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.

9. Enter the required port in the backend *Virtual Service Port* field
10. Define the list of accepted ciphers using the *Ciphers to use* field. By default the cipher is set to:

```
ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:AES128-SHA:DHE-RSA-AES256-SHA256!RC4:!MD5:!aNULL:!EDH:!3DES
```

This can be modified as required, or the field can be cleared (blank) to allow all available ciphers (not recommended)

11. 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.
12. Ensure *SSL Terminator* is set to **Pound**
13. Configure *Enable WebDAV Verbs*
When enabled extends which HTTP/WebDAV verbs are accepted.
14. Configure *Rewrite HTTP Redirects*
Pound to change the Location: and Content-location: headers in responses If they point to the backend 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).

15. 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.
16. Configure *Client Cipher 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.

17. 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.

18. Configure *Disable SSLv3 Ciphers*

When ticked this option disables all SSLv3 Ciphers by using the OpenSSL 'SSL_OP_NO_SSLv3' option. Please note that entering '!SSLv3' into a Cipher list can have an adverse effect and we recommend using this option and not entering '!SSLv3'.

19. 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).

20. Click **Update** to create the SSL VIP

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

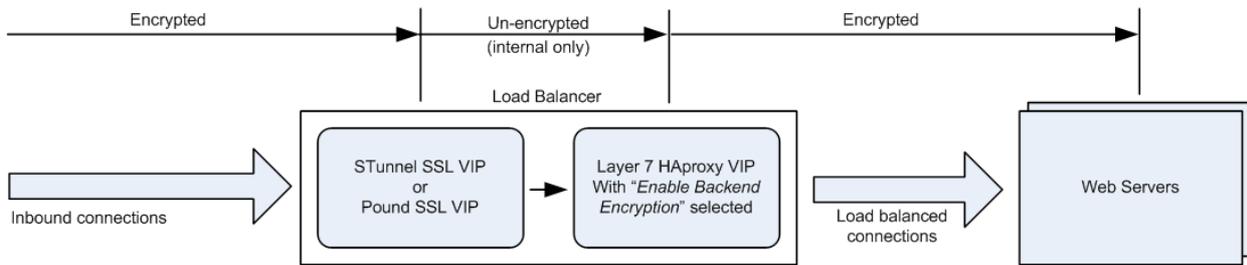
Honor Cipher Order – this option should be enabled (checked)

Client Cipher 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.

SSL TERMINATION ON THE LOAD BALANCER WITH RE-ENCRYPTION - AKA SSL BRIDGING



Notes:

- This is similar to SSL Offload, the only difference is that the connection from the load balancer to the real servers is encrypted using the certificate located on the real server, this could be a self-signed certificate since no client connections are terminated here, only at the STunnel or Pound VIP
- This mode can be enabled for the entire VIP and all associated Real Servers using the VIP option *Enable Backend encryption* or per Real Server using the *Re-Encrypt to Backend* option as detailed below

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.

It's also possible to terminate SSL on the load balancer and then re-encrypt the HTTP traffic between the load balancer and each Real Server. Each Real Server must have an SSL certificate and be correctly configured for HTTPS. This option does not check the state of the installed SSL Certificate on the Real Server which will allow for the use of locally generated SSL Certificates.

To enable re-encryption at the Virtual Server level:

1. Use the WebUI menu option: *Cluster Configuration > Layer 7 – Virtual Servers > Modify*

The screenshot shows a configuration panel with a header 'Enable Backend Encryption' and a checked checkbox. There is a help icon (question mark) to the right of the checkbox. At the bottom right of the panel are two buttons: 'Cancel' (red) and 'Update' (green).

2. Enable the option *Re-Encrypt to Backend*
3. Click **Update**
4. Now add the real servers ensuring that you specify the correct HTTPS port – typically 443

Note:

This setting only applies to Real Servers added after setting this option, it auto enables the Re-Encrypt to Backend option (see below) for all new Real Servers.

To enable re-encryption at the Real Server level:

1. For each Real Server use the WebUI menu option: *Cluster Configuration > Layer 7 – Real Servers > Modify*

Label	<input type="text" value="IIS1"/>	?
Real Server IP Address	<input type="text" value="192.168.210.240"/>	?
Real Server Port	<input type="text" value="443"/>	?
Re-Encrypt to Backend	<input checked="" type="checkbox"/>	?
Weight	<input type="text" value="100"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

2. Set *Real Server Port* to **443**
3. Enable the option *Re-Encrypt to Backend*
4. Click **Update**
5. Repeat for your other Real Server(s)

SSL – ADVANCED CONFIGURATION

POUND GLOBAL SETTINGS

Lock Pound Configuration	<input type="checkbox"/>	?
Logging	<input type="text" value="Off"/>	?
Client Timeout	<input type="text" value="30"/>	?
Global Server Timeout	<input type="text" value="60"/>	?
ulimit	<input type="text" value="81000"/>	?
Process Threads	<input type="text" value="250"/>	?
Transparent Proxy	<input type="text" value="Off"/>	?
		<input type="button" value="Update"/>

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 the process. This setting will change the maximum number of file descriptors available to the Pound process. The default is 81000.

Ulimit – Set Ulimit value for Pound the 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 also automatically enables TProxy for HAProxy.

Note:

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 [148](#).

STUNNEL GLOBAL SETTINGS

STunnel Global Settings

Debug Level Emergency (0) ▼ ?

Disable Nagle Algorithm ?

[Update](#)

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 emergency (0), Alert (1), Critical (2), err (3), Warning (4), Notice (5), Information (6), or Debug (7). The higher the number the more detail will be contained in the STunnel Logs.

Disable Nagle Algorithm – With this option ticked (enabled) the Nagle Algorithm will be disabled. More details can be found in RFC 896.

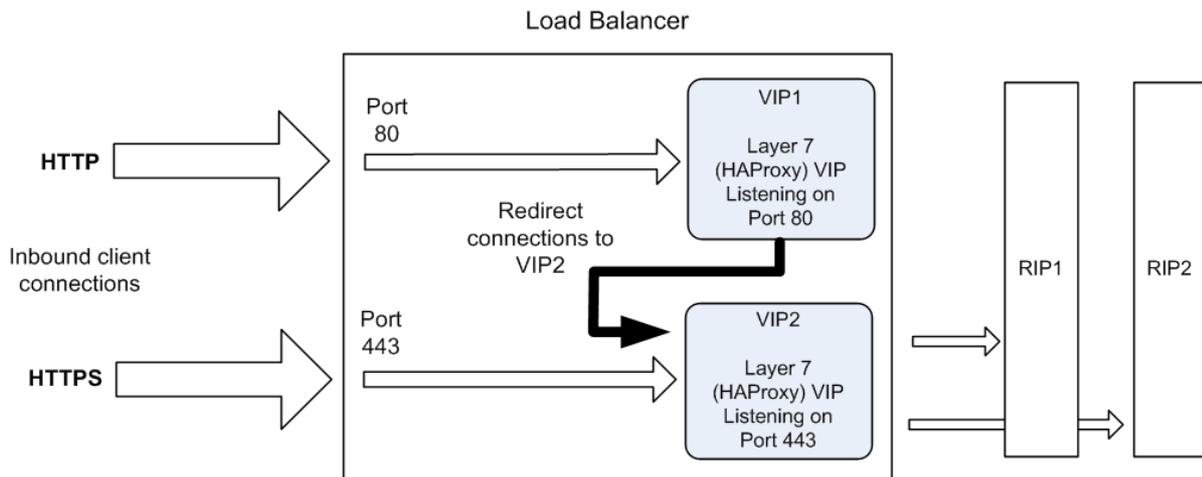
HTTP to HTTPS Redirection

The appliance supports the ability to force HTTP to HTTPS redirection. This can be achieved both when terminating SSL on the Real Servers and when offloading SSL on the load balancer as described in the following sections.

SSL TERMINATION ON THE REAL SERVERS

This method requires 2 VIPs

VIP1 & VIP2 are configured on the same IP address for HTTP/HTTPS client connections



- **VIP1** – This is a layer 7 HTTP mode VIP that listens on port 80 and redirects all connections to VIP2

Note:

VIP1 will show purple/green in the System Overview. This occurs once **Force to HTTPS** is enabled (see below).

- **VIP2** – This is a layer 7 TCP mode VIP that listens on port 443 and load balances connections between real servers RIP1 & RIP2

VIP1 Redirect Configuration

Enable the **Force to HTTPS** options as shown below and set the redirect code as required:

Force to HTTPS	<input checked="" type="radio"/> Yes <input type="radio"/> No	?
HTTPS Redirect Code	301 (Moved Permanently) ▼	?

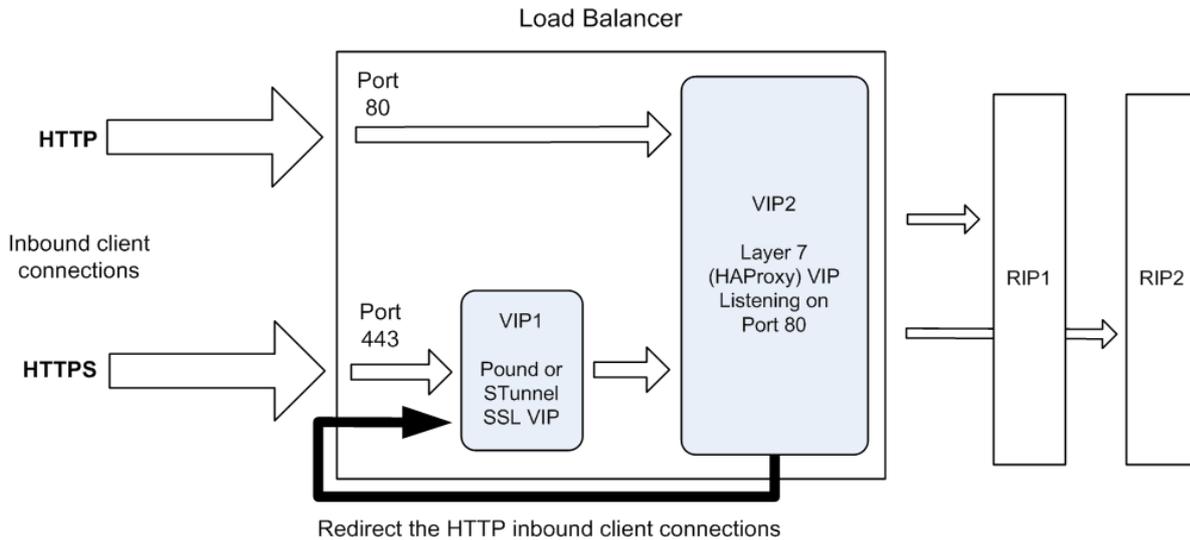
Note:

It's not possible to enable TProxy when using this configuration.

SSL TERMINATION ON THE LOAD BALANCER

This method requires 2 VIPs

VIP1 & VIP2 are configured on the same IP address for HTTP/HTTPS client connections



- **VIP1** – This is a Pound or STunnel VIP that listens on port 443, terminates the SSL connection and then forwards the decrypted HTTP connections to VIP2 on port 80
- **VIP2** – This is a layer 7 HTTP mode VIP that listens on port 80 and load balances connections between real servers RIP1 and RIP2
It also has the option *Force to HTTPS* enabled which redirects the HTTP client connections (see below)

VIP2 Redirect Configuration

Enable the **Force to HTTPS** options as shown below and set the redirect code as required:

Force to HTTPS	<input checked="" type="radio"/> Yes <input type="radio"/> No	?
HTTPS Redirect Code	301 (Moved Permanently) ▼	?

Note:

It's not possible to enable TProxy when using this configuration.

Note:

If you require to re-encrypt the data from the load balancer to the Real Server, enable the *Re-encrypt to Backend* option for the each Real Server. See page [142](#) for more details.

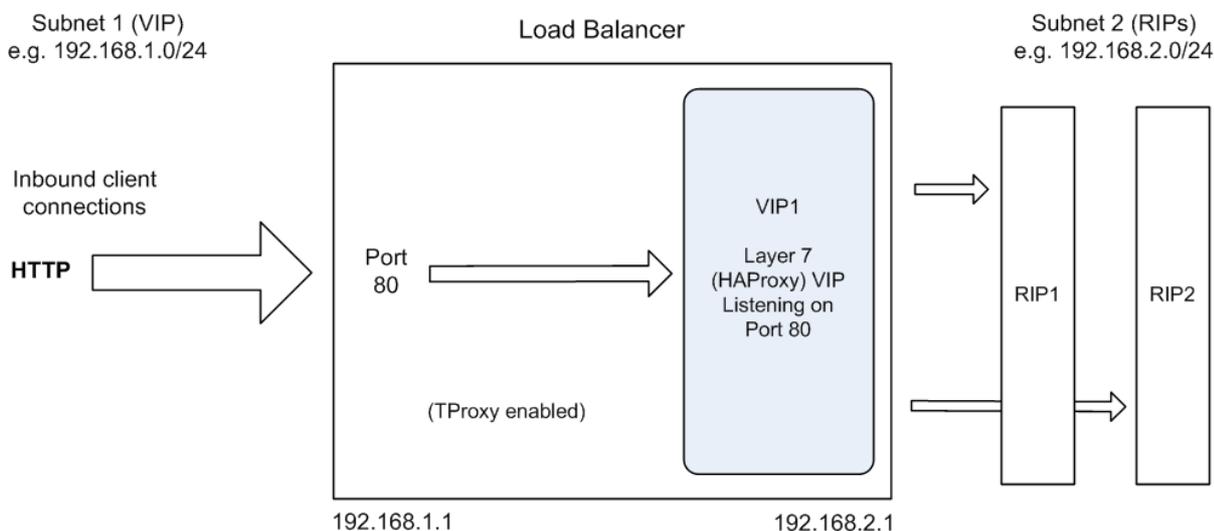
Using Transparent Proxy (TProxy)

HAProxy, Pound and STunnel are proxies which means that a new connection is established from the proxy out to the backend 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 enable transparency, i.e. 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 simple example, TProxy is enabled with a layer 7 Virtual Service.



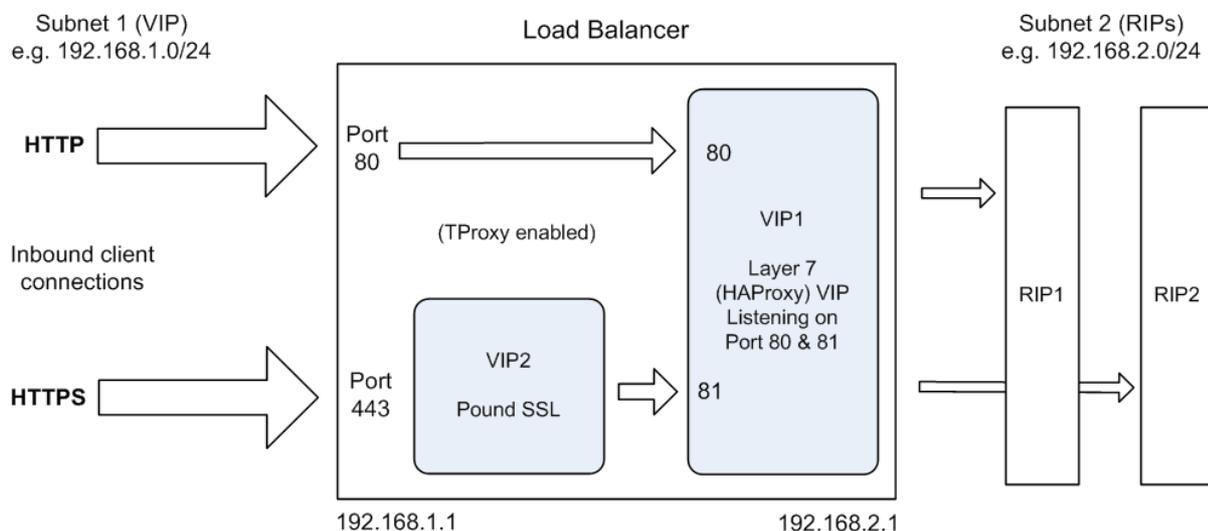
Topology Requirements/Notes

- A 2-arm configuration **must** be used, i.e. VIP1 must be on a different subnet to the RIPs – this can be achieved by using 2 IP addresses assigned to a single interface, or two separate interfaces (in the above example, eth1 = 192.168.1.1 and eth0 = 192.168.2.1)
- TProxy must be enabled using the WebUI menu 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 (e.g. 192.168.2.1). When using a clustered pair, this should be a floating IP to allow failover to the slave

TPROXY, HAPROXY & POUND

In this example, a Pound VIP (VIP2) is used to terminate SSL. Pound then passes the decrypted traffic to a layer 7 backend VIP (VIP1).

VIP1 & VIP2 are configured on the same IP address for HTTP/HTTPS client connections



Note:

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

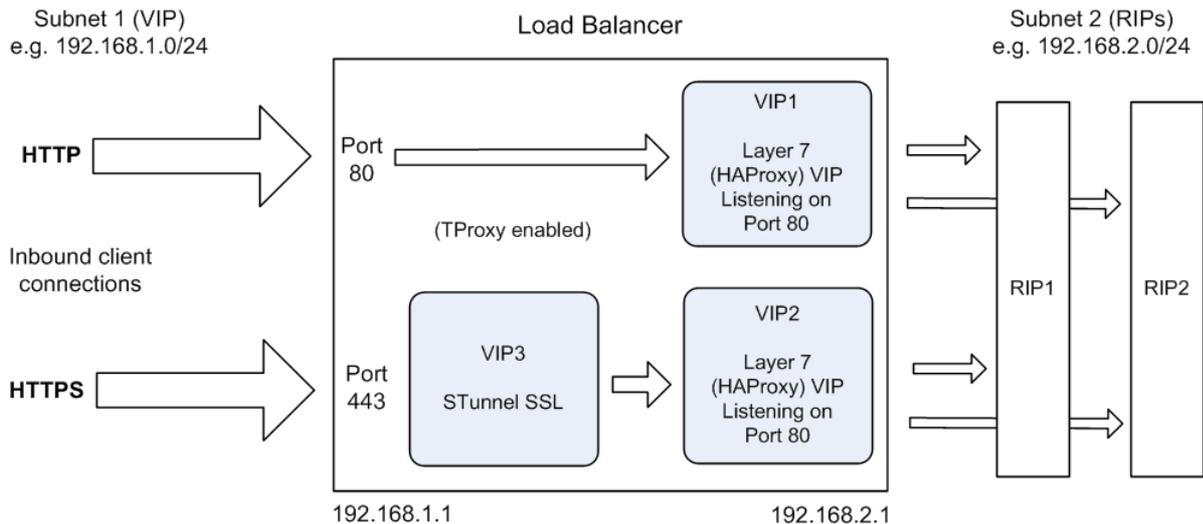
Topology Requirements/Notes

- A 2-arm configuration **must** be used, i.e. the VIPs must be on a different subnet to the RIPs – this can be achieved by using 2 IP addresses assigned to a single interface, or two separate interfaces (in the above example, eth1 = 192.168.1.1 and eth0 = 192.168.2.1)
- Configure the Layer 7 VIP (VIP1) to listen on 2 ports – e.g. 80 & 81, then use port 80 for client connections on HTTP and port 81 for the Pound backend
- When defining Real Servers for VIP1, ensure that the *Real Server Port* field is set and not left blank
- Configure the Pound VIP (VIP2) to listen on the same IP address as VIP1 but on port 443 and set its backend to be port 81 of the HAProxy VIP. Clients connect to a single IP address listening on port 80 & 443
- TProxy for HAProxy must be enabled using the WebUI menu option: *Cluster Configuration > Layer 7 – Advanced Configuration* and setting **Transparent Proxy** to 'On'
- TProxy for Pound must be enabled using the WebUI menu option: *Cluster Configuration > SSL – Advanced Configuration* and setting **Transparent Proxy** to 'On'
- On the load balanced Back-end Servers, the default gateway must be configured to be an IP address on the load balancer (e.g. 192.168.2.1). When using a clustered pair, this should be a floating IP to allow failover to the slave appliance
- If you want to enable HTTP to HTTPS redirection, you'll need to split VIP1 into 2 separate VIPs, one on port 80 with *Force to HTTPS* enabled and the other configured to accept traffic from Pound

TProxy, HAProxy & STUNNEL

In this example, a STunnel VIP (VIP3) is used to terminate SSL. STunnel then passes the decrypted traffic to a layer 7 backend VIP (VIP2). As mentioned in the previous section, when STunnel is used, 2 separate backend HAProxy VIPs are required (VIP1 & VIP2).

