Appliance Administration Manual

v6.14

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

Copyright © 2002 - 2011 Loadbalancer.org, Inc.

Table of Contents

Section A – Introduction 7
- Appliance details 8
- Initial configuration 8
- Additional information 8
- Deployment guides 8

Section B – Load balancing Concepts 9
- Load balancing algorithms 10
  - Round Robin 10
  - Weighted Round Robin 10
  - Least Connection 10
  - Weighted Least Connection 10
  - Destination Hashing 10
  - Source Hashing 10
  - Agent Based 10
- Layer 4 vs Layer 7 11

Section C - Quick Start Guide 12
- Loadbalancer.org terminology 13
  - What is a virtual IP address? 13
  - What is a floating IP address? 13
- What are your objectives? 14
- What is the difference between a one-arm and a two-arm configuration? 15
- What are the different load balancing methods supported? 16
  - High-availability configuration of two Loadbalancer.org appliances 18
  - Network diagram: One-Arm – DR Direct Routing (clustered pair) 18
  - Network diagram: Two-Arm - NAT Network Address Translation (clustered pair) 19
  - Network diagram: One-Arm – DR Direct Routing (single unit) 20
  - Network diagram: Two-Arm - NAT Network Address Translation (single unit) 21
- Un-packing and setting up the Loadbalancer.org appliance 22
- Configuring the Loadbalancer.org appliance using the web based wizard 23
  - Network interface configuration 23
  - Accessing the Web User Interface (WUI) 23
  - Example answers using the wizard for a two-arm NAT configuration 24
- Additional Loadbalancer.org configuration (web interface) 25
- Additional real servers (web interface) 26
- Real server configuration for NAT mode 27
- Real server configuration for DR mode (Linux) 27
  - Solving for Linux (with iptables) 27
  - Solving for Linux – alternative method (with arp_ignore sysctl values) 27
- Real server configuration for DR mode (Windows) 28
  - Configuring IIS to respond to both the RIP and VIP 29
  - Resolving ARP issues for Windows server 2000 / 2003 (DR mode only) 30
    - Installing the Microsoft loopback adapter 30
    - Configuring the loopback adapter 31
  - Resolving ARP issues for Windows server 2008 (DR mode only) 33
    - Installing the Microsoft loopback adapter 33
    - Configuring the loopback adapter 34
    - Configuring strong / weak host behavior 35
- Testing the load balancer configuration 36
  - Connection error diagnosis 36
  - Health check diagnosis 37
  - Testing high-availability for a Loadbalancer.org HA-pair 37
- Does your application cluster correctly handle its own state? 38
  - Replication solutions for shared data 38
  - Solutions for session data 38
  - What do you do if your application is not stateless? 39
  - Loadbalancer.org persistence methods 39

Section D – Typical Deployment Examples....40
Example 1 – single appliance (web interface)....41
Network interface configuration....41
Accessing the Web User Interface (WUI)....41
Example 2 - clustered pair (web interface)....42
Network interface configuration....42
Accessing the Web User Interface (WUI)....42
Configuring the virtual servers (VIP) in one-arm DR mode....43
Layer 4 configuration....43
Real server configuration (RIP)....44
Configuring the virtual servers (VIP) in two-arm NAT mode....45
Layer 4 configuration....45
Real server configuration (RIP)....46
Example 3: layer 7 configuration one-arm SNAT mode (HAProxy)....47
Network diagram for layer 7 SNAT mode (clustered pair)....47
Network diagram for layer 7 SNAT mode (single unit)....48
Network Diagram for layer 7 SNAT mode (off site backup)....49
Virtual server configuration....50
Real server configuration....51
SSL termination configuration (Pound)....52
Manage SSL certificate....53
Section E – Detailed Configuration Information....55
Console configuration methods....56
Console access via a serial cable....56
Remote configuration methods....57
Network interface configuration....58
Advanced DR considerations....59
What is the ARP problem?....59
Solving the ARP problem....59
Solving for Linux (with iptables)....59
Solving for Linux – althernative method (with arp_ignore sysctl values)....59
Solving for Solaris....60
Solving for Mac OS X or BSD....60
Solving for Windows 2000 / 2003....60
Windows 2003 R2 / R1 (With SP1) Firewall Settings....63
Solving for Windows 2008....64
Windows 2008 R2 Firewall Settings....67
Windows 2008 R1 Firewall Settings....67
Advanced NAT considerations....68
Explaining the RIP & VIP in NAT mode....70
Network Diagram: one arm – NAT Network Address Translation (clustered pair) 71
Route configuration for Windows Server with one arm NAT mode....72
Route configuration for Linux with one arm NAT mode....72
Advanced Layer 7 considerations....73
Load balancing based on URL match with HAProxy....73
Handling Manual Changes to the HAProxy configuration file....74
HAProxy error codes....74
SSL Certificates & Pound....75
SSL termination concepts....75
Layer 7....75
Layer 4....75
Health monitoring....76
Load balancer health....76
Heartbeat Configuration....76
Serial Cable....76
Unicast (ucast)....76
Broadcast (bcast)....77
Ping Node....77
Real server health....77
Configuration – Layer 4....77
Configuration – Layer 7....80
Advanced firewall considerations....81

Firewall marks....81
FTP....82
Changing the FTP Port in NAT Mode....83
FTP negotiate health check....84
FTP recommended persistence settings....84
Limiting passive ports....85
For Linux....85
For Windows 2008....85
For Windows 2003....86
For Windows 2000....86
Persistence considerations....87
Persistence > 15 minutes....87
Server maintenance when using persistence....87
Persistence state table replication....88
Terminal Server RDP considerations....89
RDP – Layer 4....89
Layer 7 (RDP Cookies)....89
NIC bonding and high-availability....90
Example 1: Bonding for bandwidth....90
Example 2: Bonding for high-availability (recommended)....90
Example 3: Bonding for high-availability & bandwidth....90
SNMP reporting....91
SNMP for layer 4 based services....91
SNMP for layer 7 based services....91
Feedback agents....92
Installing the Windows agent....92
Installing the Linux/Unix agent....93
Custom HTTP agent....93
Changing the local date, time & time zone....94
NTP configuration....94
Restoring Manufacturer's settings....94
From the console....94
From the WUI....94
Force master/slave take-over in a clustered pair....95
Force the Master to become passive....95
Force the Master to become active....95
Active / Active load balancer configuration....96
Section F – Disaster Recovery....97
Being prepared....98
Backing up to a remote location....98
Backing up to the load balancer....98
Appliance recovery using a USB memory stick....99
Disaster recovery after master failure....100
Disaster recovery after slave failure....103
Section G – Web User Interface Reference....106
View Configuration....107
System Overview....107
XML....107
Layer 4....107
Layer 7 (HAProxy)....107
SSL Termination (Pound)....107
Network Configuration....107
Heartbeat Configuration....107
Heartbeat Resources....107
Routing Table....107
Firewall Rules....107
Edit Configuration....108
Logical Layer 4 Configuration....108
Virtual Servers....108
Real Servers....112
Logical Layer 7 Configuration....114
Virtual Servers (HAProxy)....114