VIP1 & VIP3 are configured on the same IP address for HTTP/HTTPS client connections



Topology Requirements/Notes

- A 2-arm configuration **must** be used, i.e. the VIPs must be on a different subnet to the RIPs – this can be achieved by using 2 IP addresses assigned to a single interface, or two separate interfaces (in the above example, eth1 = 192.168.1.1 and eth0 = 192.168.2.1)
- Configure each Layer 7 VIP (VIP1 & VIP2) to listen on the same port, e.g. port 80. Set the IP address for VIP2 to a convenient value. Alternatively, set the IP address for VIP2 the same as VIP1 and use a different port, e.g. port 81
- Configure the same Real Servers for both VIP1 and VIP2 making sure that the Real Server Port field is set and not left blank
- TProxy for HAProxy must be enabled using the WebUI menu option: *Cluster Configuration > Layer 7 – Advanced Configuration* and setting **Transparent Proxy** to 'On'
- For VIP2, the proxy protocol must be enabled by checking the **Accept Proxy Protocol** option when modifying the VIP. Once this is enabled, it won't be possible to browse to VIP2 on port 80 directly, it will only work when accessed via the STunnel VIP (VIP3).
- Configure the STunnel VIP (VIP3) to listen on the same IP as VIP1. Clients connect to a single IP address listening on port 80 & 443. Set the Back-end IP address and port to VIP2
- The proxy protocol must also be enabled for the STunnel VIP (VIP3). This is done by checking the **Enable Proxy Protocol** 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 appliance
- If you want to enable HTTP to HTTPS redirection, enable *Force to HTTPS* on VIP1

Proxy Protocol

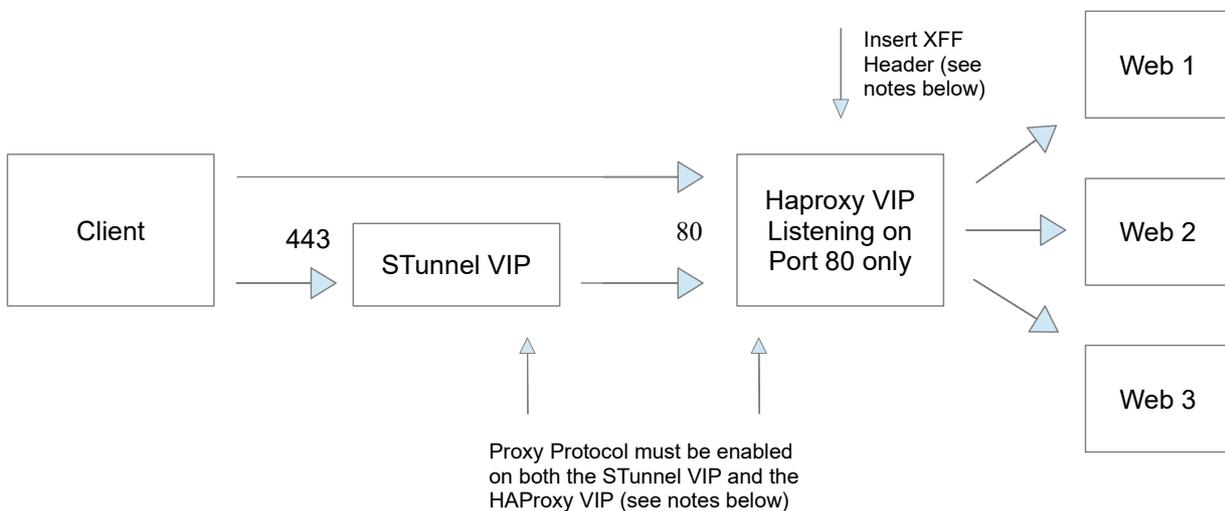
Proxy Protocol is an Internet protocol used to carry connection information from the source requesting the connection to the destination for which the connection was requested. Several systems/software applications have implemented Proxy Protocol, an up to date list can be found [here](#). Both HAProxy and STunnel support the Proxy Protocol in the Loadbalancer.org appliance.

Note:

For more information about Proxy Protocol, please refer to the following URL:
<http://www.haproxy.org/download/1.8/doc/proxy-protocol.txt>

EXAMPLE CONFIGURATION

The following example shows how Proxy Protocol can be used with STunnel and HAProxy (both are non transparent proxies) to present the original client source IP address to the load balanced servers (Web 1, Web 2 & Web 3). This is very simple to deploy because a one-arm single subnet configuration can be used.



Configuration Notes

- To enable Proxy Protocol, modify the STunnel VIP and check (tick) the *Enable Proxy Protocol* checkbox, then select the layer 7 HAProxy VIP from the *Bind Proxy Protocol to L7 VIP* drop-down (this will configure both the STunnel and HAProxy VIPs)
- Ensure that *Set X-Forward-For header* is checked (ticked), this is the default for new layer 7 VIPs

Once configured, the **X-Forwarded-For** header received by the load balanced servers Web 1, Web 2 & Web 3 will contain the source IP address of the Client.

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. Floating IP(s) are added automatically when new Virtual Services are created.

It's also possible to manually define floating IP(s) if required, this is normally only required when manually configuring firewall marks or when using layer 4 NAT mode or 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 balancer appliance's (normally the master) owns the Floating IP(s) at any time.

To manually add a Floating IP:

1. Using the WebUI, navigate to: *Cluster Configuration > Floating IPs*

FLOATING IPs	
192.168.111.40	Delete
192.168.111.42	Delete

New Floating IP

Add Floating IP

2. Specify the new floating IP
3. Click **Add Floating IP**

Note:

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 the IP address is changed, this must be done manually.

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. For layer 4 VIPs the feedback method can be set to either agent or HTTP, for Layer 7 VIPs, only the agent method is supported.

A telnet to port 3333 on a Real Server with the agent installed will return the current idle stats as an integer value in the range 0 – 100. The figure returned can be related to CPU utilization, RAM usage or a combination of both. This can be configured using the XML configuration file located in the agents installation folder (by default C:\ProgramData\LoadBalancer.org\LoadBalancer).

The load balancer typically expects a 0-99 integer response from the agent which by default relates to the current CPU idle state, e.g. a response of 92 would imply that the Real Servers CPU is 92% idle. The load balancer will then use the formula $(92/100 * \text{requested_weight})$ to find the new optimized weight.

Note:

The 'Requested Weight' is the weight set in the WebUI for each Real Server. For more information please also refer to the following blog article:

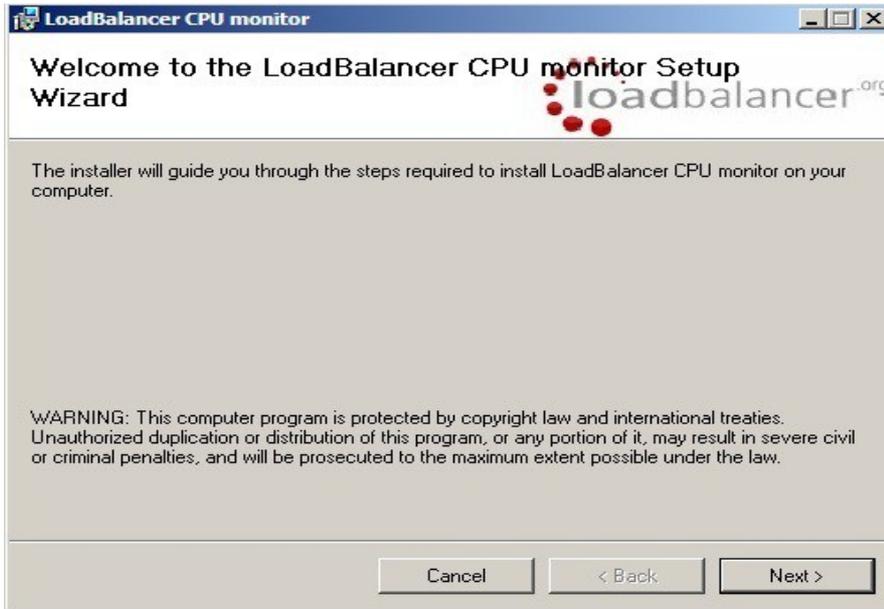
<http://www.loadbalancer.org/blog/open-source-windows-service-for-reporting-server-load-back-to-haproxy-load-balancer-feedback-agent>

WINDOWS AGENT

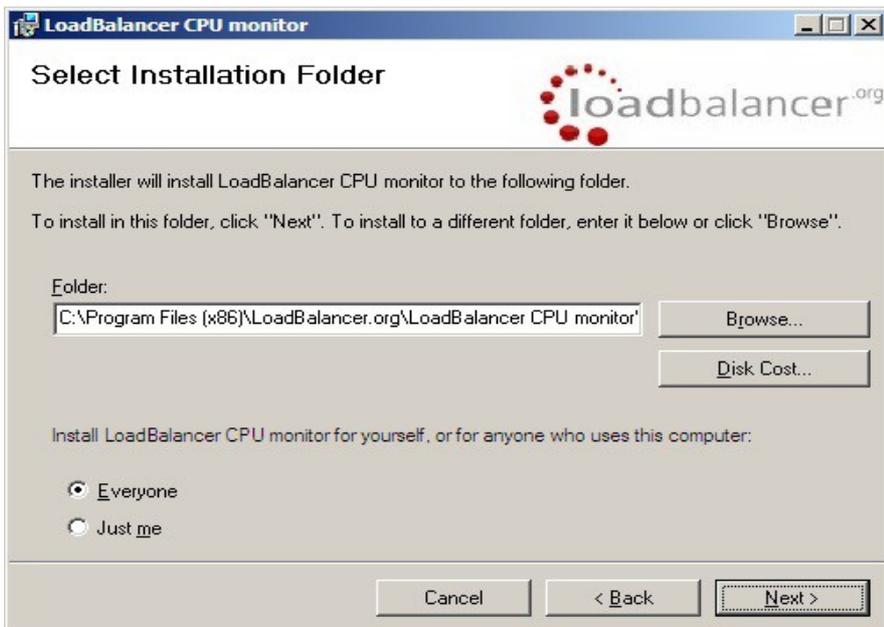
The latest Windows feedback agent can be downloaded from:

<http://downloads.loadbalancer.org/agent/loadbalanceragent.msi>

To install the agent, run loadbalanceragent.msi on each Real Server:



Click **Next**



Select the installation folder and click **Next**



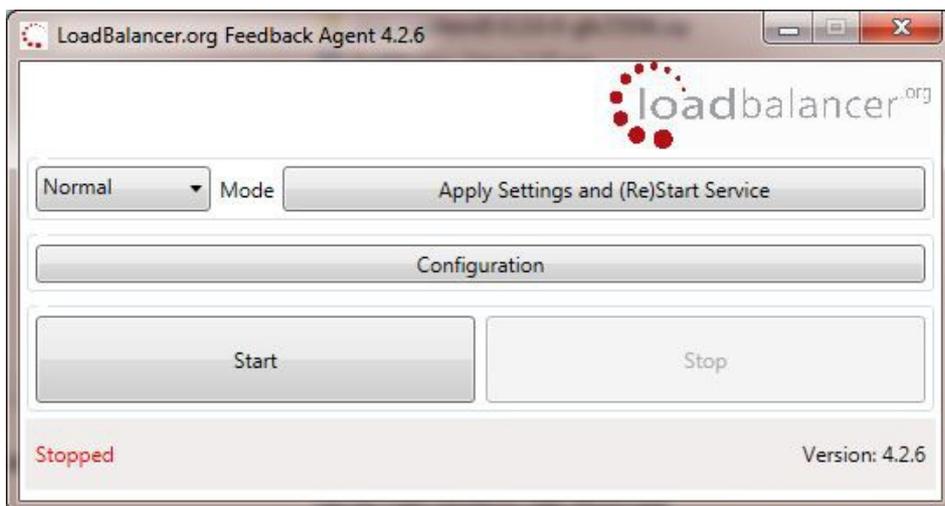
Click **Next** to start the installation

Note:

The agent should be installed on all Real Servers in the cluster.

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

LINUX/UNIX AGENT

The Linux feedback agent files can be downloaded using the following links:

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 & Testing

```
# 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.
```

Note:

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

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.

CONFIGURATION

As mentioned, both layer 4 and layer 7 VIPs can be configured to use the feedback agent. To Configure Virtual Services to use Agent/HTTP Feedback follow the steps below:

1. Go to *Cluster Configuration > Layer 4 - Virtual Services*
 or
Layer 7 - Virtual Services

2. Click **Modify** next to the Virtual Service

3. Change the Feedback Method to either **Agent** or **HTTP** for layer 4 VIPs
4. Change the Feedback Method to **Agent** for layer 7 VIPs
5. Click **Update**
6. Reload/restart services as prompted

Configuring the Appliance via CLI, API & direct Service Calls

A command line interface (CLI) is included that enables various appliance features to be configured and controlled. A JSON based Application Programming Interface (API) has also been added that enables CLI commands to be called from a Web Service.

It's also possible to directly control layer 4 and layer 7 services, although the disadvantage here is that changes made will not be reflected in the System Overview. If changes are made via the CLI or API, the System Overview is kept in sync.

COMMAND LINE INTERFACE (CLI)

Action Category	Action	Example Command
Appliance Status actions:	Get node status	<code>lbcli--action nodestatus</code>
System Overview actions:	Drain a server	<code>lbcli --action drain --vip <VIP Name> --rip <RIP Name></code>
	Halt a server	<code>lbcli --action halt --vip <VIP Name> --rip <RIP Name></code>
	Online a server	<code>lbcli --action online --vip <VIP Name> --rip <RIP Name></code>
VIP actions:	Add a VIP	<p>Layer 4 VIP</p> <pre>lbcli --action add-vip --layer 4 --vip <VIP Name> --ip <VIP IP Address> --ports <ports> --forwarding <gate masq ipip> --protocol <tcp udp></pre> <p>Layer 7 VIP</p> <pre>lbcli --action add-vip --layer 7 --vip <VIP Name> --ip <VIP IP Address> --ports <ports> --mode <http tcp></pre> <p><i>optional syntax:</i></p> <pre>--fallback_ip <IP Address> --fallback_port <Port></pre>

	Delete a VIP	lbcli --action delete-vip --vip <VIP Name>
RIP actions:	Add a RIP	lbcli --action add-rip --vip <VIP Name> --rip <RIP Name> --ip <RIP IP Address> --port <Port Value> --weight <Weight Value> <i>optional syntax – layer 4 only:</i> --minconn <minconn> --maxconn <maxconn> <i>optional syntax – layer 7 only:</i> --encrypted <on off>
	Delete a RIP	lbcli --action delete-rip --vip <VIP Name> --rip <RIP Name>
WAF actions:	Add a WAF	lbcli --action add-waf --vip <VIP Name> --waf <WAF Name> <i>optional syntax:</i> --in_anom_score <1-99> --out_anom_score <1-99> --req_data <on off> --resp_data <on off> --audit <on off> --rule_engine <on off>
	Delete a WAF	lbcli --action delete-waf --vip <VIP Name> --waf <WAF Name>
Floating IP actions:	Add a FIP	lbcli --action add-floating-ip --ip <IP Address>
	Delete a FIP	lbcli --action delete-floating-ip --ip <IP Address>
Service actions:	Restart HAProxy	lbcli --action restart-haproxy
	Reload HAProxy	lbcli --action reload-haproxy
	Restart Ldirectord	lbcli --action restart-ldirectord
	Reload Ldirectord	lbcli --action reload-ldirectord
	Generate Support Archive	lbcli --action support-download

Note:

The CLI / API are constantly being developed, so if lbcli functionality that you require is not listed in the table above, please contact support@loadbalancer.org to check the latest command availability.

CLI command help

for a complete list of all lbcli commands, use the following command:

```
lbcli --help lbcli
```

to obtain more detailed help for a particular action including optional sub values, use the following syntax:

```
lbcli --help <action>
```

e.g.

```
lbcli --help add-vip
```

Running lbcli from a remote Linux Host

These commands can be run from a remote Linux host. This example halts VIP1/RIP1:

```
ssh root@192.168.111.42 "lbcli --action halt --vip VIP1 --rip RIP1"
```

Running lbcli from a remote Windows Host

These commands can be run from a remote Windows host. This example halts VIP1/RIP1:

```
plink -pw loadbalancer root@192.168.111.42 "lbcli --action halt --vip VIP1 --rip RIP1"
```

Notes:

- PuTTY must be installed to use the *plink* command (see: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>)
- 'loadbalancer' is the default password for the root user
- 192.168.111.42 is the IP address of the load balancer

APPLICATION PROGRAMMING INTERFACE (API)

Enabling the API

By default, the API is disabled. To enable the API, edit the file `/etc/loadbalancer.org/api-credentials` and un-comment the *username*, *password* and *apikey* lines, then save the file. The default username, password and apikey can be changed as required. Once enabled, API calls can be made using HTTP POST requests. As mentioned, the API enables CLI commands to be called from a Web Service.

HTTP POST Request URL

The JSON requests must be posted to the following URL on the load balancer:

```
https://<appliance IP address>:9443/api/
```

Testing

To test the functionality of the API, a browser add-on such as *HttpRequester* or *Poster* can be useful to form and post the requests.

Syntax Validation

For validating JSON syntax, the website <http://jsonlint.com/> can be used. Simply paste the JSON into the window provided, then click **Validate JSON**.

Examples

To illustrate how the JSON API calls are formed, the following examples show the CLI command and the equivalent JSON API command in each case.

Example 1 – Halt a Server

This example shows how RIP1 of VIP1 is halted.

lbcli command:

```
lbcli --action halt --vip VIP1 --rip RIP1
```

JSON equivalent:

```
{
  "auth": {
    "apikey": "eP68pvSMM8dvn051LL4d35569d438ue0"
  },
  "action": [{
    "command": "halt"
  }],
  "syntax": [{
    "vip": "VIP1",
    "rip": "RIP1"
  }]
}
```

Example 2 – Add a Layer 7 VIP

This example shows how to add a Layer 7 HTTP mode VIP.

lbcli command:

```
lbcli --action add-vip --layer 7 --vip VIP1 --ip 192.168.1.1 --ports 80 --mode http
```

JSON equivalent:

```
{
  "auth": {
    "apikey": "eP68pvSMM8dvn051LL4d35569d438ue0"
  },
  "action": [{
    "command": "add-vip"
  }],
  "syntax": [{
    "layer": "7",
    "vip": "VIP1",
  }]
}
```

```

        "ip": "192.168.1.1",
        "ports": "80",
        "mode": "http"
    }
}

```

Example 3 – Add a RIP

This example shows how to add a RIP.

lbcli command:

```
lbcli --action add-rip --vip VIP1 --rip RIP1 --ip 192.168.1.2 --ports 80 --weight 100
```

JSON equivalent:

```

{
    "auth": {
        "apikey": "eP68pvSMM8dvn051LL4d35569d438ue0"
    },
    "action": [{
        "command": "add-rip"
    }],
    "syntax": [{
        "vip": "VIP1",
        "rip": "RIP1",
        "ip": "192.168.1.2",
        "port": "80",
        "weight": "100"
    }]
}

```

Example 4 – Restart HAProxy

This example shows how to restart HAProxy.

lbcli command:

```
lbcli --action restart-haproxy
```

JSON equivalent:

```

{
    "auth": {
        "apikey": "eP68pvSMM8dvn051LL4d35569d438ue0"
    },
    "action": [{
        "command": "restart-haproxy"
    }]
}

```

Example 5 – Multiple actions in a single command

This example shows how multiple actions can be called with one POST. This example adds a layer 7 VIP, a layer 7 RIP and an Stunnel VIP, then restarts HAProxy and STunnel.

```
{
  "auth": {
    "apikey": "eP68pvSMM8dvn051LL4d35569d438ue0"
  },
  "action": [{
    "command": "add-vip",
    {
      "command": "add-rip",
      {
        "command": "termination"},
      {
        "command": "restart-haproxy"},
      {
        "command": "restart-stunnel"}
  ],
  "syntax": [{
    "layer": "7",
    "vip": "VIP1",
    "ip": "192.168.111.225",
    "ports": "80",
    "mode": "http"},
    {
      "vip": "VIP1",
      "rip": "RIP1",
      "ip": "192.168.110.240",
      "port": "80",
      "rip_type": "ipv4",
      "weight": "100"},
    {
      "function": "add",
      "type": "stunnel",
      "vip": "SSL1",
      "ip": "192.168.111.225",
      "port": "443",
      "backend_ip": "192.168.111.225",
      "backend_port": "80",
      "sslcert": "cert1"}
  ]
}
```

USING IPVSADM TO CONFIGURE LAYER 4 SERVICES

For layer 4 services, the `ipvsadm` command can be used. Several examples are provided below.

Add a TCP based Virtual Service & use weighted round robin scheduling:

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

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 weighted least connection scheduling:

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

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
```

Command output:

```
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
```

Note:

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 when `ipvsadm/socat` commands are used. Using the `lbcli` command or the API does not have this disadvantage since the System Overview will show the correct VIP and RIP status.

USING LINUX SOCKET COMMANDS TO CONFIGURE LAYER 7 SERVICES

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
```

Note:

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

<http://haproxy.1wt.eu/download/1.7/doc/configuration.txt>

Note:

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 when ipvsadm/socat commands are used.

Using the **lbcli** command or the API does not have this disadvantage since the System Overview will show the correct VIP and RIP status.

Chapter 7 – Web Application Firewall (WAF)

Introduction

The Web Application Firewall (WAF) is based on the Modsecurity Open Source Project.

The default vulnerability rule-set is based on the “OWASP top 10”. This defines 10 areas of vulnerability that can effect Web Applications. These are summarised in the table below:

Category	Description
A1 - Injection	Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker’s hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2 - Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users’ identities.
A3 - Cross-Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim’s browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A4 - Insecure Direct Object References	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
A5 - Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.
A6 - Sensitive Data Exposure	Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.
A7 - Missing Function Level Access Control	Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.
A8 - Cross-Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim’s browser to send a forged HTTP request, including the victim’s session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim’s browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
A9 - Using Components with	Components, such as libraries, frameworks, and other software

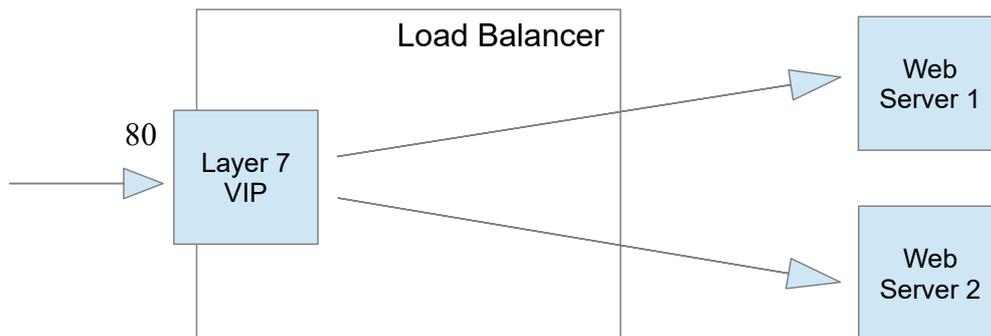
Known Vulnerabilities	modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.
A10 - Unvalidated Redirects and Forwards	Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages

More details can be found here: https://www.owasp.org/index.php/Top_10_2013-Top_10

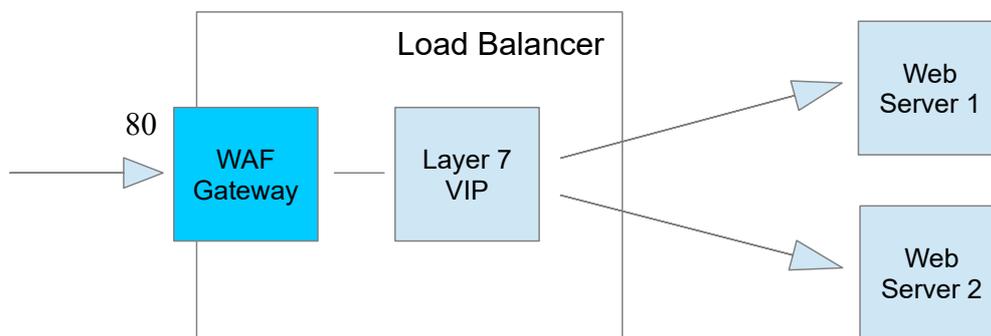
Implementation Concepts

When a WAF gateway is created on the load balancer, the data path is automatically modified so that the WAF becomes the initial connection point for inbound client connections as illustrated below:

Data flow before WAF is deployed



Modified data flow once WAF is deployed



Notes:

- When defining a WAF Gateway on the load balancer, the associated layer 7 VIP must be selected from a drop-down list. This enables the WAF to be automatically configured to listen on the same

TCP socket as the original layer 7 VIP

The WAF gateway is then automatically configured to forward packets to the original layer 7 VIP

- Each WAF gateway is associated with one layer 7 VIP
- Once the WAF gateway is defined, the *Label*, *IP Address*, *Port* and *Protocol* of the associated layer 7 VIP cannot be edited to ensure the association remains intact. If changes to these settings are required, remove the WAF, make the changes, then recreate the WAF
- Each WAF gateway is comprised of an additional layer 7 VIP which acts as the WAF front-end and an Apache/ModSecurity config. Both are auto-created when the WAF Gateway is configured

WAF Gateway Configuration

INITIAL SETUP

For reasons mentioned in the previous section, the layer 7 VIP must be created first, followed by the WAF gateway.

Step 1 – Create the Layer 7 VIP

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Services* and click **Add a new Virtual Service**

Label	<input type="text" value="HTTP-Cluster"/>	?
Virtual Service	IP Address <input type="text" value="192.168.110.46"/>	?
	Ports <input type="text" value="80"/>	?
Layer 7 Protocol	<input type="text" value="HTTP Mode"/>	?
Manual Configuration	<input type="checkbox"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

2. Enter a suitable Label (name) for the VIP, e.g. **HTTP-Cluster**
3. Enter a valid IP address, e.g. **192.168.110.46**
4. Enter a valid port, e.g. **80**
5. Click **Update**

Step 2 – Define the associated Real Servers (RIPs)

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Real Servers* and click **Add a new Real Server** next the the VIP just created

Label	<input type="text" value="Web1"/>	?
Real Server IP Address	<input type="text" value="192.168.110.241"/>	?
Real Server Port	<input type="text" value="80"/>	?
Weight	<input type="text" value="100"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

2. Enter a suitable Label (name) for the RIP, e.g. **Web1**
3. Enter a valid IP address, e.g. **192.168.110.241**
4. Enter a valid port, e.g. **80**
5. Click **Update**

Step 3 – Define the WAF Gateway

1. Using the WebUI, navigate to: *Cluster Configuration > WAF - Gateway* and click **Add a new WAF gateway**

Select Layer 7 Virtual Service	<input type="text" value="HTTP-Cluster"/>	?
WAF Label	<input type="text" value="WAF1"/>	?
Rule Engine Traffic Blocking	<input type="checkbox"/>	?
Process Request Data	<input checked="" type="checkbox"/>	?
Process Response Data	<input type="checkbox"/>	?
Inbound Anomaly Score	<input type="text" value="5"/>	?
Outbound Anomaly Score	<input type="text" value="5"/>	?
Audit Mode	<input type="checkbox"/>	?
WAF Proxy Timeout	<input type="text" value="60"/>	?
Double Login Enable	<input type="checkbox"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

2. Select the VIP created in step 1 in the drop down
3. Enter a suitable Label (name) for the WAF, e.g. **WAF1**
4. Leave other options at their default settings
5. Click **Update**

Step 4 – Reload Services to Apply the new Settings

1. Click *System Overview* in the WebUI
2. Reload the services (Apache and HAProxy) as prompted in the blue message box

Step 5 – View Configured Services

1. The original layer 7 VIP and the auto created layer 7 WAF front-end VIP are now displayed in the system overview as shown below:

SYSTEM OVERVIEW ?								2015-07-02 13:20:18 UTC
VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE		
HTTP-Cluster	192.168.110.46	65435	0	HTTP	Layer 7	Proxy		
WAF1	192.168.110.46	80	0	HTTP	Layer 7	Proxy		

WAF GATEWAY TIMEOUT

The WAF Gateway timeout value is set to 60 seconds by default. This can be changed if required.

To enable blocking mode:

1. Using the WebUI, navigate to: *Cluster Configuration > WAF – Gateway* and click **Modify** next to the relevant WAF
2. Set *Web Proxy Timeout* to the required value
3. Click **Update**

WEB GATEWAY AUTHENTICATION

The Web Gateway supports the following authentication modes:

- Locally defined static user
- Google OpenID

Once enabled, users will be prompted for credentials when accessing the WAF:

SECURE GATEWAY

To enable Authentication:

1. Using the WebUI, navigate to: *Cluster Configuration > WAF – Gateway* and click **Modify** next to the relevant WAF
2. Set *Double login Mode* to either **Static Users** or **Google – OpenID**
3. Configure the related authentication fields as required
4. Click **Update**

WAF GATEWAY OPERATING MODE

By default, the WAF Gateway only logs any breaches of the ModSecurity rules, it doesn't block any requests.

The WAF gateway should initially be left in this mode so that any rule matches are logged. If there are no false positives, blocking mode can be enabled to reject any malicious requests and respond with a 403 Forbidden response.

To enable blocking mode:

5. Using the WebUI, navigate to: *Cluster Configuration > WAF – Gateway* and click **Modify** next to the relevant WAF
6. Enable the *Rule Engine Traffic Blocking* checkbox
7. Click **Update**
8. Restart/Reload the services (WAF and HAProxy) as prompted in the blue message box

WAF GATEWAY RULE CONFIGURATION

The WAF supports two methods of controlling what's blocked and what's allowed through. These are:

- Rule White-Listing
- Anomaly Scoring

RULE WHITE-LISTING

Rules can easily be switched off if required. This maybe required if the default settings prove to be too restrictive and you're seeing false positives.

To disable rules:

1. Using the WebUI, navigate to: *Cluster Configuration > WAF – Manual Configuration*
2. Select the relevant WAF in the drop-down

```

1  # Default ruleset generated by Loadbalancer.org.
2  # These can be removed.
3
4  # Do not allow an invalid range from ping of death attack MS15034
5
6  #SecRule REQUEST_HEADERS:Range "@rx (?i)^(bytes\s*=)(.*)?([0-9]{10,})" \
7  # "id:'100007',phase:1,t:none,block,setvar:tx.anomaly_score+={tx.critical_anomaly_score},msg:'Invalid header r
8
9
10 #Example for whitelisting an ip address
11 #replace the ip in the example with the one you want to whitelist
12
13 #SecRule REMOTE_ADDR "^192.168.2.21"
14 # "phase:1,nolog,allow,ctl:ruleEngine=Off,id:100008\"
15
16 #Example to allow ALL users to access the website by ip address.
17 #Rather than just by URL
18
19 #SecRuleRemoveById 960017

```

3. Add an extra line specifying the rule to disable, e.g.

```
SecRuleRemoveById 960022
```

Note:

The rule ID can be obtained from the logs whilst in non blocking mode. For more details on viewing the logs please refer to page [170](#).

4. Click **Update**
5. Restart/Reload system services as directed in the blue message box

Browsing by IP Address

The default rules block browsing by IP address. e.g. `http://192.168.110.10/`. This particular rule can be disabled by going to *Cluster Configuration > WAF - Manual Configuration*, selecting the WAF in the dropdown, then un-commenting the following line, i.e. removing the `#`

```
#SecRuleRemoveById 960017
```

Then restarting/reloading services as directed in the blue message box.

Any rule can be excluded in this way, as long as you know the ID, this can be obtained from the log entry as explained in the next section.

ANOMALY SCORING

Inbound and outbound anomaly scores can be configured using the *Inbound Anomaly Score* and *Outbound Anomaly Score* fields. The default value for both is 5. This is equivalent to an occurrence of one Critically rated anomaly. These values can be adjusted to suit your specific environment. In the examples presented in the section below, the total inbound score is 10, so the inbound anomaly score would need to be set to at least 11 to ensure that the page is not blocked by the WAF.

WAF Gateway Logging & Monitoring

The WAF always logs malicious requests. The actual log entry depends on whether the WAF is running in logging only mode or blocking mode.

To view the log:

1. In the WebUI select: *Logs > WAF Logs*
2. In the drop-down select *Error <WAF_NAME>*

Example Log Entries:

1 - Example log entry (LOGGING ONLY mode)

```
[Fri Jul 08 13:07:26 2016] [error] [client 192.168.111.105] ModSecurity: Warning. Operator LT
matched 5 at TX:inbound_anomaly_score. [file
"/etc/httpd/modsecurity.d/activated_rules/modsecurity_crs_60_correlation.c onf"] [line "33"] [id
"981203"] [msg "Inbound Anomaly Score (Total Inbound Score: 10, SQLi=2, XSS=):
981243-Detects classic SQL injection probings 2/2"] [hostname "support.lbtestdom.com"] [uri
"/__swift/themes/client/images/icon_widget_submitticket.png"] [unique_id
"V3@ljn8AAAEAAgt7s@oAAAAD"]
```

In this example, the matching rule is: **981203** as highlighted above.

The total inbound score is **10**.

2 - Example log entry (BLOCKING mode)

```
[Fri Jul 08 13:38:41 2016] [error] [client 192.168.111.105] ModSecurity: Access denied with
code 403 (phase 2). Pattern match "(.*)" at TX:981246-Detects basic SQL authentication bypass
attempts 3/3-OWASP_CRS/WEB_ATTACK/SQLI-REQUEST_COOKIES:SWIFT_client. [file
"/etc/httpd/modsecurity.d/activated_rules/modsecurity_crs_49_inbound_blocking.conf"] [line
"26"] [id "981176"] [msg "Inbound Anomaly Score Exceeded (Total Score: 10, SQLi=2,
XSS=): Last Matched Message: 981243- Detects classic SQL injection probings 2/2"] [data "Last
Matched Data: \\x22templategroupid\\x22:\\x22"] [hostname "support.lbtestdom.com"] [uri
"/favicon.ico"] [unique_id "V3@s4X8AAAEAAafGAOEAAAA"]
```

in this example, the matching rule is: **981176** as highlighted above

The total inbound score is **10** and **access is denied** because the default threshold of 5 has been exceeded.

Modifying Default Actions

Default actions can easily be modified is required, a good example is to modify the response when access is denied. By default a 403 (forbidden) response is returned to the requesting client. This can be changed to redirect to a different URL using the **SecDefaultAction** as detailed below.

To customize default behavior:

1. Using the WebUI navigate to: *Cluster Configuration > WAF – Manual configuration*
2. Using the drop-down at the top of the page, select the required WAF
3. In the Edit Window, add the following lines at the bottom of the page as shown below:

```
SecDefaultAction "phase:1,deny,log,redirect:https://www.yourdomain.com/pageforbidden.html"
SecDefaultAction "phase:2,deny,log,redirect:https://www.yourdomain.com/pageforbidden.html"
```

```

1  # Default ruleset generated by Loadbalancer.org.
2  # These can be removed.
3
4  # Do not allow an invalid range from ping of death attack MS15034
5
6  #SecRule REQUEST_HEADERS:Range "@rx (?i)^(bytes\s*=)(.*){10,}" \
7  #id:'100007',phase:1,t:none,block,setvar:tx.anomaly_score+%(tx.critical_anomaly_score),msg:'Invalid header r
8
9
10 #Example for whitelisting an ip address
11 #replace the ip in the example with the one you want to whitelist
12
13 #SecRule REMOTE_ADDR "^192.168.2.21" \
14 #phase:1,nolog,allow,ctl:ruleEngine=Off,id:100008
15
16 #Example to allow ALL users to access the website by ip address.
17 #Rather than just by URL
18
19 #SecRuleRemoveById 960017
20
21 SecDefaultAction "phase:1,deny,log,redirect:https://www.yourdomain.com/pageforbidden.html"
22 SecDefaultAction "phase:2,deny,log,redirect:https://www.yourdomain.com/pageforbidden.html"
23
```

4. Click **Update**
5. Reload the services (Apache and HAProxy) as prompted in the blue message box at the top of the screen

Note:

For more information, please refer to the [ModSecurity Reference Manual](#) or contact support@loadbalancer.org

Chapter 8 – Real Server Health Monitoring & Control

Configuring Health Checks

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.

HEALTH CHECKS FOR 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 WebUI menu option: *Cluster Configuration > Layer 4 - Virtual Services > Modify*

The exact options available depend on the check type selected. For the default (*Connect to port*), one field is required – *Check Port*.

Health Checks	Check Type	Connect to port ▼	?
	Check Port	<input type="text"/>	?

As the *Check Type* drop-down is changed, the field list changes. The full list of options is shown below:

Health Checks	Check Type	Connect to port ▼	?
	Check Port	<input type="text"/>	?
Feedback Method			?
Fallback Server	IP Address		?
	Port	<input type="text"/>	?

Connect to port
 Negotiate
Connect to port
 ping server
 External script
 No checks, always Off
 No checks, always On
 5 Connects, 1 Negotiate
 10 Connects, 1 Negotiate

Check Types:

Negotiate – Sends a request and looks for a specific response

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 external script for the health check. Specify the script in the drop-down

No checks, always Off – All Real Servers are assumed to be down

No checks, always On – All Real Servers are assumed to be up

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

Note:

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

Option	Sub Option	Health-Check Description
Negotiate		<i>Send a request and matches a receive string</i>
	Check Port	The port to monitor. This can normally be left blank in which case the port checked is the same port defined for the Real Servers. Note that for DR mode, the port cannot be specified at the Real Server level, so the port specified for the VIP is used. Sometimes the check port differs from service port.
	Protocol	<p>HTTP – use HTTP as the negotiate protocol (also requires filename, path + text expected)</p> <p>HTTPS – use HTTPS as the negotiate protocol (also requires filename, path + text expected)</p> <p>HTTP Proxy – Use an HTTP proxy check</p> <p>FTP – use FTP as the negotiate protocol (also requires login/password, filename in the default folder)</p> <p>IMAP (IPv4 only) – use IMAP as the negotiate protocol (requires login/password)</p> <p>IMAPS (IPv4 only) - use IMAPS as the negotiate protocol (requires login/password)</p> <p>POP – use POP as the negotiate protocol (also requires login/password)</p> <p>POPS – use POPS as the negotiate protocol (also requires login/password)</p> <p>LDAP (IPv4 only) – use LDAP as the negotiate protocol (also requires username/password)</p> <p>SMTP – use SMTP as the negotiate protocol</p> <p>NNTP (IPv4 only) – use NNTP as the negotiate protocol</p> <p>DNS – use DNS as the negotiate protocol</p> <p>MySQL (IPv4 only) – use MySQL as the negotiate protocol (also requires username/password)</p> <p>SIP – use SIP as the negotiate protocol (also requires username/password)</p> <p>Simple TCP – Sends a request string to the server and checks the response</p> <p>RADIUS (IPv4 only) – use RADIUS as the negotiate protocol (also requires username/password)</p> <p>Additional Negotiate Check Options (depending on type selected) :</p> <p><i>Login</i> – the username when authentication is required</p> <p><i>Password</i> – the password when authentication is required</p> <p><i>Database Name</i> - The database to use for the MySQL check</p> <p><i>Radius Secret</i> - the RADIUS secret string for the RADIUS negotiate check</p>
	Virtual Host	Used when using a negotiate check with HTTP or HTTPS. Sets the host header used in the HTTP request. In the case of HTTPS this generally needs to match the common name of the SSL certificate. If not set then the host header will be derived from the request url for the real server if present. As a last resort the IP

		address of the real server will be used.
	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 a SQL SELECT query. (Note: 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 check to be a success. The 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.
Connect to Port		<i>Attempt to make a connection to the specified port</i>
	Port	The port to monitor. This can normally be left blank in which case the port checked is the same port defined for the Real Servers. Note that for DR mode, the port cannot be specified at the Real Server level, so the port specified for the VIP is used. Sometimes the check port differs from service port.
Ping Server		<i>Test Real Server availability using an ICMP ping</i>
External Script		<i>Call an external script to perform the health-check</i>
	External Script	Select the required external check script from the drop-down. New scripts should be placed in /var/lib/loadbalancer.org/check, and given world read and execute permissions. The drop-down will be auto-updated to include any new scripts added. Note: By default the Microsoft SQL external health-check is not available in the drop down. This health-check requires several Microsoft related pre-requisites such as the Microsoft Linux ODBC driver, and these must first be installed and configured. To install the pre-requisites and configure required settings, at the console or via an SSH session, login as root and run the following command:

		\$ lb_mssql -i Once completed, the additional option "ms-sql-check" will appear in the External Script drop-down.
No Checks Always Off		<i>No checking will take place and no real or fallback servers will be activated</i>
No Checks Always On		<i>No checking will take place and real servers will always be assumed to be up</i>
5 Connects, 1 Negotiate		<i>Repeating pattern of 5 Connect checks followed by 1 Negotiate check</i>
	Check Port	(See above)
	Protocol	(See above)
	Virtual Host	(See above)
	Request to Send	(See above)
	Response Expected	(See above)
10 Connects, 1 Negotiate		<i>Repeating pattern of 10 Connect checks followed by 1 Negotiate check</i>
	Check Port	(See above)
	Protocol	(See above)
	Virtual Host	(See above)
	Request to Send	(See above)
	Response Expected	(See above)

Custom External Script Example

The custom check script, used with the external check type. The script should be placed in **/var/lib/loadbalancer.org/check**, and given world read and execute permissions. The drop-down will be auto-updated to include any new scripts added.