Real Servers (HAProxy)....115
SSL Termination (Pound)....116
Manage this SSL certificate....117
Create and Upload a PEM file....118
Adding an Intermediate key to the certificate chain....119
Windows Servers....121
Import certificates exported from Windows Server....122
Converting an encrypted private key to an unencrypted key....122
Limiting Ciphers....122
Physical Load Balancer Configuration....123
Network Interface Configuration....123
Aliases....123
VLANS....123
DNS & Hostname....123
Floating IP(s)....123
Setup Wizard....123
Upgrade License Key....124
Advanced....124
Execute a shell command....124
Heartbeat Configuration....124
Global Settings....125
Layer 4....126
Pound SSL....126
Layer 7 HAProxy....127
Internet Access....128
Firewall....128
Maintenance....129
Maintain Real Servers....129
System Overview....129
Take a real server offline or online....129
Backup & Recovery....130
Configuration Backup....130
Disaster Recovery....130
Services....130
Restart HAProxy....130
Restart Pound-SSL....130
Restart Heartbeat....130
Restart Ldirectord....130
Power Control....131
Shut down and restart server....131
Shut down and halt server....131
Security & Maintenance....131
Online Software Update....131
Fallback Page....131
Firewall Script....132
Firewall Lock Down Wizard....133
Initialize Graphs (rrdtool)....134
Passwords....134
Reports....135
System Overview....135
Status....135
Status (HA Proxy)....135
Traffic Rate Per Second....136
Traffic Qty....136
Current Connections....136
Current Connections (Resolve Host name)....136
Graphical Stats Over Time....136
Logs....137
Ldirectord....137
Lbadmin....137
Heartbeat....137
HAProxy....137
Pound (SSL)....137

Reset all packet counters to zero.....137
Change the date/time settings.....137

Section A – Introduction

Appliance details

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

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

Initial configuration

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

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

NB. If the appliance is already running you can plug a USB keyboard in and it will work, we recommend you leave it plugged into a KVM switch preferably with Remote IP Console access.

Additional information

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

Deployment guides

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

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

- Load Balancing IIS Web Servers
- Load Balancing Web Proxies / Filters
- Load Balancing OCS 2007 R2
- Load Balancing Terminal Services
- Load Balancing Exchange 2010

Section B – Load balancing Concepts

Load balancing algorithms

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

Round Robin

With this method incoming requests are distributed equally amongst the available real servers. If this method is selected, all the servers assigned to a virtual service should have similar specifications. if the servers have different capacities, then another method such as weighed round robin would be more suitable.

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.

Least Connection

This method assigns new jobs to real servers that have fewer active jobs. Connections that are maintained over time are taken into consideration, whereas for the two round robin approaches above this does not happen and therefore servers can become overloaded with connections that remain active for long periods of time.

Weighted Least Connection

This method works in a similar way to the Least Connection method but in addition also considers the servers weight. 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.

Destination Hashing

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

Source Hashing

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

Agent Based

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

Layer 4 vs Layer 7

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

The Basics

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

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

Performance

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

Persistence

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

Real Server Changes

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

Transparency

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

Our Recommendation

Where possible, we recommend that Layer 4 Direct Routing (DR) mode is used. This offers the best possible performance since the client replies directly bypassing the load balancer and is also relatively simple to implement. Ultimately, the final choice does depend on your specific requirements and infrastructure.

Section C - Quick Start Guide

(Also available as a separate download)

Loadbalancer.org terminology

Acronym	Terminology
Load Balancer	An IP based traffic manager for clusters
VIP	The Virtual IP address that a cluster is contactable on (Virtual Server)
RIP	The Real IP address of a back-end server in the cluster (Real Server)
GW	The Default Gateway for a back-end server in the cluster
WUI	Web User Interface
Floating IP	An IP address shared by the master & slave load balancer when in a high-availability configuration (shared IP)
Layer 4	Part of the seven layer OSI model, descriptive term for a network device that can route packets based on TCP/IP header information
Layer 7	Part of the seven layer OSI model, descriptive term for a network device that can read and write the entire TCP/IP header and payload information at the application layer
DR	Direct Routing is a standard load balancing technique that distributes packets by altering only the destination MAC address of the packet
NAT	Network Address Translation – Standard load balancing technique that changes the destination of packets to and from the VIP (external subnet to internal cluster subnet)
SNAT *(HAProxy)*	Source Network Address Translation – Load balancer acts as a proxy for all incoming & outgoing traffic
SSL Termination *(Pound)*	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 physical network cards connected to two subnets
Eth0	Usually the internal interface also known as Gb0
Eth1	Usually the external interface also known as Gb1

What is a virtual IP address?

Most load balancer vendors use the term virtual IP address (VIP) to describe the address that the cluster is accessed from.