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="$4"
REQUEST="check.txt"
RESPONSE="OK"

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

exit $EXIT_CODE
```

Notes:

EXIT_CODE="0" indicates success, EXIT_CODE="1" indicates failure

\$3 and \$4 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
- \$4 – the RIP port

Global Health Check Settings

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

Note:

For more details of these options, please refer to page [110](#).

HEALTH CHECKS FOR 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 WebUI menu option: *Cluster Configuration > Layer 7 - Virtual Services > Modify*

The exact options available depend on the check type selected. For the default (*Connect to port*), one field is required – *Check Port*.

Health Checks	Connect to port ▼	?
Check Port	<input type="text"/>	?

As the Check Type drop-down is changed, the field list changes. The full list of options is shown below:

As with Layer 4 Services, as the drop-down is changed, the field list changes.

Check Types:

Negotiate HTTP – Sends an HTTP request and looks for a specific response

Negotiate HTTPS – Sends an HTTPS request and looks for a specific response

Note:

If a Negotiate check is selected and Response Expected is left blank, the appliance will check the location specified in Request to Send (if blank the root will be checked) and look for a 200 OK response from the real server.

Connect to port – Just do a simple TCP connect to the specified port/service & verify that it's able to accept a connection

External Script – Use a custom external script for the health check. Specify the script in the drop-down

MySQL – Check MySQL by sending two MySQL packets, one Client Authentication packet, and one QUIT packet to correctly close the MySQL session

No checks, always On – All Real Servers are assumed to be up

Note:

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

Option	Sub Option	Health-Check Description
Negotiate HTTP		<i>Sends an HTTP request and looks for a specific response</i>
	Check Port	The port to monitor. This can normally be left blank in which case the port checked is the same port defined for the Real Servers. However, sometimes the check port differs from service port in which case it can be specified here. Also useful for multiport VIPs where the real server port field is left blank. In this case, the default checkport is the first in the list. This can be changed using this field if required.
	Request to Send	Specify a specific location/file for the health check. Open the

		specified location and check for the response expected. Useful for checking a server sided script to check the health of the backend application.
	Response Expected	<p>The content expected for a valid health check on the specified file. The response expected can be any valid regular expression statement.</p> <p>For example, if the Real Servers have a virtual directory called /customers, with a default page that contains the word 'welcome', Request to Send would be set to "customers" (without quotes) and Response Expected would be set to "Welcome" (without quotes). Provided that the load balancer can access the page and see the text 'Welcome', the health-check would pass.</p> <p>Note: If Response Expected is left blank, the appliance will check the location specified in Request to Send (if blank the root will be checked) and look for a 200 OK response from the real server.</p>
	Host Header	If the real server's web server is configured to require a Host header, the value to be used in health checks may be set here.
Negotiate HTTPS		<i>Sends an HTTPS request and looks for a specific response</i>
	Check Port	(see above)
	Request to Send	(see above)
	Response Expected	(see above)
	Host Header	(see above)
Connect to port		<i>Attempt to make a connection to the specified port</i>
	Check Port	(See above)
External Script		<i>Call an external script to perform the health-check</i>
	Check Script	<p>Select the required external check script from the drop-down. New scripts should be placed in /var/lib/loadbalancer.org/check, and given world read and execute permissions. The drop-down will be auto-updated to include any new scripts added.</p> <p>Note: By default the Microsoft SQL external health-check is not available in the drop down. This health-check requires several Microsoft related pre-requisites such as the Microsoft Linux ODBC driver, and these must first be installed and configured. To install the pre-requisites and configure required settings, at the console or via an SSH session, login as root and run the following command:</p>

		<p>\$ lb_mssql -i</p> <p>Once completed, the additional option "ms-sql-check" will appear in the External Script drop-down.</p>
MySQL		<i>Check MySQL</i>
	Check Port	(See above)
	Username	<p>perform a Client Authentication check, using <username> This requires an update into the MySql servers, as shown below, using MySQL client software:</p> <pre>USE mysql; INSERT INTO user (Host,User) values ('<ip_of_haproxy>','<username>'); FLUSH PRIVILEGES;</pre> <p>Note: without the user option, a MySql Handshake is performed</p>
No checks, always On		<i>No checking will take place and Real Servers will always be assumed to be up</i>

Custom External Script Example

The custom check script, used with the external check type. The script should be placed in **/var/lib/loadbalancer.org/check**, and given world read and execute permissions. The drop-down will be auto-updated to include any new scripts added.

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 HAProxy to control whether the server should be left online or removed.

```
#!/bin/bash
export PATH=/bin:/usr/bin:/sbin:/usr/sbin

# Variables
REALIP="$3"
PORT="$4"
REQUEST="check.txt"
RESPONSE="OK"

# Get the Page/File
wget -qO- --header="Host: host.domain.com" http://$REALIP:$PORT/$REQUEST |grep -qe $RESPONSE
if [ "$?" -eq "0" ]; then
EXIT_CODE="0"
```

```
else
EXIT_CODE="1"
fi

exit $EXIT_CODE
```

Notes:

EXIT_CODE="0" indicates success, EXIT_CODE="1" indicates failure

\$3 and \$4 are HAProxy variables that are passed to the script. The following HAProxy variables are available and can be used as required:

\$1 – the VIP address

\$2 – the VIP port

\$3 – the RIP address

\$4 – the RIP port

Note:

It's important that the commands are set the run in quiet mode, i.e. no output. Otherwise HAProxy may misinterpret the return data. This is achieved in the above example with -q options for the commands 'wget' and 'grep'.

Global Health Check Settings

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

Note:

For more details of these options, please refer to page [126](#).

Testing External Health-Check Scripts at the Command Line

All health-check scripts require 4 passed parameters. These 4 values represent *Virtual Service IP Address*, *Virtual Service Port*, *Real Server IP Address* and *Real Server Port*. If a script does not use all 4 values, for example the ping.sh script, then a zero should be entered as a place-holder.

```
# ./<check-script-name> <$1> <$2> <$3> <$4>
```

Examples:

```
# ./SMTP-check.sh 192.168.1.1 25 192.168.1.10 25
```

```
# ./ping.sh 192.168.1.1 0 192.168.1.10 0
```

to check the return value, use the command:

```
# echo $?
```

A return value of 0 means the check has passed, any other value means it has failed.

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 WebUI option *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`

Note:

Make sure these rules are cleared after testing & verification is complete!

Disabling Health-Checks

In some cases it may be desirable to completely disable health checking and simply assume that the real servers are up and working correctly. This can be configured simply by setting the health-check option to **No Checks, Always On** – this applies to both layer 4 and layer 7 services.

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 are unavailable and when all servers are taken offline via the WebUI. The fallback page can be hosted on the load balancer or on an external server. It can also be configured to be a Layer 7 VIP. Set the Fallback Server option of the VIP accordingly.

Layer 7

For layer 7 VIPs the fallback page is displayed when all Real Servers are unavailable and when all servers are taken offline via the WebUI. 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 WebUI menu option: *Maintenance > Fallback Page*

```

FALLBACK PAGE
1 <html>
2 <head>
3 <title>The page is temporarily unavailable</title>
4 <style>
5 body { font-family: Tahoma, Verdana, Arial, sans-serif; }
6 </style>
7 </head>
8 <body bgcolor="white" text="black">
9 <table width="100%" height="100%">
10 <tr>
11 <td align="center" valign="middle">
12 The page you are looking for is temporarily unavailable.<br/>
13 Please try again later.<br/>
14 (WUI port reminder 9080)
15 </td>
16 </tr>
17 </table>
18 </body>
19 </html>
20

```

Notes:

- 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

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 (port re-mapping is not possible with DR mode). Also, don't forget to solve the ARP problem for the dedicated fallback server (see page [79](#))
- 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.

Using a Layer 7 VIP as the fallback server for Layer 4 VIPs:

- It's possible to set the fallback server for a layer 4 VIP to be a layer 7 VIP. This is especially useful in WAN/DR site environments.
It also enables an external fallback server to be easily configured for Layer 4 VIPs – simply create a fallback VIP and configure the fallback server as an associated RIP, then enable the MASQ option for the Layer 4 VIP and set the fallback VIP as its fallback server. If all servers are down, requests will then be routed via the Layer 7 VIP to the external server. If the layer 4 VIP is multi-port, specify

0 as the port for the fallback server. Requests will then be forwarded to the correct port.

Setting the Fallback Server as one of the Real Servers:

- It's possible to configure one of the Real Servers as the fallback server. This can be useful for example when all servers are very busy and health checks start to fail simply because the response is taking longer than the configuration allows. In this case, traffic will still be sent to one of the Real Servers rather than to a separate fallback page.

Configuring Email Alerts

Email alerts can be configured for 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.

LAYER 4

At layer 4, settings can be configured globally that apply to all VIPs or individually to each VIP.

GLOBAL SETTINGS

Once configured, these settings apply to all layer 4 VIPs by default.

To configure global email alert settings for layer 4 services:

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 Advanced Configuration*

Lock Idirectord Configuration	<input type="checkbox"/>	?
Check Interval	<input type="text" value="5"/>	?
Check Timeout	<input type="text" value="3"/>	?
Negotiate Timeout	<input type="text" value="5"/>	?
Failure Count	<input type="text" value="2"/>	?
Quiescent	<input type="text" value="no"/>	?
Email Alert Source Address	<input type="text" value="lbmaster1@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"/>	?
		<input type="button" value="Update"/>

2. Enter an appropriate email address in the *Email Alert Source Address* field
e.g. lbmaster1@loadbalancer.org

3. Enter an appropriate email address in the *Email Alert Destination Address* field
e.g. **alerts@loadbalancer.org**
4. Click **Update**

Note:

Make sure that you also configure an SMTP smart host using the WebUI menu 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 SETTINGS

Once configured, these settings apply to individual VIPs.

To configure VIP level email alerts:

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 Advanced Configuration*
2. Enter an appropriate email address in the *Email Alert Source Address* field
e.g. **LB1@loadbalancer.org**
3. Using the WebUI, navigate to: *Cluster Configuration > Layer 4 Virtual Service* and click **Modify** next to the VIP to be configured

The screenshot shows a web form with a label 'Email Alert Destination Address' on the left. To its right is a text input field. On the far right of the input field is a small blue question mark icon. Below the input field, there are two buttons: a red 'Cancel' button and a green 'Update' button.

4. Enter an appropriate email address in the *Email Alert Destination Address* field
e.g. **alerts@loadbalancer.org**
5. Click **Update**

Note:

Make sure that you also configure an SMTP smart host using the WebUI menu 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.

LAYER 7

At layer 7, email settings must be configured globally rather than at the individual VIP level.

To configure global email alert settings for layer 7 services:

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 Advanced Configuration*

eMail Alert From	<input type="text"/>	?
eMail Alert To	<input type="text"/>	?
eMail Server Address	<input type="text"/>	?
eMail Server Port	<input type="text" value="25"/>	?
		<input type="button" value="Update"/>

2. Enter an appropriate email address in the *eMail Alert From* field
e.g. **lbmaster1@loadbalancer.org**
3. Enter an appropriate email address in the *eMail Alert To* field
e.g. **alerts@loadbalancer.org**
4. Enter an appropriate IP address/FQDN in the *eMail Server Address* field
e.g. **email.domain.com**
5. Enter an appropriate port in the *eMail Server Port* field
e.g. **25**
6. Click **Update**

Real Server Monitoring & Control using System Overview

REAL SERVER MONITORING

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

SYSTEM OVERVIEW ?							2015-04-21 10:36:58 UTC
VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE	
HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy	
RDS-Cluster	192.168.110.152	3389	0	TCP	Layer 7	Proxy	

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

SYSTEM OVERVIEW ? 2015-04-21 10:38:59 UTC

VIRTUAL SERVICE	IP	PORTS	CONN	PROTOCOL	METHOD	MODE	
 HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy	
<i>REAL SERVER</i>							
 IIS1	192.168.110.240	80	100	0	Drain	Halt	
 IIS2	192.168.110.241	80	100	0	Drain	Halt	
 RDS-Cluster	192.168.110.152	3389	0	TCP	Layer 7	Proxy	
<i>REAL SERVER</i>							
 RDS1	192.168.110.240	3389	100	0	Drain	Halt	
 RDS2	192.168.110.241	3389	100	0	Drain	Halt	

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)
- **Purple/Green** – Used to indicate that a particular VIP is used for HTTP to HTTPS redirection

This information is also displayed when clicking the system overview help button:

System Overview ✕

The following colors and icons are used to show the real-time status of your Load balanced Virtual Services

Colour	Image	Details
Green		Virtual Service / Real Server healthy
Yellow		Virtual Service needs attention
Blue		Real Server taken offline
Red		Virtual Service / Real Server down
Purple		Virtual Service FORCE-TO-HTTPS

The Virtual Services may be sorted using drag and drop, or by clicking on the column headings.

REAL SERVER CONTROL

The System Overview also enables the state of each Real Server to be controlled. Real Servers can be put in the following modes:

- **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

The screen shot below shows that RDS2 has been put into drain mode:

	RDS-Cluster	192.168.110.152	3389	0	TCP	Layer 7	Proxy	
	REAL SERVER	IP	PORTS	WEIGHT	CONNS			
	RDS1	192.168.110.240	3389	100	0	Drain	Halt	
	RDS2	192.168.110.241	3389	0	0	Online (drain)	Halt	

To bring RDS2 back online, click the *Online (drain)* link. If the server had been halted rather than drained, then the link would be displayed as *Online (Halt)*.

Note:

If a particular Real Server is used in multiple VIPs you can choose to apply the offline/online action to all relevant VIPs or only a single VIP. This simplifies taking Real Servers offline for maintenance purposes.

Note:

Halting or draining all Real Servers in a cluster activates the fallback server.

ORDERING OF VIPS

The display order of configured VIPs can be changed either by clicking on the column heading, or by drag and drop.

SORT BY COLUMN

If VIPs are ordered by a particular column, this is indicated using arrows next to the column heading as shown below:

SYSTEM OVERVIEW								2015-04-21 12:01:46 UTC
	VIRTUAL SERVICE ▼	IP ↕	PORTS ↕	CONNS ↕	PROTOCOL ↕	METHOD ↕	MODE ↕	
	HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy	
	RDS-Cluster	192.168.110.152	3389	0	TCP	Layer 7	Proxy	

In this example, the VIPs are ordered alpha-numerically by Virtual Service name. To change the order, click on the required column heading then click save. If you want to reverse the order for a particular column, click that column heading again.

e.g. Clicking on the IP column heading shows the following:

EDIT MODE Cancel Save

	VIRTUAL SERVICE	IP	PORTS	CONN	PROTOCOL	METHOD	MODE
	RDS-Cluster	192.168.110.152	3389	0	TCP	Layer 7	Proxy
	HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy

Clicking on the IP column heading again changes the order to:

EDIT MODE Cancel Save

	VIRTUAL SERVICE	IP	PORTS	CONN	PROTOCOL	METHOD	MODE
	HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy
	RDS-Cluster	192.168.110.152	3389	0	TCP	Layer 7	Proxy

Once you've set the required order, click **Save**.

DRAG & DROP

To re-order VIPs by drag and drop, simply click the mouse on any part of the VIP:

EDIT MODE Cancel Save

	VIRTUAL SERVICE	IP	PORTS	CONN	PROTOCOL	METHOD	MODE
	RDS-Cluster 	192.168.110.152	3389	0	TCP	Layer 7	Proxy
	HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy

Then drag it to the required position:

EDIT MODE Cancel Save

	VIRTUAL SERVICE	IP	PORTS	CONN	PROTOCOL	METHOD	MODE
	HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy
	RDS-Cluster 	192.168.110.152	3389	0	TCP	Layer 7	Proxy

And release it. Once you've set the required order, click **Save**.

Real Server Control using the HAProxy Statistics Page

It's also possible to control layer 7 Real Servers using the HAProxy statistics page. By default this is not enabled.

To enable this:

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Advanced*
2. Enable the **Advanced Stats** option as shown below:

HAProxy Statistics Page

Password:

Port: 7777

Advanced Stats:

3. Click **Update**
4. Reload HAProxy using the button at the top of screen
5. With this setting, the HAProxy stats page has the ability to control the state of real servers as shown below:

HAProxy
Statistics Report for pid 29981

> **General process information**

pid = 29981 (process #1, nproc = 1)
uptime = 0d 0h00m03s
system limits: memmax = unlimited; ulimit-n = 80033
maxsock = 80033; maxconn = 40000; maxpipes = 0
current conns = 3; current pipes = 0/0; conn rate = 3/sec
Running tasks: 1/8; idle = 100 %

active UP, active UP, going down, active DOWN, going up, active or backup DOWN, active or backup DOWN for maintenance (MAINT), active or backup SOFT STOPPED for maintenance, backup UP, backup UP, going down, backup DOWN, going up, not checked

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option: Scope: []
[Hide DOWN servers](#)
[Refresh now](#)
[CSV export](#)

External resources:
[Primary site](#)
[Updates \(v1.5\)](#)
[Online manual](#)

L7		Queue		Session rate		Sessions			Bytes		Denied		Errors		Warnings		Server																		
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle				
<input type="checkbox"/>	Frontend	0	0	-	0	0	40 000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	backup	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-	Y								
<input type="checkbox"/>	rip	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3s UP	L4OK in 0ms	100	Y	-	0	0	0	0	0	0	
<input type="checkbox"/>	Backend	0	0	-	0	0	4 000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	1	1	0	0	0	0	0	0	0	

Choose the action to perform on the checked servers: [] Apply

stats

	Queue	Session rate		Sess		
	Cur	Max	Limit	Cur	Max	Limit
Frontend	3	4	-	3	4	2 000
Backend	0	0	-	0	0	200

Set state to READY
Set state to DRAIN
Set state to MAINT
Health: disable checks
Health: enable checks
Health: force UP
Health: force NOLB
Health: force DOWN
Agent: disable checks
Agent: enable checks
Agent: force UP
Agent: force DOWN
Kill Sessions

6. Use the check-boxes to select the relevant Real Server(s), then select the required action in the drop-down, then click **Apply**

Chapter 9 – Appliance Clustering for HA

Introduction

Appliances can be deployed as single units or as a clustered pair.

Note:

We always recommend deploying 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 active unit (normally the master) handles all traffic under normal circumstances. If the active unit fails, the passive unit (normally the slave) becomes active and handles all traffic.

MASTER/SLAVE OPERATION

HEARTBEAT

By default, heartbeat uses ucast over UDP port 6694 to communicate between the master and slave appliances. The link enables the state of each to be monitored by the other and permits a failover to the passive unit if the active unit should fail. For hardware appliances, it's possible to configure both ucast and serial communication if required.

Note:

For hardware appliances, if the load balancer pair is located in close proximity, enabling serial communication in addition to ucast is recommended. Once the serial cable is connected between the appliances, serial comms can be enabled using the WebUI menu option: *Cluster Configuration > Heartbeat Configuration*. When serial communication is disabled, console access via the serial port is activated.

Ping checks to a common node such as the default gateway can also be configured. If the active node loses access to the ping node, the system will fail-over to the peer. However, if both nodes lose access, no fail-over will occur.

MASTER SLAVE REPLICATION

Once the master and slave are paired, 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. Note that replication of the configured load balanced services from the master to the slave appliance occurs over the network using SSH/SCP.

Settings that are NOT Replicated to the Slave Appliance

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

- Physical – Advanced Configuration settings including Internet Proxy IP address & port, Firewall table size, SMTP relay and Syslog server
- SNMP settings
- Graphing settings
- Firewall Script & Firewall Lockdown Script settings
- Software updates

High Availability Configuration

Units can be combined into a clustered pair for high-availability and resilience. Points to note:

- Pairing should be performed on the unit that is to be the master appliance
- The master and slave appliance must be able to perform an ICMP echo request (ping) to each other
- The master and slave appliance must be able to communicate with each other on TCP port 22
- The master and slave appliance must be able to communicate with each other on UDP port 6694 (or the selected custom port if this has been changed)

TO CREATE AN HA PAIR (ADD A SLAVE)

1. Power up a second appliance that will be the slave and configure initial network settings – please refer to chapter 4 starting on page [31](#) for more details on initial deployment and network setup
2. In the WebUI of the master appliance, navigate to: *Cluster Configuration > High-Availability Configuration*

CREATE A CLUSTERED PAIR

loadbalancer.org

Local IP address
192.168.110.40

IP address of new peer
192.168.110.41

Password for *loadbalancer* user on peer
.....

Add new node

3. Specify the IP address and the *loadbalancer* users password (the default is 'loadbalancer') for the slave (peer) appliance as shown above
4. Click **Add new node**
5. A warning will be displayed indicating that the pairing process will overwrite the new slave appliance's existing configuration, click **OK** to continue

- The pairing process now commences as shown below:

- Once complete, the following will be displayed:

- To finalize the configuration, restart heartbeat as prompted in the blue message box

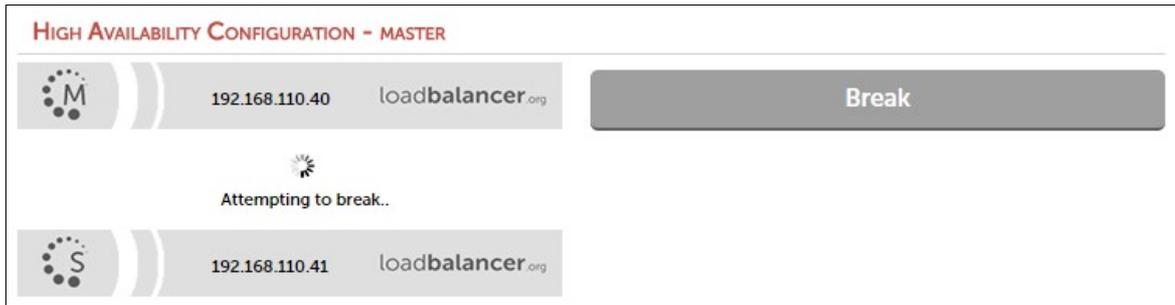
Note:

Clicking the **Restart Heartbeat** button on the master appliance will also automatically restart heartbeat on the slave appliance.

TO BREAK AN HA PAIR (REMOVE A SLAVE)

- In the WebUI of the master or slave appliance, navigate to: *Cluster Configuration > High-Availability Configuration*

- To break the pair, click the red **Break** button



- Once the process is complete, the pairing configuration screen will be displayed:

- To complete the reconfiguration, restart the system services on both appliances as directed in the blue message box

Notes:

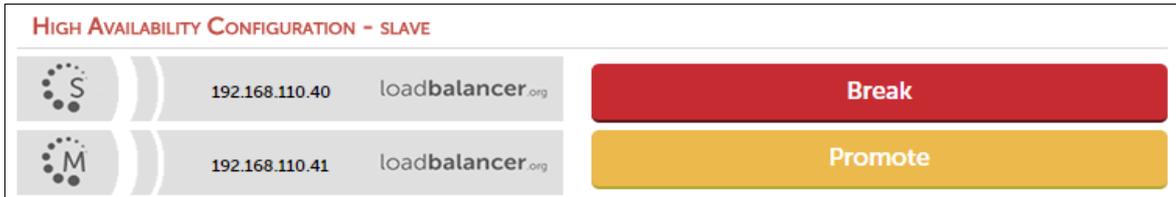
- Load balanced services will be momentarily interrupted as system services are restarted
- After the pair is broken, the slave will be left configured as a slave and any configured load balanced services will remain.
- If you later want to use the slave as a master, use the *Cluster Configuration > High Availability Configuration* menu option on the slave to setup a new pair. The slave will then be re-configured as a master, and the added peer will be configured as a slave.

Promoting a Slave to Master

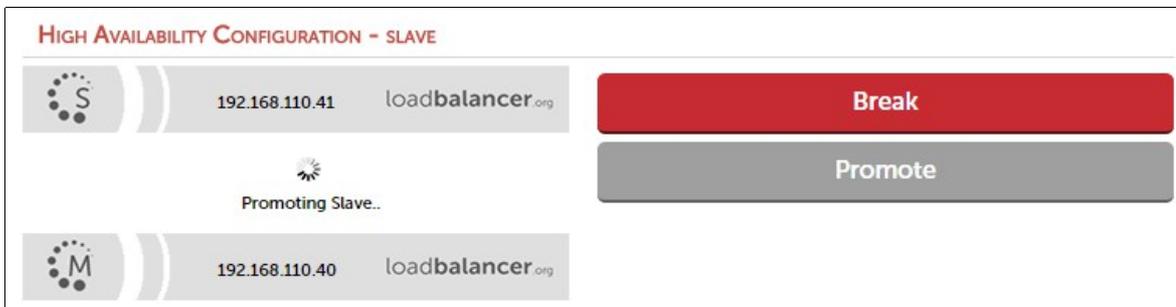
This is useful if the master unit fails and you'd like to change the now active slave to be a master, and then add the repaired/replaced master back as a slave unit.

To promote a slave unit to become a master:

1. In the WebUI of the slave appliance, navigate to: *Cluster Configuration > High-Availability Configuration*



2. Click **Promote**



Note:

If the master is still up and operational, it will not be possible to promote the slave.

3. Once complete, the unit will be configured as a master

Note:

Please refer to page [255](#) for details on how to recover from various failure scenarios.

Configuring Heartbeat

To configure Heartbeat:

1. In the WebUI of the Master appliance, navigate to: *Cluster Configuration > Heartbeat Configuration*

Note:

The screen shot below shows the configuration screen for a hardware appliance. The virtual appliance does not have the serial option checkbox in the communications method section.

Communication method		
UDP Unicast	<input checked="" type="checkbox"/>	?
UDP Broadcast <i>(Deprecated)</i>	Off ▾	?
UDP Port for broadcast & unicast	6694	?
Peer Failure Detection		
Keep-alive message interval	3 seconds	?
Dead peer timer	10 seconds	?
Warning timer	5 seconds	?
Routing Failure Detection		
Test IP addresses	<input type="text"/>	?
Test time-out	10 seconds	?
Email Alerts		
Email Alert Destination Address	<input type="text"/>	?
Email Alert Source Address	<input type="text"/>	?
Automatic Fail-back		
	<input type="checkbox"/>	?
Modify Heartbeat configuration		

Serial – Enable or disable heartbeat master/slave communication over the serial port. Ucast is the default heartbeat communication method. However, if the load balancer pair is located in close proximity, enabling serial communication in addition to ucast is recommended. This method requires a null modem cable (one 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. When serial communication is disabled, console access via the serial port is activated.

UDP Unicast – Enable or disable unicast heartbeat master/slave communication. This is the default method of heartbeat communication and uses unicast UDP between master and slave, with a destination port specified 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.

UDP Broadcast (Deprecated) – Enable or disable 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 specified by the *UDP Port for broadcast & unicast* parameter. Care must be taken when using broadcast on multiple pairs of load balancers in the same network. Each high-availability pair must operate on a different UDP port if they are not to interfere with each other. If heartbeat communication over the network is required, it is recommended that unicast be used in preference to broadcast.

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

Keep-alive message interval – Specify the number of seconds between keepalive pings. The Keepalive setting must be less than the warntime and deadtime.

Dead peer timer – The number of seconds communication can fail before a fail over is performed. A very low setting of deadtime could cause unexpected failovers.

Warning timer – 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.

Test IP address – Specify one or more 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 the active node loses access to the ping node, the system will fail-over to the peer. However, if both nodes lose access, no fail-over will occur. Multiple IP addresses may be given, separated by spaces or commas. In this case, if any one address is reachable the routing test will pass.

Test time-out - Specify the time-out, in seconds, for the routing test. If a response is not received from the test address within the time-out period, the route to that host will be considered dead.

Email Alert Destination Address – Specify the email address where to send heartbeat alerts. In the event of failover or failback the email address specified will receive an alert.

Email Alert Source Address – Specify the email address from where to send heartbeat alerts. In the event of failover failback the email specified will send an alert.

Note:

Both master and slave appliances will send an email in the event of a failover or failback.

Automatic Fail-back – Enable/disable auto-failback. 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 must then be controlled manually.

Note:

Automatic Fail-back is disabled by default. Manual intervention is required to force the repaired master to become active and the slave unit to return to passive mode. For more details refer to page [201](#). Auto fail-back can be enabled if required using the WebUI menu option: *Cluster Configuration > Heartbeat Configuration* and enabling **Automatic Fail-Back**.

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:

The screenshot shows the Load Balancer Enterprise VA MAX interface. At the top, the status is displayed as **Master | Slave**, **Active | Passive**, and **Link**. The system overview table shows the following data:

VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE
VIP1	192.168.110.45	80	0	HTTP	Layer 7	Proxy

This shows that the master unit is active and that the heartbeat link is up between master & slave. If no VIPs are defined, the status on master & slave appears as follows:

Master | Slave Active | Passive Link

Master | Slave Active | Passive Link

Other states:

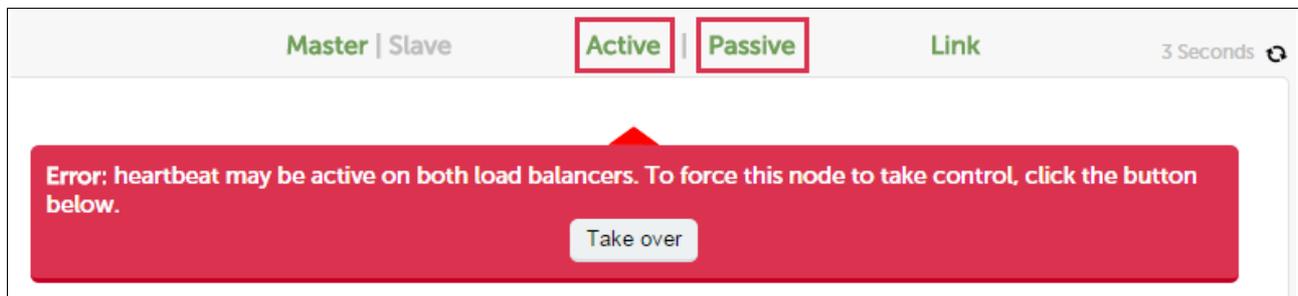
Master Slave	Active Passive	Link	this is a master unit, it's active, no slave unit has been defined
Master Slave	Active Passive	Link	this is a master unit, it's active, a slave has been defined but the link to the slave is down. Action: check & verify the heartbeat configuration
Master Slave	Active Passive	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

SPLIT BRAIN SCENARIOS

Split brain can occur if heartbeat on the master/slave clustered pair can no longer communicate with one another. In this case both units will bring up the Virtual Services and the system status will look similar to the following on both units:



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 automatically, the system status will show as follows on both units:

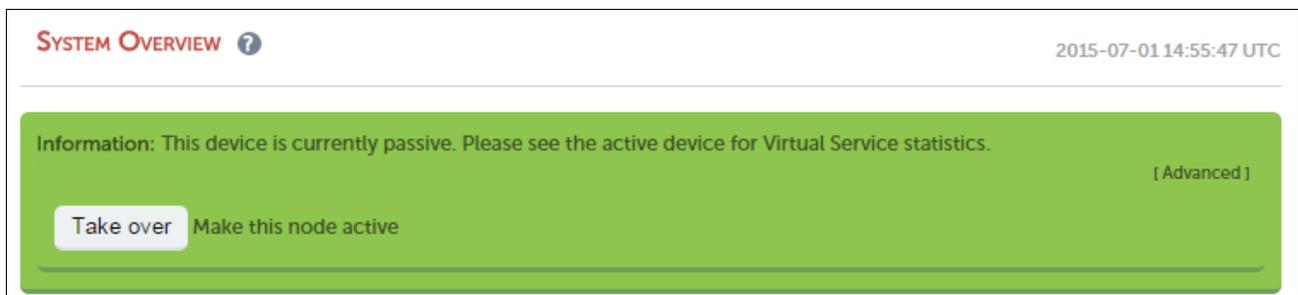


The **Take over** button can then be used on either master or slave to attempt to force that unit to become active.

FORCING MASTER/SLAVE FAILOVER & FAILBACK

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

Either use the **Take over** button in the slave's system overview:



Note:

Click the **[Advanced]** link to show this button.

Or 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:

Either use the **Take over** button on the master as explained above or run the following command on the master:

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

Note:

These commands can either be run on the console, via an SSH session or via the WebUI using: *Local Configuration > Execute Shell Command*.

TESTING & VERIFYING MASTER/SLAVE REPLICATION & FAILOVER

Note:

It's very important to verify that master/slave failover occurs correctly before going live. This proves the resilience of the HA cluster and makes you aware of the failover/failback process.

Note:

When testing appliance fail-over, if heartbeat is configured to use only 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. Testing must be done by pulling both the network and serial cable (if used) as detailed below.

STEP 1 - Verify Basic Settings for the clustered pair

1) On the master unit verify that the system status appears as follows:

Master		Slave	Active		Passive	Link
--------	--	-------	--------	--	---------	------

2) On the slave unit verify that the system status appears as follows:

Master		Slave	Active		Passive	Link
--------	--	-------	--------	--	---------	------

STEP 2 - Verify Replication

1) 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.

STEP 3 - Verify Failover to the Slave (using the Take over button)

1) On the slave unit, click the **[Advanced]** option in the green information box, then click the **Take Over** button

2) Verify that the slave's status changes to *Active*:



3) Verify that the master's status changes to *Passive*:



4) Using the WebUI menu 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

e.g. the partial screen shot below from the View Network Configuration screen on the slave unit shows the status of eth0:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:92:18:4f brd ff:ff:ff:ff:ff:ff
    inet 192.168.111.223/18 brd 192.168.127.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet 192.168.111.72/18 brd 192.168.127.255 scope global secondary eth0
        valid_lft forever preferred_lft forever
```

This shows the secondary IP address 192.168.111.72 (the VIP address) is up and therefore the slave has become active as intended.

STEP 4 - Verify Fallback to the Master (using the Take over button)

1) On the master unit, click the **[Advanced]** option in the green information box, then click the **Take Over** button

2) Verify that the master's status has changed to *Active*:



3) Verify that the slave's status has changed to *Passive*:



4) Also, using the WebUI menu 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 (see STEP 3 above for more details)

STEP 5 - Verify Failover to the Slave (when removing the network and serial cable from master)

- 1) Remove the network cable and serial cable (if applicable) from the master
- 2) verify that the slave's status has changed as follows:



This indicates that the slave is unable to communicate with the master. This means that either the master is down, or is still up but is unreachable. In both cases the slave will go active.

- 3) On the slave using the WebUI menu option: *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up (see STEP 3 above for more details)

STEP 6 - Verify normal operation resumes (when reconnecting the network & serial cable to master)

- 1) Reconnect the cables to the master
- 2) Verify that the master's status is set to *Active*:



- 3) Verify that the slave has changed to *Passive*:



- 4) Also, using the WebUI menu option: *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up on the master unit and brought down on the slave

Note:

If the power cable on the master had been removed rather than disconnecting the network cable and serial cable (if applicable), once the master is brought back up the slave would remain active and the master would come back up in a passive state. The **Take over** button on the master would then need to be used to force the master to become active.

Chapter 10 – 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.

Note:

Since the VIP is auto-configured for multi-port operation, ensure the checkport is set manually as shown in the image below (typically port 21).

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	?
	Check Port	21	?
	Protocol	FTP	?
	Login	health	?
	Password	*****	?
	Request to send	check.txt	?
	Response expected	OK	?

Key Points:

- Change the *Check Type* to **Negotiate**
- Ensure the *Check Port* is set to **21**
- 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 page [147](#).

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)

Note:

These rules can be added to the firewall script using the WebUI menu 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

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

Label	FTP-ClusterACTV		?
Virtual Service	IP Address	192.168.2.150	?
	Ports	21	?
Layer 7 Protocol	TCP Mode		?
TCP Keep-alive	<input type="checkbox"/>		?
Balance Mode	Weighted Least Connections		?
Persistence Mode	Source IP		?

2. 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	100		?

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

7. Configure the default gateway on each FTP server to be the load balancer. Ideally this should be a floating IP address to allow it to move between the master & slave appliance. This can be added using the WebUI menu option: *Cluster Configuration > Floating IPs*
8. 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 [211](#) covers this for a range of OS's & FTP servers.

Note:

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

Passive Mode – Key Points:

- It's sensible to use a controlled passive port range, this can be configured on the FTP server
- 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 'TCP Mode'
- If transparency is required (for passive mode this is optional), enable TProxy using the WebUI menu option: *Cluster Configuration > Layer 7 – Advanced Configuration*

Note:

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 [147](#) for more details on using TProxy).

- Using the WebUI menu option: *Cluster Configuration > Layer 7 – Advanced Configuration* increase the default *Client timeout & Real Server Timeout* to **5m** (i.e. 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

1. 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	192.168.2.150	?
	Ports	21,50000-50100	?
Layer 7 Protocol	TCP Mode		?
TCP Keep-alive	<input type="checkbox"/>		?
Balance Mode	Weighted Least Connections		?
Persistence Mode	Source IP		?

2. Configure the VIP to listen on both the control port (21) and passive range (e.g. 50000-50100) as shown
3. 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	100		?

4. Set *Client Timeout* and *Real Server timeout* to **5m** (i.e. 5 minutes) using the WebUI menu option: *Cluster Configuration > Layer 7 – Advanced Configuration*
5. Now restart HAProxy using the WebUI menu option: *Maintenance > Restart Services*
6. 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 in the example 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:

 Example: 5000-6000

External IP Address of Firewall:

 Example: 10.0.0.1

Note:

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.

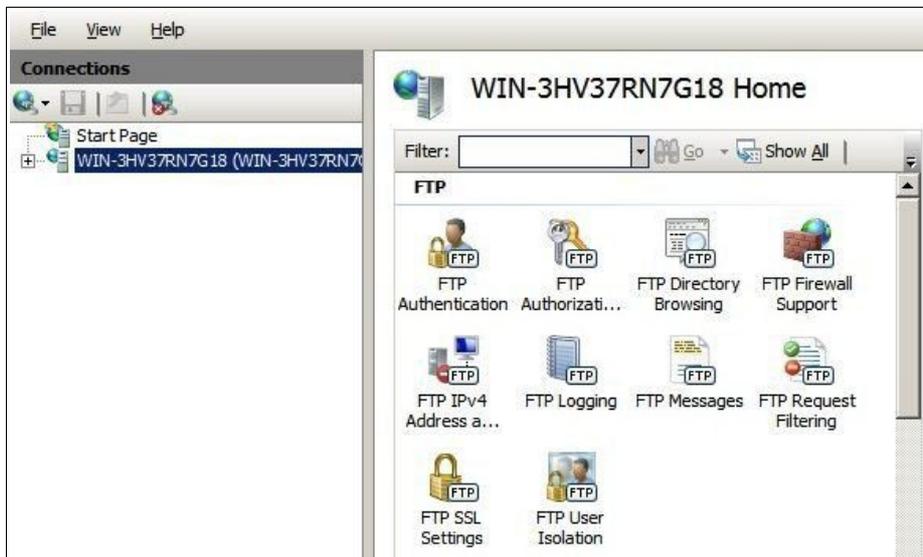
7. If TProxy is enabled, make sure the gateway of each FTP server is set to be an IP on the load balancer (preferably a floating IP to allow failover to the slave unit)
8. Now restart both IIS **and** the Microsoft FTP Service on each FTP server
9. Passive FTP clients should now be able to connect to 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:



Specify a suitable range, in the example above this is 50000-50100

IMPORTANT! - Make sure you restart IIS and 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 check-box** 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 (SP4 and later)

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

Note:

The range that FTP will validate is from 5001 to 65535.