It is important to understand that the virtual IP (VIP) refers both to the physical IP address and also to the logical load balancer configuration. Likewise the real IP (RIP) address refers both to the real servers physical IP address and its representation in the logical load balancer configuration.

What is a floating IP address?

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

What are your objectives?

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

Hardware 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 advise that you only deploy load balancers as clustered pairs to remove this single point of failure.
Maintenance	Using the appliance, you can easily bring servers on and off line to perform maintenance tasks, without disrupting your users.

In order to achieve all three objectives of performance, reliability & maintenance in a web based application, you must not require persistence on the load balancer.

What is the difference between a one-arm and a two-arm configuration?

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

NB: To add even more confusion, having a 'one-arm' or 'two-arm' solution may or may not imply the same number of network cards.

Loadbalancer.org topology definition:

One-Arm	The load balancer has **one** physical network card connected to **one** subnet
Two-Arm	The load balancer has **two** physical network cards connected to **two** subnets

What are the different load balancing methods supported?

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

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

Key:

Recommended

Recommended if cookie insertion is mandatory

Only required for Direct Routing implementation across routed networks

Loadbalancer.org Recommendation:

The one-arm direct routing (DR) mode is the recommended mode for Loadbalancer.org installation because it's a very high performance solution with very little change to your existing infrastructure.

Sometimes it is not possible to use DR mode. The two most common reasons being: if the application cannot bind to the RIP & VIP at the same time; or if the host operating system cannot be modified to handle the ARP issue.

The second choice is Network Address Translation (NAT) mode. This is also a fairly high performance solution but it requires the implementation of a two-arm infrastructure with an internal and external subnet to carry out the translation (the same way a firewall works).

Network engineers with experience of hardware load balancers will have often used this method.

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

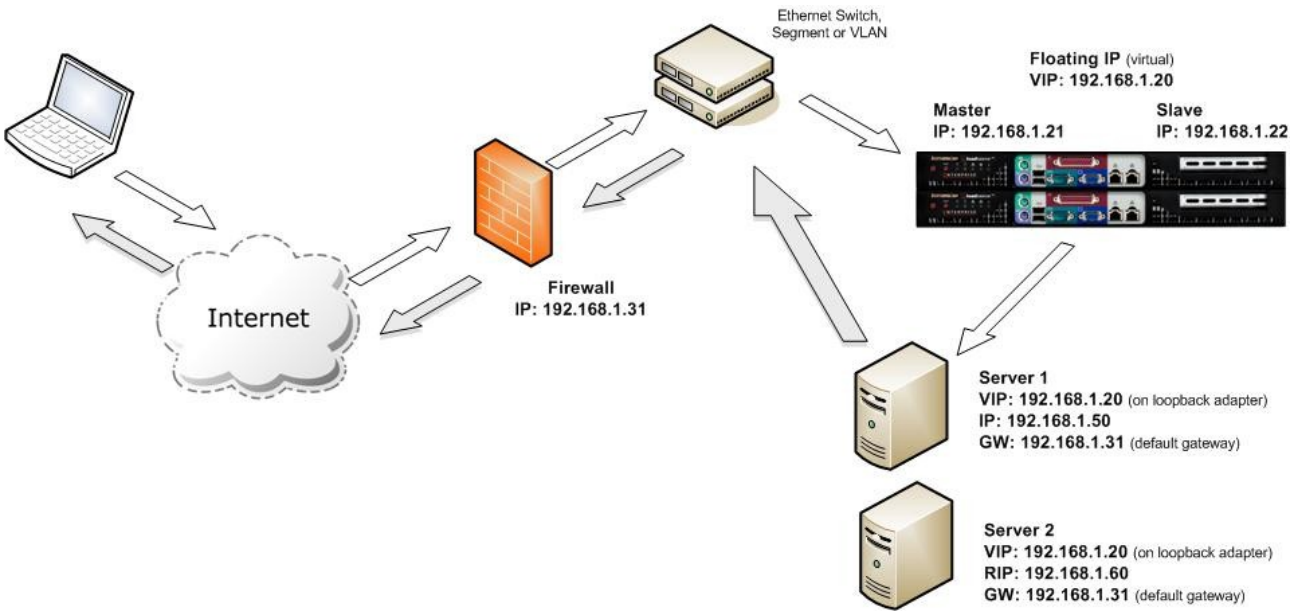
The following section describes the different network configuration possibilities for NAT & DR mode in more detail.