For Linux

For **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
```

For **proftpd**, the following line can be added to the proftpd.conf file to limit the port range:

```
PassivePorts 50000 – 50100
```

For **pureftpd**, the following startup switch can be used:

```
-p --passiveportrange <min port:max port>
```

Terminal Services/Remote Desktop Services

LAYER 4 – IP PERSISTENCE

RDP is a TCP based service usually on port 3389. Clients will need to be sent to the same server to allow reconnection to existing sessions. The persistence setting should be changed to suit your requirements. A typical setting to use is *persistence* = 7200 (i.e. 7200s = 2 hours). This means that when a client reconnects within this time, they will be sent to the same Terminal Server/Remote Desktop 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	RDP-Cluster		?
Virtual Service	IP Address	192.168.10.20	?
	Ports	80	?
Protocol	TCP		?
Forwarding Method	Direct Routing		?
Balance Mode	Weighted Least Connection		?
Persistent	<input checked="" type="checkbox"/>		?
	Timeout	7200 seconds	?

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 server. Simply create Layer 7 VIP as shown below:

Label	RDP-Cluster		?
Virtual Service	IP Address	192.168.10.20	?
	Ports	3389	?
Layer 7 Protocol	TCP Mode		?
TCP Keep-alive	<input type="checkbox"/>		?
Balance Mode	Weighted Least Connections		?
Persistence Mode	MS Session Broker		?

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 (msthash). 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. If the cookie is not found, it will fallback to source IP persistence.

Label	RDP-Cluster		?
Virtual Service	IP Address	192.168.10.20	?
	Ports	3389	?
Layer 7 Protocol	TCP Mode		?
TCP Keep-alive	<input checked="" type="checkbox"/>		?
Balance Mode	Weighted Least Connections		?
Persistence Mode	RDP Client Cookie		?
Persistence	Timeout	120	?

Again, persistence can be set as required, but as per the previous example 2 hours (120m) has been configured in the example above.

Initial connections are distributed to the Real Servers based on the balance mode selected (defaults to weighted least connection). Re-connecting clients 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:

For additional information, please refer to the following Deployment Guides:

[Remote Desktop Services Deployment Guide](#)

[Terminal Services Deployment Guide](#)

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 [14](#) earlier in this manual.

Note:

Don't hesitate to contact support@loadbalancer.org for advice on load balancing your application if it's not listed.

Chapter 11 – 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 WebUI please refer to page [36](#) and page [39](#).

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 (backend) 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

Note:

This step can be skipped if all network settings have already been configured.

Configure the various network settings as outlined below:

1. Using the WebUI, navigate to: *Local Configuration > Network Interface Configuration*



2. 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**
3. Click **Configure Interfaces**

- Using the WebUI, navigate to: *Local Configuration > Hostname & DNS*
- Specify the DNS server(s)

Domain Name Server	Primary	<input type="text" value="192.168.2.254"/>	
	Secondary	<input type="text"/>	
	Tertiary	<input type="text"/>	

- Click **Update**
- Using the WebUI, navigate to: *Local Configuration > Routing*

Default Gateway				
IP v4	<input type="text" value="192.168.2.254"/>	via interface	<input type="text" value="auto"/>	
IP v6	<input type="text"/>	via interface	<input type="text" value="auto"/>	

- 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 forwarded to the the Real Servers associated with the Virtual Service.

- Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Virtual Services* and click **Add a new Virtual Service**

Label	<input type="text" value="ExVIP1"/>		
Virtual Service	IP Address	<input type="text" value="192.168.2.150"/>	
	Ports	<input type="text" value="80"/>	
Protocol	<input type="text" value="TCP"/>		
Forwarding Method	<input type="text" value="Direct Routing"/>		

- Enter a suitable Label (name) for the VIP, e.g. **ExVIP1**
- Enter a valid IP address, e.g. **192.168.2.150**
- Enter a valid port, e.g. **80**
- Ensure that *Forwarding Method* is set to **Direct Routing**

Note:

DR mode this is the default.

REAL SERVERS (RIPS)

Each Virtual Service requires a cluster of Real Servers (backend servers) to forward the traffic to.

- Using the WebUI, navigate to: *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="192.168.2.151"/>	?
Weight	<input type="text" value="100"/>	?
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. **192.168.2.151**

Note:

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 100 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

PHYSICAL 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 – only the load balancer must respond to these requests

Note:

Failure to correctly configure the Real Servers to handle the ARP problem is the most common problem in DR configurations. Please refer to page [79](#) for more details.

BASIC TESTING & VERIFICATION

Once configured, a few quick checks can be performed to verify the setup:

1. Using *System Overview* check that the VIP & RIPv are shown as active (green)
2. Using a browser, navigate to the VIP address, i.e. <http://192.168.2.150> to verify that you can reach the Real Servers via the Virtual Service
3. Check *Reports > Layer 4 Current Connections* to ensure that client connections are reported in state 'ESTABLISHED'. If connections are in state 'SYN_RECV', this normally indicates that the ARP problem on the Real Servers has not been correctly solved

2 – Two-Arm NAT Mode (Clustered Pair)

This example covers the process of configuring two load balancers (as a clustered pair) in layer 4 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.

Note:

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 (RIPv)** – 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

1. Using the WebUI on the master unit, navigate to: *Local Configuration > Network Interface Configuration*

IP Address Assignment

eth0 10 GB/s
eth1 10 GB/s
eth2
eth3

eth0: 192.168.2.120/24
MTU: 1500 bytes

eth1: 192.168.20.120/24
MTU: 1500 bytes

- 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. **192.168.20.120/24**

Note:

For a VA make sure that the virtual NIC associated with eth1 is connected to the virtual switch. By default only the first NIC is connected.

- Click **Configure Interfaces**

SLAVE UNIT – NETWORK SETTINGS

Configure the various network settings as outlined below:

- Using the WebUI on the slave appliance, navigate to: *Local Configuration > Network Interface Configuration*

IP Address Assignment

eth0 10 GB/s
eth1 10 GB/s
eth2
eth3

eth0: 192.168.2.121/24 MTU 1500 bytes

eth1: 192.168.20.121/24 MTU 1500 bytes

2. 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**
3. Specify the IP address & mask for eth1 – normally eth1 is configured as the *external* interface although this is not mandatory, e.g. **192.168.20.121/24**
4. Click **Configure Interfaces**

CREATING THE HA CLUSTERED PAIR

1. Using the WebUI on the master unit, navigate to: *Cluster Configuration > High Availability Configuration*

CREATE A CLUSTERED PAIR

loadbalancer.org

Local IP address: 192.168.2.120

IP address of new peer: 192.168.20.120

Password for loadbalancer user on peer:

Add new node

2. Leave the *Local IP address* set as the address assigned to eth0 , in this case **192.168.2.120**
3. Specify the *IP address of new peer* (i.e. the slave appliance) , in this case **192.168.2.121**
4. Specify the *loadbalancer* users password (the default is 'loadbalancer') for the slave appliance
5. Click **Add new node**
6. A warning will be displayed indicating that the pairing process will overwrite the new slave appliance's existing configuration, click **OK** to continue

7. The pairing process will start as shown below:

CREATE A CLUSTERED PAIR

 <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc; display: inline-block;"> 192.168.2.120 loadbalancer.org </div>	<p>Local IP address</p> <div style="border: 1px solid #ccc; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> 192.168.2.120 ▼ </div>
<p>Attempting to pair..</p>	<p>IP address of new peer</p> <div style="border: 1px solid #ccc; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> 192.168.2.121 </div>
 <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc; display: inline-block;"> 192.168.2.121 loadbalancer.org </div>	<p>Password for <i>loadbalancer</i> user on peer</p> <div style="border: 1px solid #ccc; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> </div>
<div style="background-color: #808080; color: white; padding: 10px 40px; border: 1px solid #ccc; display: inline-block; font-weight: bold;">configuring</div>	

8. Once completed successfully, the following messages will be displayed:

Commit changes

The configuration of the following services has been changed. When reconfiguration is complete, restart/reload the services to commit the changes

Restart Heartbeat

HIGH AVAILABILITY CONFIGURATION - MASTER

 <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc; display: inline-block;"> 192.168.2.120 loadbalancer.org </div>	<div style="background-color: #c00000; color: white; padding: 10px; border: 1px solid #ccc; display: inline-block; font-weight: bold;">Break Clustered Pair</div>
 <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc; display: inline-block;"> 192.168.2.121 loadbalancer.org </div>	<div style="background-color: #ffc000; padding: 10px; border: 1px solid #ccc; display: inline-block; font-weight: bold;">Make Active</div>

9. To finalize the configuration, click **Restart Heartbeat**

CHECKING THE STATUS

A successfully configured clustered pair will display the following status:

On the Master unit:

Master | Slave
Active | Passive
Link

On the Slave unit:

Master | **Slave**
Active | Passive
Link

Note:

Once the VIP has been defined, the Active text will be colored green on the master and Passive will be colored green on the slave.

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 forwarded to the the Real Servers associated with the Virtual Service.

- Using the WebUI, navigate to: *Cluster Configuration > Layer 4 – Virtual Services* and click **Add a new Virtual Service**

Label	<input type="text" value="ExVIP2"/>	?	
Virtual Service	IP Address	<input type="text" value="192.168.2.150"/>	?
	Ports	<input type="text" value="80"/>	?
Protocol	<input type="text" value="TCP"/>	?	
Forwarding Method	<input type="text" value="NAT"/>	?	
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>	

- Enter a suitable label (name) for the VIP, e.g. **ExVIP2**
- Enter a valid IP address, e.g. **192.168.2.150**
- 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 it to the slave

REAL SERVERS (RIPS)

Each Virtual Service requires a cluster of Real Servers (backend servers) to forward the traffic to.

- Using the WebUI, navigate to: *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="192.168.2.151"/>	?
Real Server Port	<input type="text" value="80"/>	?
Weight	<input type="text" value="100"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?
		<input type="button" value="Cancel"/> <input type="button" value="Update"/>

- Enter a suitable Label (name) for the RIP, e.g. **RIP1**

3. Enter a valid IP address, e.g. **192.168.2.151**
4. Enter a valid port, e.g. **80**
5. *Weight* defaults to 100 making the Real Server active immediately
6. Leave *Minimum Connections* & *Maximum Connections* set to 0 which means unrestricted
7. Click **Update**, this will save the RIP locally and also replicate IT to the slave
8. Repeat for the remaining Real Servers

PHYSICAL 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*.



The screenshot shows a web form titled "New Floating IP". On the left, there is a label "New Floating IP". To its right is a text input field containing the IP address "192.168.2.254". In the bottom right corner of the form, there is a green button labeled "Add Floating IP".

Define the IP address that you'd like to use for the default gateway, then click **Add Floating IP**. 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 WebUI on the slave and navigate to: *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 navigate to: *Maintenance > Backup & Restore* and click **Synchronize 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:

1. On the master, use *System Overview* to check that the VIP & RIPs are shown as active (green)
2. Using a browser, navigate to the VIP address, i.e. **http://192.168.2.150** to verify that you can reach the Real Servers via the Virtual Service
3. On the master, check *Reports > Layer 4 Current Connections* to ensure that client connections are reported in state '**ESTABLISHED**'. If not and instead '**SYN-RECV**' shown is double-check that you have set the default gateway on all Real Servers to be the floating IP address on the load balancer.

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.

Note:

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.

Note:

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

Note:

In this mode, no changes are required to the Real Servers.

Note:

We generally recommend that SSL is terminated on the real serves 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:

1. Using the WebUI, navigate to: *Local Configuration > Network Interface Configuration*

IP Address Assignment

eth0 10 GB/s

eth1

eth2

eth3

eth0: 192.168.2.120/24

MTU: 1500 bytes

2. 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**
3. Click **Configure Interfaces**
4. Using the WebUI, navigate to: *Local Configuration > DNS & Hostname*
5. Specify the DNS server(s)

Hostname: lbmaster

Domain Name: localhost

Role: master

Domain Name Server:

- Primary: 192.168.2.254
- Secondary:
- Tertiary:

Update

6. Click **Update**
7. Using the WebUI, navigate to: *Local Configuration > Routing*

Default Gateway

IP v4: 192.168.2.254 via interface: auto

IP v6: via interface: auto

8. Specify the Default Gateway, e.g. 192.168.2.254
9. 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.

1. Using the WebUI, navigate to: *Cluster Configuration > Layer 7 – Virtual Services* and click **Add a**

new Virtual Service

Label	<input type="text" value="ExVIP3"/>	?	
Virtual Service	IP Address	<input type="text" value="192.168.2.150"/>	?
	Ports	<input type="text" value="80"/>	?
Layer 7 Protocol	<input type="text" value="HTTP Mode"/>		?
Manual Configuration	<input type="checkbox"/>	?	

2. Enter a suitable Label (name) for the VIP, e.g. **ExVIP3**
3. Enter a valid IP address, e.g. **192.168.2.150**
4. Enter a valid port, e.g. **80**
5. Click **Update**

REAL SERVERS (RIPS)

Each Virtual Service requires a cluster of Real Servers (backend servers) to forward the traffic to.

1. Using the WebUI, navigate to: *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="192.168.111.151"/>	?
Real Server Port	<input type="text" value="80"/>	?
Weight	<input type="text" value="100"/>	?

2. Enter a suitable Label (name) for the RIP, e.g. **RIP1**
3. Enter a valid IP address, e.g. **192.168.2.151**

Note:

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.

4. Enter a valid port, e.g. **80**
5. The *Weight* defaults to 100 making Real Servers active as soon as HAProxy is restarted
6. Click **Update**
7. Repeat for the remaining Real Servers
8. Restart HAProxy to apply the new settings using the link provided in the blue box

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.

- Using the WebUI, navigate to: *Cluster Configuration > SSL Termination* and click **Add a new Virtual Service**

Label	<input type="text" value="ExSSL"/>	?
Virtual Service IP address	<input type="text" value="192.168.2.150"/>	?
Virtual Service Port	<input type="text" value="443"/>	?
Backend Virtual Service IP Address	<input type="text" value="192.168.2.150"/>	?
Backend Virtual Service Port	<input type="text" value="80"/>	?
Ciphers to use	<input type="text" value="ECDHE-RSA-AES128-GCM"/>	?
Do not insert empty fragments	<input checked="" type="checkbox"/>	?
SSL Terminator	<input type="radio"/> Pound <input checked="" type="radio"/> STunnel	?
Delay DNS Lookups	<input checked="" type="checkbox"/>	?
Disable SSLv2 Ciphers	<input checked="" type="checkbox"/>	?
Disable SSLv3 Ciphers	<input checked="" type="checkbox"/>	?
Allow Client Renegotiation	<input checked="" type="checkbox"/>	?
Disable SSL Renegotiation	<input checked="" type="checkbox"/>	?
Time To Close	<input type="text" value="0"/>	?
Set as Transparent Proxy	<input type="checkbox"/>	?

- Enter a suitable Label (name) for the VIP, e.g. **ExSSL**
- Set *Virtual Service IP address* to be the same as the layer 7 VIP created earlier, i.e. **192.168.2.150**
- 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. **192.168.2.150**
- 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.

Note:

For more detailed information on SSL termination please refer to page [130](#).

BASIC TESTING & VERIFICATION

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

1. Using *System Overview*, verify that the VIP & RIP are shown as active (green)
2. Using a browser, navigate to the VIP address, i.e. **http://192.168.2.150** to verify that you can reach the Real Servers via the Virtual Service using HTTP
3. Using a browser, navigate to the STunnel SSL VIP address, i.e. **https://192.168.2.150** to verify that you can reach the Real Servers via the Virtual Service using HTTPS
4. Check/verify the certificate details

Chapter 12 – 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://192.168.110.10>

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.

DIAGNOSING VIP CONNECTION PROBLEMS

1. **Make sure that the appliance is active** - this can be checked in the WebUI. For a single appliance, the status bar should report **Master & Active** as shown below:



2. **Check that the VIP/floating IP is up** - Using *View Configuration > Network Configuration* verify that the VIP is active on the load balancer, if not check *Logs > Heartbeat* for errors.

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:cf:18:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.110.85/18 brd 192.168.127.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet 192.168.110.90/18 brd 192.168.127.255 scope global secondary eth0
        valid_lft forever preferred_lft forever
```

The above example shows that the interface address (192.168.110.85) and the VIP address (192.168.110.90) are both up.

3. **Check that the Real Servers are up** - Using *System Overview* make sure that none of your VIPs are colored red. If they are, the entire cluster is down (i.e. all 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 all Real Server have been deliberately taken offline (by using either the *Halt* or *Drain* options next to each Real Server).

SYSTEM OVERVIEW ?							2015-03-18 11:37:15 UTC
VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE	
↑ HTTP-Cluster	192.168.110.150	80	0	TCP	Layer 4	DR	
⚠ RDP-Cluster	192.168.110.150	3389	0	TCP	Layer 4	DR	
↓ HTTP-Cluster-2	192.168.110.152	80	0	HTTP	Layer 7	Proxy	
⚙ RDP-Cluster-2	192.168.110.152	3389	0	TCP	Layer 7	Proxy	

4. **Check the connection state**

For Layer 4 DR mode VIPs check *Reports > Layer 4 Current Connections* to view the current traffic in detail. Any packets with state **SYN_RECV** imply that the 'ARP Problem' has not been correctly solved on the Real Servers. See page [79](#) for more details on solving the ARP problem.

For layer 4 NAT mode VIPs check *Reports > Layer 4 Current Connections* to view the current traffic in detail. Any packets with state **SYN_RECV** often imply that the default gateway on the Real Servers has not been 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/status report as shown in the example below:

Statistics Report for pid 3261

> General process information

pid = 3261 (process #1, nproc = 1)
 uptime = 0s 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: 1/6; 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

Display option:
 • [Hide 'DOWN' servers](#)
 • [Refresh now](#)
 • [CSV export](#)

External resources:
 • [Primary site](#)
 • [Updates \(v1.5\)](#)
 • [Online manual](#)

Note: UP with load-balancing disabled is reported as "NOLB".

L7		Queue		Session rate			Sessions			Bytes		Denied		Errors			Warnings		Server												
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend		0	15	-	0	4	40 000	56			21 698	3 385 782	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	21 698	3 385 782	0	0	0	0	0	0	0	42s UP	L4OK in 0ms	1	Y	-	0	0	0s	-	
Backend		0	0		0	16	0	2	4 000	56	56	21 698	3 385 782	0	0	0	0	0	0	0	0	42s UP		1	1	1		0	0s		

stats		Queue		Session rate			Sessions			Bytes		Denied		Errors			Warnings		Server												
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend		2	4	-	1	1	2 000	8			1 464	33 111	0	0	4							OPEN									
Backend		0	0		0	0	0	0	200	0	0	0	1 464	33 111	0	0	0	0	0	0	0	42s UP		0	0	0		0	0s		

TAKING REAL SERVERS OFFLINE

1) Using the *System Overview* check that when you Halt one of the Real Servers the connections are redirected to the other server in the cluster.

2) 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). Also check that the server is shown red (down) in the system overview.

3) 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. Also check that the server is shown green (up) in the system overview.

The *System Overview* will also show the updated status as these tests are performed:

SYSTEM OVERVIEW  2015-04-30 08:35:41 UTC

	VIRTUAL SERVICE	IP	PORTS	CONNNS	PROTOCOL	METHOD	MODE	
	HTTP-Cluster	192.168.110.150	80	0	HTTP	Layer 7	Proxy	
	<i>REAL SERVER</i>	<i>IP</i>	<i>PORTS</i>	<i>WEIGHT</i>	<i>CONNNS</i>			
	RIP1	192.168.110.240	80	100	0	Drain	Halt	
	RIP2	192.168.110.241	80	0	0	Online (halt)		
	RIP3	192.168.110.242	80	100	0	Drain	Halt	

In this example:

RIP1 is green, this indicates that it's operating normally.

RIP2 is blue, this indicates that it has been either Halted or Drained. in this example Halt has been used as indicated by *Online (Halt)* being displayed. If it had been drained it would show as *Online (Drain)*.

RIP3 is red, this indicates that it has failed a health check.

Note:

The System Overview supports sorting of VIPs. This can be done by clicking on the column headings or by drag & drop. For more details please refer to page [189](#).

USING LOG FILES

The appliance includes several log files that are very useful when diagnosing issues. Please refer to the next chapter 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 for more details on the reports available.

Note:

When testing a clustered pair, also make sure that failover to the slave appliance and failback to the master appliance is working correctly. For more details please refer to page [202](#).

Chapter 13 – Appliance Monitoring

Appliance Log Files

All appliance logs can be accessed using the *Logs* option in the WebUI.

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.

APACHE ERROR LOG

File: */var/log/httpd/error.log*

Shows Apache errors. These can be generated by the WebUI and WAF (Web Application Firewall).

APACHE USER LOG

File: */var/log/httpd/user_access.log*

Shows Apache user access logs. Can be generated by WebUI and the WAF (Web Application Firewall) since both utilize Apache for their operation.

WAF LOGS

Various log file for monitoring WAFs.

Appliance Reports

All reports can be accessed using the *Reports* option in the WebUI.

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.

<input type="button" value="Check Status"/>					
Virtual Service	Real Server	Forwarding Method	Weight	Active Connections	Inactive Connections
HTTP-Cluster1 192.168.110.120 port 80/tcp					
	RIP1 192.168.110.240	Route	100	0	0
	RIP2 192.168.110.241	Route	100	0	0
	RIP3 192.168.110.242				


```

IP Virtual Server version 1.2.1 (size=32768)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port      Forward Weight ActiveConn InActConn
TCP  192.168.110.120:80 wlc persistent 300
  -> 192.168.110.240:80      Route  100    0        0
  -> 192.168.110.241:80      Route  100    0        0
  
```

In the example above, the details for RIP3 are not displayed because it's failing its health checks.

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
HTTP-Cluster1 192.168.110.120 port 80/tcp	RIP1 192.168.110.240	0	0	0	0	0
	RIP2 192.168.110.241	0	0	0	0	0
	RIP3 192.168.110.242					
IP Virtual Server version 1.2.1 (size=32768)						
Prot	LocalAddress:Port	CPS	InPPS	OutPPS	InBPS	OutBPS
	-> RemoteAddress:Port					
TCP	192.168.110.120:80	0	0	0	0	0
	-> 192.168.110.240:80	0	0	0	0	0
	-> 192.168.110.241:80	0	0	0	0	0

In the example above, the details for RIP3 are not displayed because it's failing its health checks.

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.

Check Status
Reset Counters

Virtual Service	Real Server	Connections	Incoming Packets	Outgoing Packets	Incoming Bytes	Outgoing Bytes
HTTP-Cluster1 192.168.110.120 port 80/tcp		0	0	0	0	0
	RIP1 192.168.110.240	0	0	0	0	0
	RIP2 192.168.110.241	0	0	0	0	0
	RIP3 192.168.110.242					


```

IP Virtual Server version 1.2.1 (size=32768)
Prot LocalAddress:Port          Conns  InPkts  OutPkts  InBytes  OutBytes
-> RemoteAddress:Port
TCP  192.168.110.120:80           0      0      0      0      0
-> 192.168.110.240:80         0      0      0      0      0
-> 192.168.110.241:80         0      0      0      0      0
  
```

Note:

These reports are generated in real time. Direct Routing is the default load balancing method and you will not see any stats for return packets as 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.

In the example above, the details for RIP3 are not displayed because it's failing its health checks.

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**. For layer 4 DR mode this is normally a good indication that the ARP problem has not been solved. For 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.

Check Status

```

IPVS connection entries
pro expire state      source                virtual                destination
TCP 04:44  NONE                192.168.64.7:0        192.168.110.120:80    192.168.110.241:80
TCP 00:49  SYN_RECV            192.168.64.7:28808    192.168.110.120:80    192.168.110.241:80
TCP 00:49  SYN_RECV            192.168.64.7:28809    192.168.110.120:80    192.168.110.241:80
    
```

Note:

The IPVS connection entries in state **NONE** represent the persistence related entries for client connections, and are not actual client connections. These only appear when persistence is enabled.

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

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

Display option:
 • [Hide 'DOWN' servers](#)
 • [Refresh now](#)
 • [CSV export](#)

External resources:
 • [Primary site](#)
 • [Updates \(v1.5\)](#)
 • [Online manual](#)

Note: UP with load-balancing disabled is reported as "NOLB".

L7-HTTP																																
		Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Server											
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle		
Frontend		0	0	-	0	0	-	0	0	40 000	0	0	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			1	-	Y						
rip1		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	0	0s	-	
Backend		0	0	-	0	0	-	0	0	4 000	0	0	0	0	0	0	0	0	0	0	0	22s UP		1	1	1		0	0	0s		

stats																																
		Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Server											
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle		
Frontend		2	2	-	2	2	-	2	2	2 000	5	0	1 406	20 676	0	0	0	0	0	0	0	22s UP										
Backend		0	0	-	0	0	-	0	0	200	0	0	1 406	20 676	0	0	0	0	0	0	0	22s UP		0	0	0						

Note:

This password can be changed using the 'statistics password' field available under Cluster Configuration > Layer 7 – Advanced Configuration.

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

ID	Key	Use	Expires	Server	Remove
0x1338964	192.168.64.7	use=0	1762056	WEB1	✘

Page 1 of 1

Notes:

- Stick tables are used when either source IP persistence or RDP cookie persistence is used with layer 7 Virtual Services
- Individual stick table entries can be removed by clicking the red 'X' in the remove column, the whole table can be cleared by clicking the **Clear Table** button

Graphing

Graphs are automatically configured when new Virtual and Real Servers are defined.

GRAPHS – LOAD BALANCED SERVICES

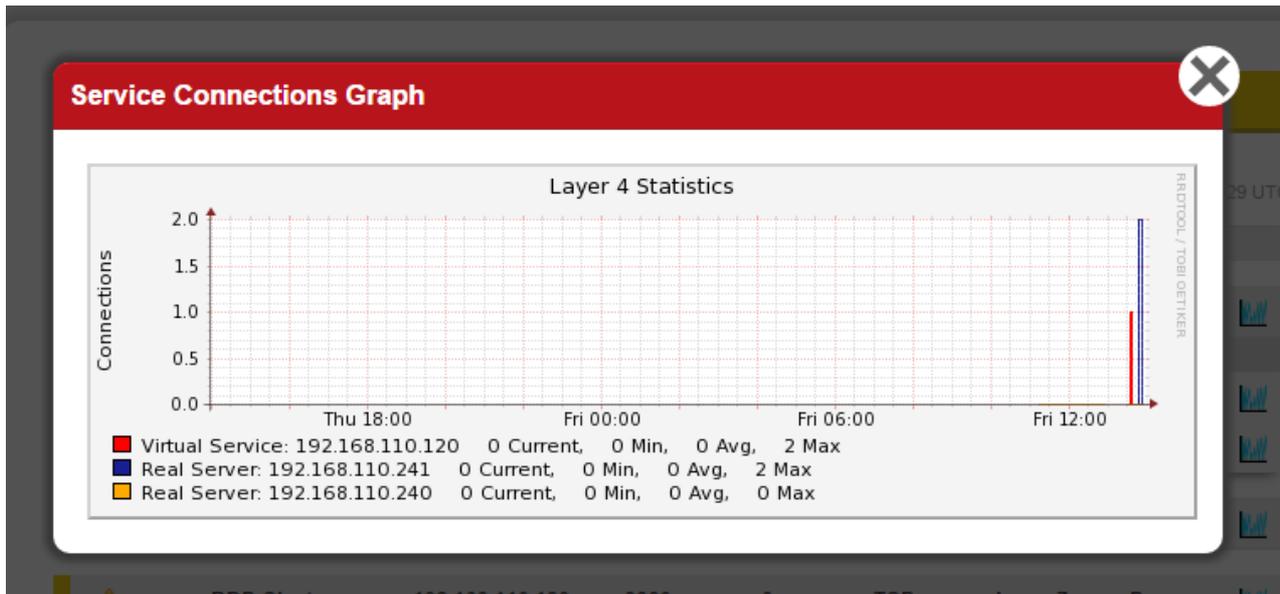
Graphs for the configured Virtual & Real Servers can be accessed either from the System Overview using the appropriate blue colored graph icon that appears next to each VIP and RIP or from the drop-down available in the WebUI under *Reports > Graphing*.

Using the System Overview

The graph is displayed by clicking the relevant blue icon that's displayed next to each VIP/RIP:

↑	HTTP-Cluster1	192.168.110.120	80	0	TCP	Layer 4	DR	
<i>REAL SERVER</i>								
<i>IP</i> <i>PORTS</i> <i>WEIGHT</i> <i>CONNS</i>								
↑	RIP1	192.168.110.240	80	100	0	Drain	Halt	
↑	RIP2	192.168.110.241	80	100	0	Drain	Halt	

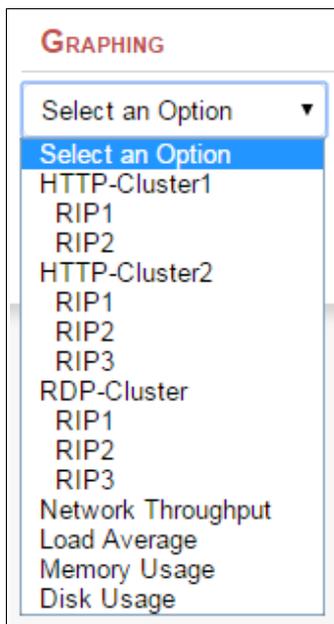
When this method is used, the daily Service Connections Graph (i.e. the last 24 hrs) is displayed for the particular VIP or RIP:



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 > Graphing* menu options as explained below.

Using the WebUI menu option: Reports > Graphing

When selected, a drop-down similar to the following is displayed:



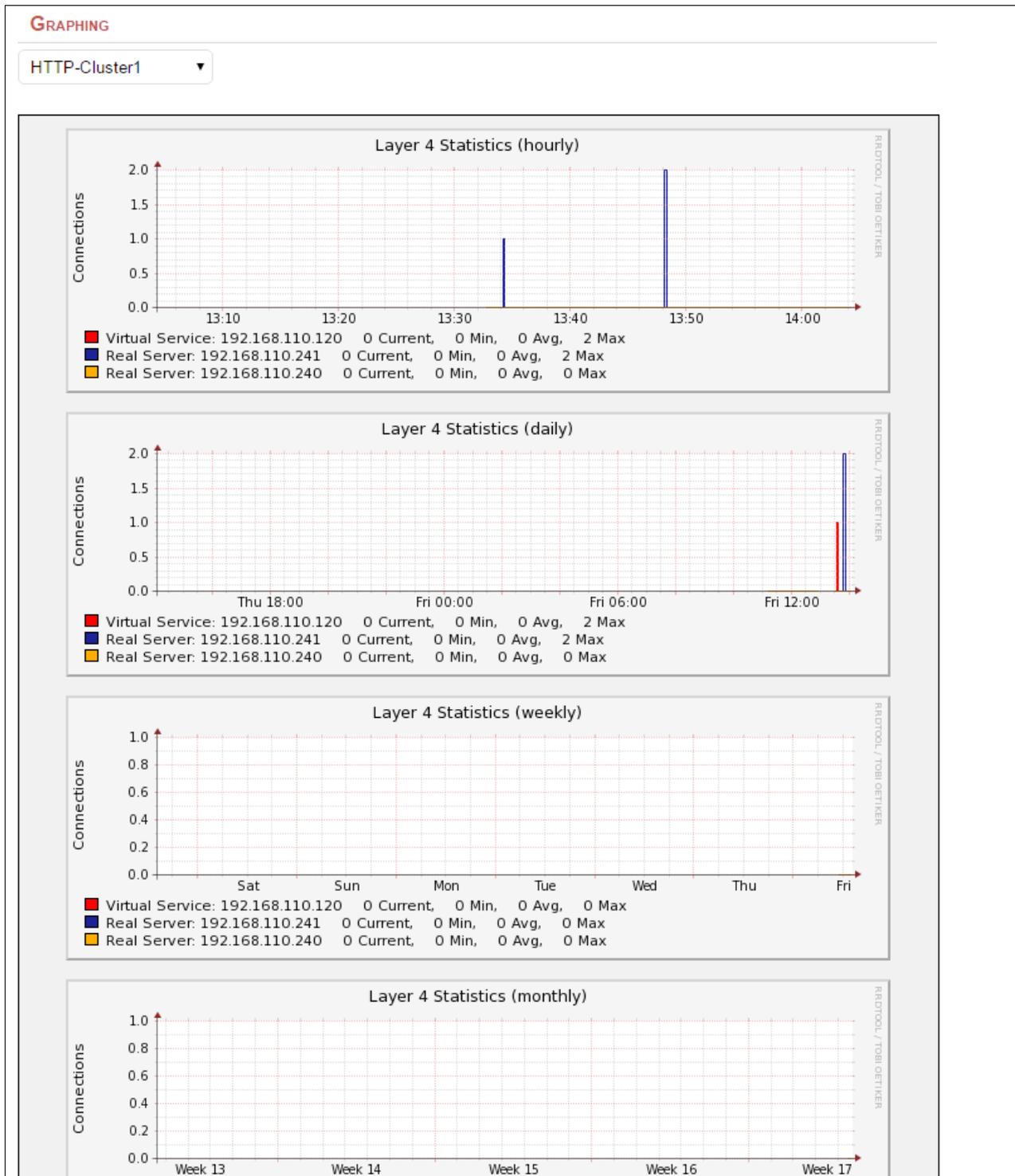
Note:

As VIPs & RIPs are added or removed, these are automatically added/removed from the drop-down list.

Note:

For more information on using the the System Overview, please refer to page [187](#) in chapter 8 – Real Server Health Monitoring & Control.

When selected in this way, a complete list of graphs is displayed for the VIP/RIP selected as shown below:



The following graphs are displayed for each VIP or RIP selected:

- 5 x **Connection graphs**: Hourly, daily, weekly, monthly and yearly
- 5 x **Bytes/s graphs**: Hourly, daily, weekly, monthly and yearly

GRAPHS – APPLIANCE SPECIFIC

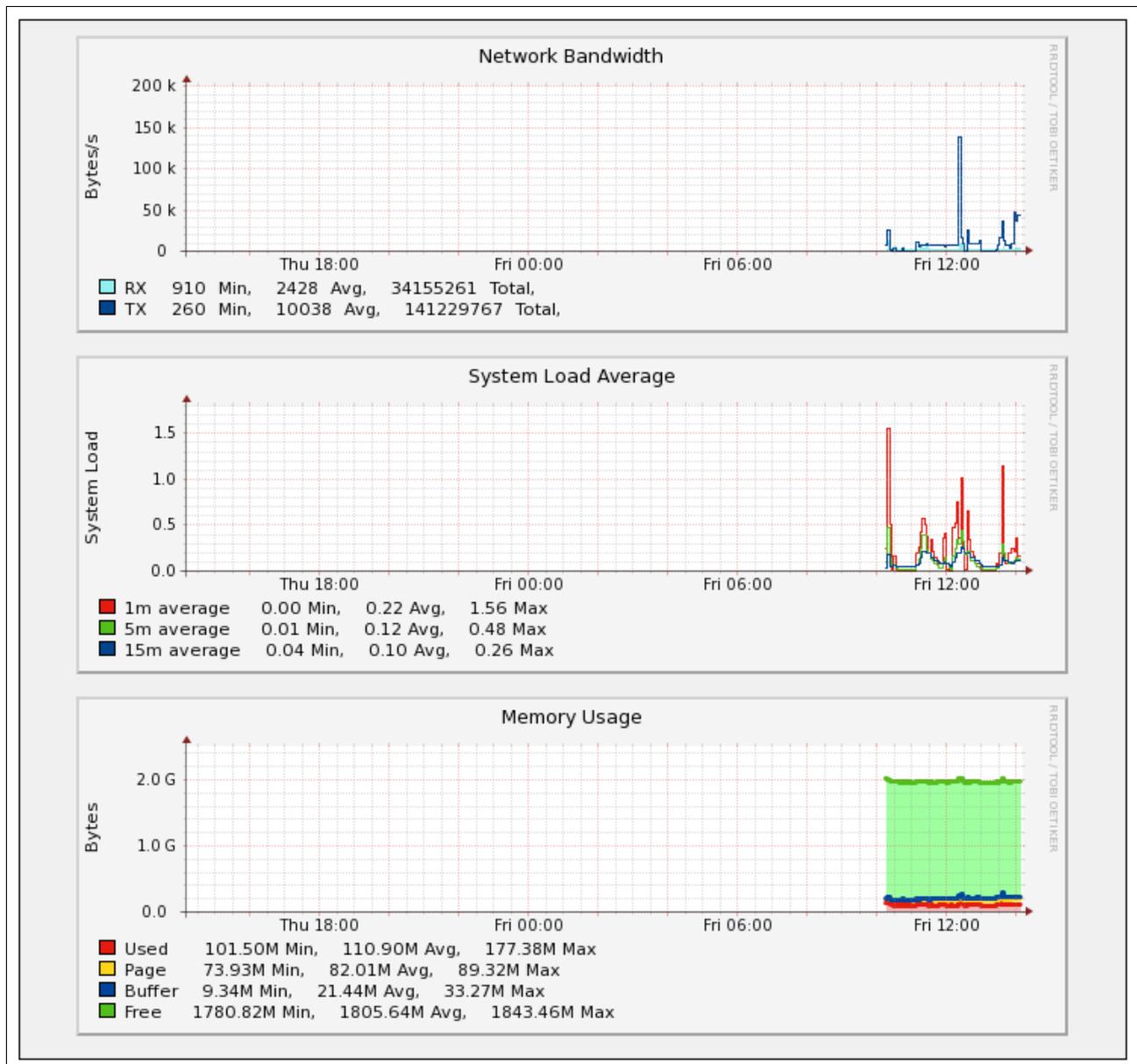
Appliance specific graphs are available for the following statistics:

- Network Throughput
- Load Average
- Memory Usage
- Disk Usage

The first three graphs listed above are displayed in the System Overview by default although these can be disabled/hidden if preferred using the WebUI menu option: *Local Configuration > Graphing*.

All four graphs can also be accessed using the WebUI menu option: *Reports > Graphing*, then selecting the required graph from the bottom of the list.

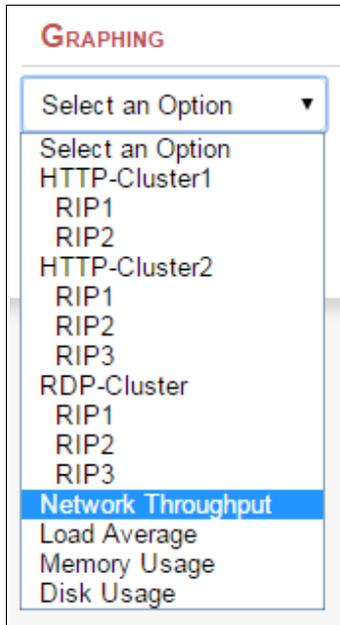
System Overview Graphs



As shown above, daily graphs for **Network Bandwidth**, **System Load Average** and **Memory Usage** are displayed by default in the System Overview. Clicking anywhere within these graph opens the full list of related graphs (hourly, daily, weekly etc.). 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 all VIPs/RIPs as well as the 4 appliance specific graphs is displayed:



GRAPH OPTIONS

A number of graph options are available.

To change the settings:

1. Using the WebUI, navigate to: *Local Configuration > Graphing*

Layer 4	On ▼	?
Layer 7	On ▼	?
Interfaces	On ▼	?
Load Average	On ▼	?
Memory	On ▼	?
Disk Usage	On ▼	?

- Data collection for each graphing category can be enabled (default) by selecting *On* and clicking **Update**
- Data collection for each graphing category can be disabled by selecting *Off* and clicking **Update**

- The stored data for each graphing category can be removed by selecting *Delete* and clicking **Update**

Advanced Configuration Settings

Advanced Configuration		
Interval	<input type="text" value="10"/>	?
Timeout	<input type="text" value="2"/>	?
Threads	<input type="text" value="6"/>	?
Logging	<input type="text" value="Off"/> ▾	?

Interval - Set the 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. Do not change unless advised to do so by support. *WARNING – Changing this value will reset the RRD database files and you will loose all your previous data!!*

Timeout - Set the data collector timeout specified in seconds. Change the timeout for the data collector when querying the various services. Do not change unless advised to do so by support.

Threads - Set the number of data collector process threads. Change the number of collector process threads to use for reading stats. Do not change unless advised to do so by support.

Logging - Enable collector logging for collectd. Warning this is incredibly verbose and should only be used for debugging purposes.

SNMP Reporting

By default, SNMP is disabled on the appliance. Once the SNMP settings are configured using the WebUI menu option: *Local Configuration > SNMP Configuration*, the SNMP service is set to auto start at boot.

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.
```

Note:

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 VIPS & 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 backend services is: 1.3.6.1.4.1.29385.106.1.1

To list the Front End stats use the command shown below:

```
[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.
```

To list the Back End stats use the command shown below:

```
[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.

Note:

Please refer to page [54](#) for details on configuring SNMP settings such as community string etc.

Chapter 14 – 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 WebUI menu 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`

Command 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	19216/haproxy
tcp	0	0	127.0.0.1:7778	0.0.0.0:*	LISTEN	19216/haproxy
tcp	0	0	127.0.0.1:199	0.0.0.0:*	LISTEN	19938/snmpd
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	1466/sshd
tcp	0	0	0.0.0.0:9081	0.0.0.0:*	LISTEN	16114/nginx
tcp	0	0	:::9443	:::*	LISTEN	1627/httpd
tcp	0	0	2001:470:1f09:d72::146:80	:::*	LISTEN	19216/haproxy
tcp	0	0	:::22	:::*	LISTEN	1466/sshd
tcp	0	0	:::9080	:::*	LISTEN	1627/httpd

TELNET

The telnet command is used to communicate with another host using the TELNET protocol. It's very 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 WebUI.

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 WebUI.

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: 1000baseT/Full

10000baseT/Full

Supports auto-negotiation: No

Advertised link modes: Not reported

Advertised pause frame use: No

Advertised auto-negotiation: No

Speed: 10000Mb/s

Duplex: Full

Port: Twisted Pair

PHYAD: 0

Transceiver: internal

Auto-negotiation: off

MDI-X: Unknown

Supports Wake-on: uag

Wake-on: d

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

MICROSOFT NETWORK MONITOR

Network Monitor is a simpler alternative to Wireshark that has some nice features. It can be downloaded from the following location:

<https://www.microsoft.com/en-gb/download/details.aspx?id=4865>

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

The Loadbalancer.org Support Department uses **Teamviewer** for remote desktop support. The client-side software is available at the following links:

Windows clients: <http://downloads.loadbalancer.org/support/quicksupport/WindowsQS.exe>

Mac clients: <http://downloads.loadbalancer.org/support/quicksupport/MacQS.zip>

Linux clients: <http://downloads.loadbalancer.org/support/quicksupport/LinuxQS.tar.gz>

Once downloaded, the client should be installed on a local machine that has access to the load balancer's WebUI and also to the load balancer via SSH (Putty, WinSCP for Windows). Our Support Engineers will provide guidance as required.

Note:

The download links mentioned above can also be accessed using the WebUI menu option:
Support > Useful Links.

Chapter 15 – 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 can be done using the WebUI.

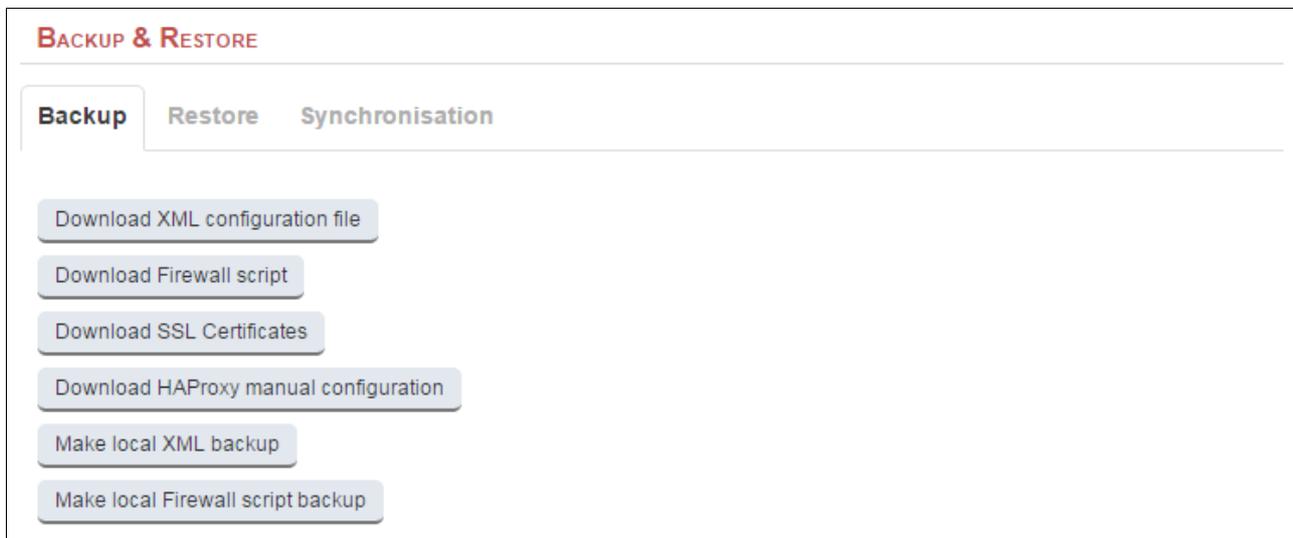
Firewall Script – If manual changes such as manual firewall marks have been made, this file should be backed up - this can be done using the WebUI.

SSL Certificates – If SSL is terminated on the appliance, the certificate files should be backed up - this can be done using the WebUI.

Layer 7 Manual Configuration – If layer 7 (HAProxy) services have been configured this file should be backed up - this can be done using the WebUI.

Backup & Restore

The WebUI can be used for to perform backup and restore functions. To access these options use the WebUI menu option: *Maintenance > Backup & Restore*



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 as a compressed archive file

Download HAProxy manual configuration – download and save the load balancer's layer 7 manual configuration file

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 and Restore – upload an XML file and restore load balancer settings

Upload and Restore SSL Certificates – upload and restores an SSL certificate compressed archive file that was created using the *Download SSL Certificates* backup option

Note:

This option should be used in conjunction with the XML restore option to ensure the certificates are uploaded and correctly recreated in the WebUI.

Restore from the last local XML backup – Restore the last local backup created with the 'Make local XML Backup' option

Restore Manufacturer's defaults – Restore system settings to default values

Note:

The XML restore feature is not backward compatible with previous major versions of the software, e.g. it's not possible to restore a V7.6.4 XML file to a v8.2.x appliance.

Synchronization Options:

Synchronize Configuration with peer – replicate the load balanced services configuration to the slave device.

Note:

For details of which settings are NOT replicated from master to slave when using this option, please refer to page [193](#).

RESTORING XML FILES

The screen shot below shows an ongoing restore from a local XML file backup:

BACKUP & RESTORE

Restoring Configuration from local backup...

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...

Once complete, you'll need to either restart or reload heartbeat to complete the restore process as explained in the yellow message box:

Information: Restored configuration from local backup.

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 (for a single unit) or reloaded (when using a clustered pair)..

Disaster Recovery

V8.2 brings new appliance recovery features which make recovering from a failed master or slave extremely simply, quick and seamless. This method can be used when either the master or slave of an HA Clustered Pair has failed and must be re-introduced/replaced. It works by recovering the configuration from the remaining working appliance, whether this is the master or the slave and restoring the HA pair without any disruption to load balanced services. For more details on this method please refer to page [259](#).

For a failed single appliance, recovery is achieved by restoring the XML configuration file and firewall script/SSL certificates/Layer 7 manual configuration as appropriate.

BEING PREPARED

To be able to quickly recover your appliance if 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 CONFIGURATION FILES TO A REMOTE LOCATION

Login to the Web User Interface:

Username: loadbalancer

Password: loadbalancer

Backup the XML configuration file:

1. Select *Maintenance > Backup & Restore* and click **Download XML configuration file**
2. Select an appropriate location to store the file
3. 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:

1. Select *Maintenance > Backup & Restore* and click **Download Firewall Script**
2. Select an appropriate location to store the file
3. Update the filename if required then save the file

If you're terminating SSL on the load balancer, backup your certificates as well:

1. Select *Maintenance > Backup & Restore* and click **Download SSL Certificates**

2. Select an appropriate location to store the file
3. Update the filename if required then save the file

If you have manual layer 7 services, back these up too:

1. Select *Maintenance > Backup & Restore* and click **Download Haproxy manual configuration**
2. Select an appropriate location to store the file
3. Update the filename if required then save the file

USING WGET TO COPY THE FILES

It's also possible to use wget from a 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
```

Note:

Replace the password 'loadbalancer' with your password if it's been changed.

BACKING UP LOCALLY ON THE LOAD BALANCER

To create local backups of the various configuration files, follow these steps:

1. Log in to the web interface:

Username: loadbalancer

Password: loadbalancer

2. Select *Maintenance > Backup & Restore* and click **Make local XML backup**
3. 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

Note:

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 using the WebUI menu option: *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) of at least 4GB in size. If you're 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 or SSD. If you're upgrading from v5.x 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.

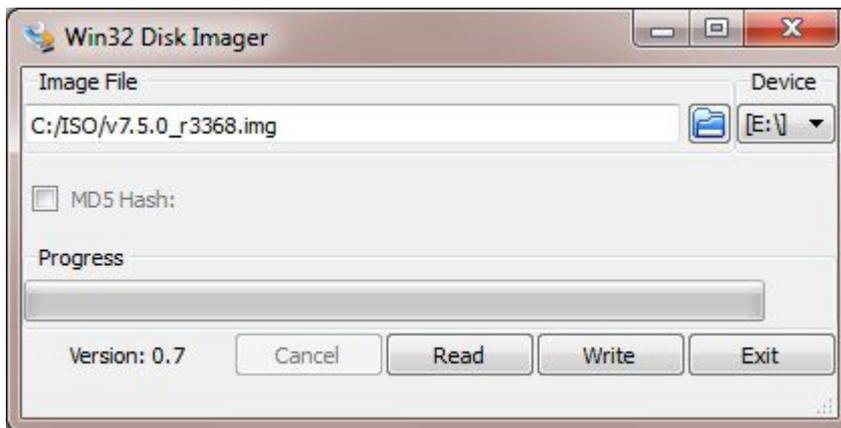
Be careful using this command – make sure you specify the correct disk !!!

Under Windows:

For Windows, a third party image writer must be used. Several free ones are available, the example below uses **Win32 Disk Imager** which can be downloaded here:

<http://sourceforge.net/projects/win32diskimager/>

First extract the archive, then run the executable to install, then run the application



Select the image file and set the appropriate output Device as shown above

Click **Write**

Be careful using this command – make sure you specify the correct disk !!!

Using the USB Stick to restore the Appliance

1. 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)
2. 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
3. As directed, press any key to shutdown the load balancer
4. Once shutdown, remove the USB stick
5. Power up the appliance
6. Login at the console:

Username: root
Password: loadbalancer

7. Run the following command:

lbrestore <Enter>

8. Reboot the appliance once again
9. Set the required IP address using the network setup wizard as described on page [36](#).

Note:

You'll need to reapply your license key file to ensure the newly restored appliance is correctly licensed. Please contact support@loadbalancer.org if you have any issues.

DISASTER RECOVERY AFTER NODE (MASTER OR SLAVE) FAILURE

For a Clustered Pair of load balancers, recovery of a failed node is quick and simple. The procedure is the same whether the master has failed or the slave has failed.

RECOVERY STEPS

Note:

This procedure ensures that the HA pair is re-established *without* disrupting currently running services.

1. If the failed node is still on, power it down
2. For a hardware appliance:
 - Disconnect all cables
 - If the SSD/HD has failed and has been replaced and needs to be re-imaged, follow the steps on page [256](#) to restore the appliance firmware
3. Power up the new/repaired/re-imaged appliance
4. Login to the console as:

username: setup

password: setup

5. Now run through the network setup wizard to configure the initial network settings – ensure these are the same as the failed appliance
6. Once the initial network settings have been configured, you'll be asked if you're recovering from node failure as shown below:

```
Are you recovering from node failure?

Only use this facility if your master or slave appliance has failed
and you'd like this new appliance to be a replacement.
The configuration will be recovered from the remaining
node and the HA clustered pair will be restored without
disrupting running services

(If you are simply deploying a new appliance, hit N)

Do you want to continue? [y/N]
-
```

7. At this point, type 'y' and press <ENTER>
8. You'll now be prompted for information about the remaining appliance as shown below:

```
Enter the IP address of the remaining active
load balancer: 192.168.111.81

Please enter the password for the WUI loadbalancer
user on the active loadbalancer:loadbalancer

Enter the HTTPS port for the WUI on the active loadbalancer
(Default 9443):9443_
```

9. Enter all required details (IP address , password & WebUI HTTPS Port) , then press <ENTER>
10. The process will continue as shown below:

```
Extracting peer config archive
Performing Cleanup.

Ready to begin restoring configuration
The next step will restore -
    Network Configuration
    Heartbeat Configuration
    HAProxy Configuration
    Ldirectord Configuration
    Stunnel Configuration
    Pound Configuration
    WAF Configuration

This may take some time so please be patient.

You will see a restore completed message when done
```

11. Once complete, the restore complete message is displayed as shown below:

```
This may take some time so please be patient.

You will see a restore completed message when done

**The restore has completed.**
**Please reboot the appliance.**
```

12. To complete the process, reboot the appliance as mentioned in the message

**** Once rebooted, the HA pair will be re-synchronized and fully recovered ****

13. The master will now display: **Master | Active | Link**
14. The slave will now display: **Slave | Passive | Link**

Chapter 16 – 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 cover period of any active support agreement
(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: <http://www.loadbalancer.org//resources/free-trial>)

WebUI 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.

CONTACT US

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.

TECHNICAL SUPPORT DOWNLOAD

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_2015-04-28_11_57_59+0000.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 upload server – please ask our support staff about this option.

USEFUL LINKS

This option presents a number of self explanatory web links.

USEFUL LINKS

Manuals

- All Manuals
- Quick Start Guide
- Full Administration Manual
- Virtual Appliance Quick Start Guide
- Amazon Web Services Quick Start Guide
- Azure Cloud Quick Start Guide

Deployment Guides

- Deployment Guides All

Support

- Support Site
- Open A Ticket
- News
- Upload Support Archive

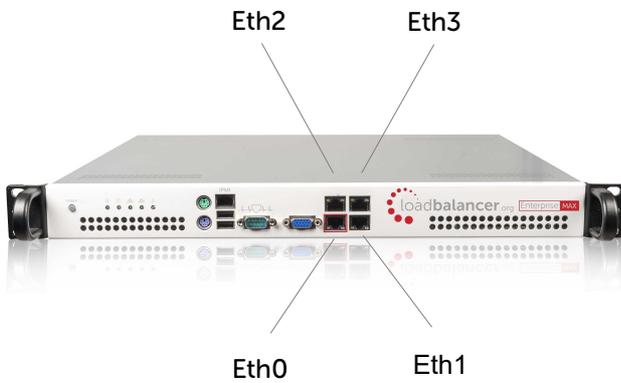
Executables

- Feed-Back Agent (Win)
- Teamviewer (Win)
- Teamviewer (OSX)
- Teamviewer (Linux)

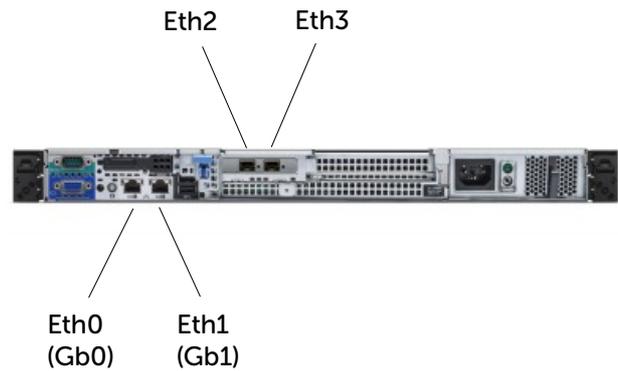
Appendix

Front & Rear Panel Layouts

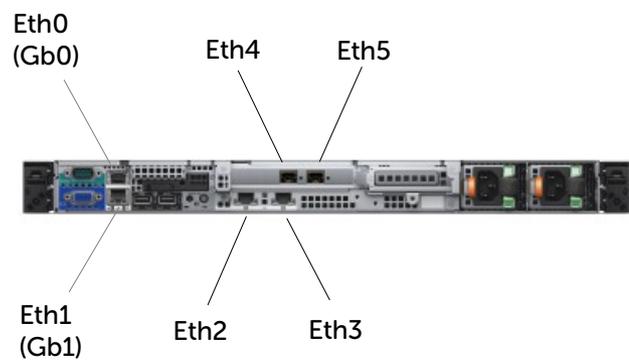
Enterprise R20 & Enterprise Max



Enterprise 10G



Enterprise Ultra



IPMI (Remote Management) Configuration for the Enterprise R20 & MAX

The Enterprise R20 and Enterprise MAX includes 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.

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:

```
Virtual Media  Record  Macro  Options  UserList  Capture  Exit
Aptio Setup Utility - Copyright (C) 2011 American Megatrends, Inc.
  IPMI
BMC network configuration
Lan channel 1
Update IPMI LAN configuration      [No]
Configuration Address source      [Dynamic]

Lan channel 1
Current IP Address                 192.168.65.129
Current Subnet Mask                255.255.192.0
Current MAC Address                00-25-90-74-20-78
Current Gateway IP Address         192.168.64.1

BIOS will set below setting to
IPMI in next BOOT

←→: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit

Version 2.14.1219. Copyright (C) 2011 American Megatrends, Inc.
```

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:



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:



SUPERMICRO[®]

Please Login

Username

Password

username: ADMIN
default password: ADMIN

Once logged in, the following screen is displayed:

The screenshot displays the Supermicro IPMI web interface. At the top, there is a navigation menu with tabs for System, Server Health, Configuration, Remote Control, Virtual Media, Maintenance, and Miscellaneous. The 'System' tab is active, and the 'Summary' section is expanded. The summary shows the following information:

- Firmware Revision: 01.64
- Firmware Build Time: 2012-05-08
- IP address: 192.168.075.111
- MAC address: 00:25:90:74:20:78

Below the summary, there is a 'Remote Console Preview' section with a 'Refresh Preview Image' button and a black console window. At the bottom, there is a 'Power Control via IPMI' section with a status indicator 'Host is currently on' and three buttons: 'Power On', 'Power Down', and 'Reset'. The footer of the interface reads 'Copyright © 2010 Super Micro Computer, Inc.'

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)

Remote Control

To access the systems console, simply click on the Remote Console Preview image. A new window will open with access to the console of the appliance.

Note:

You cannot SSH into the module directly. You need to connect via the IPMI's web interface, then use the remote control option as mentioned above. This can also be accessed using the 'Remote Control' option in the top menu. From here you can use the Launch Console option to launch a virtual Java console which will allow you to use the device as if you stood in front of the device. Next the 'Power Control' options menu will give you several options such as Restart Server, Power off and Power Cycle server. these options will perform the same function as pressing the physical reset button on the unit (Reset Server) as well as being able to perform the same functions as the physical power switch as well.

Please do remember that the IPMI power control options are completely independent of the Loadbalancer software and that the reset option is the same as pressing reset on your PC.

iDRAC (Remote Management) Configuration for the Enterprise 10G & Ultra

iDRAC enables remote management of the Enterprise 10G and Enterprise Ultra appliances.

Default IP Address

By default the following static IP address & mask is assigned to the iDRAC interface:

IP address: **192.168.0.120**

Mask: **255.255.255.0**

This can be changed using the iDRAC management interface accessible at boot-up.

Default Username & Password

The default username & password is:

username: **root**

password: **calvin**

Appliance IPv4 Address Format (CIDR notation)

When specifying IP addresses on the appliance, CIDR format is used. The following table shows the various masks and the corresponding IPv4 IP/CIDR equivalents:

Mask	IP/CIDR
255.255.255.255	a.b.c.d/32
255.255.255.254	a.b.c.d/31
255.255.255.252	a.b.c.d/30
255.255.255.248	a.b.c.d/29
255.255.255.240	a.b.c.d/28
255.255.255.224	a.b.c.d/27
255.255.255.192	a.b.c.d/26
255.255.255.128	a.b.c.d/25
255.255.255.000	a.b.c.d/24
255.255.254.000	a.b.c.d/23
255.255.252.000	a.b.c.d/22
255.255.248.000	a.b.c.d/21
255.255.240.000	a.b.c.d/20
255.255.224.000	a.b.c.d/19
255.255.192.000	a.b.c.d/18
255.255.128.000	a.b.c.d/17
255.255.000.000	a.b.c.d/16
255.254.000.000	a.b.c.d/15
255.252.000.000	a.b.c.d/14
255.248.000.000	a.b.c.d/13
255.240.000.000	a.b.c.d/12
255.224.000.000	a.b.c.d/11
255.192.000.000	a.b.c.d/10
255.128.000.000	a.b.c.d/9
255.000.000.000	a.b.c.d/8
254.000.000.000	a.b.c.d/7
252.000.000.000	a.b.c.d/6
248.000.000.000	a.b.c.d/5
240.000.000.000	a.b.c.d/4
224.000.000.000	a.b.c.d/3
192.000.000.000	a.b.c.d/2
128.000.000.000	a.b.c.d/1

Company Contact Information

Website	URL: www.loadbalancer.org
North America (US)	<p>Loadbalancer.org, Inc. 4250 Lancaster Pike, Suite 120 Wilmington DE 19805 USA</p> <p>Tel: +1 888.867.9504 Fax: +1 302.213.0122 Email (sales): sales@loadbalancer.org Email (support): support@loadbalancer.org</p>
North America (Canada)	<p>Loadbalancer.org Ltd 300-422 Richards Street Vancouver, BC V6B 2Z4 Canada</p> <p>Tel: +1 866.998.0508 Fax: +1 302.213.0122 Email (sales): sales@loadbalancer.org Email (support): support@loadbalancer.org</p>
Europe (UK)	<p>Loadbalancer.org Ltd. Compass House North Harbour Business Park Portsmouth, PO6 4PS UK</p> <p>Tel: +44 (0)330 3801064 Fax: +44 (0)870 4327672 Email (sales): sales@loadbalancer.org Email (support): support@loadbalancer.org</p>
Europe (Germany)	<p>Loadbalancer.org GmbH Tengstraße 27 D-80798 München Germany</p> <p>Tel: +49 (0)89 2000 2179 Fax: +49 (0)30 920 383 6495 Email (sales): vertrieb@loadbalancer.org Email (support): support@loadbalancer.org</p>