If your application doesn't maintain its own state information then you may need to use cookie insertion, please refer to the full administration manual for configuration details.

High-availability configuration of two Loadbalancer.org appliances

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

Network diagram: One-Arm – DR Direct Routing (clustered pair)

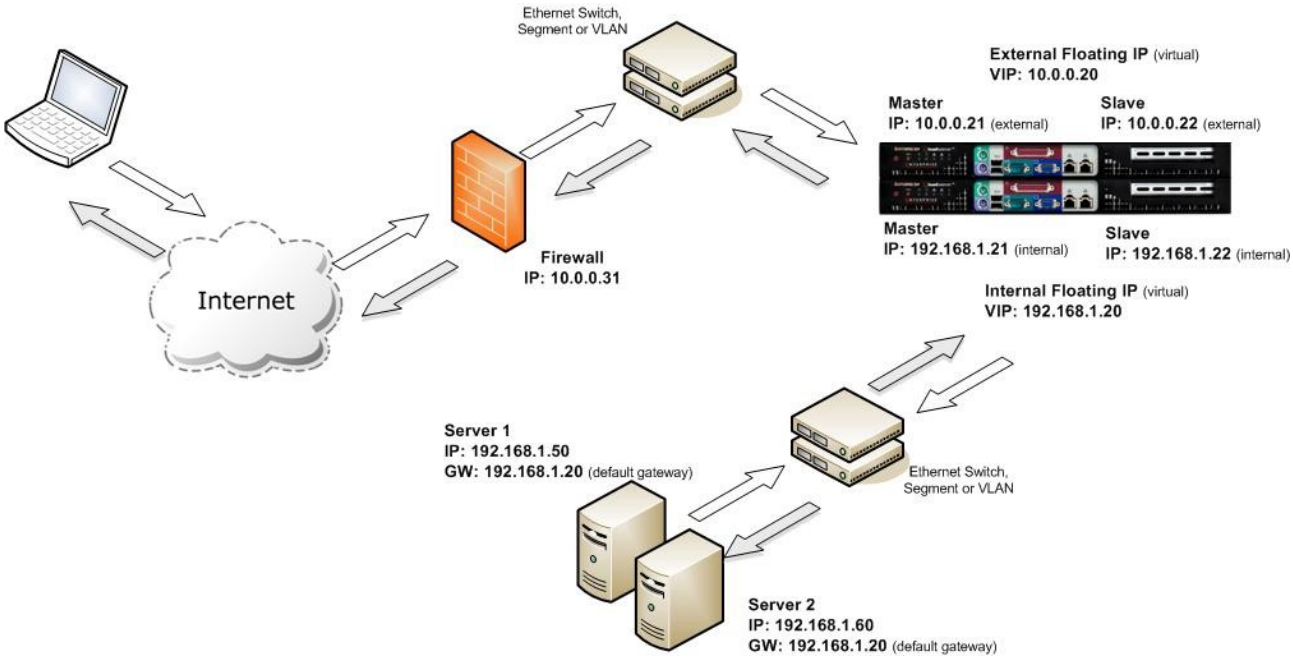


Notes:

- Direct routing works by changing the destination MAC address of the incoming packet on the fly which is very fast
- However, this means that when the packet reaches the real server it expects it to own the VIP. This means you need to make sure the real server responds to the VIP, but does not respond to ARP requests. Please refer to pages 27 - 35 for more details on resolving the ARP issue
- 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
- Direct routing mode enables servers on a connected network to access either the VIPs or RIPs, no extra subnets or routes are required on the network
- The real server must be configured to respond to its own IP address & the VIP
- Port translation is not possible in DR mode i.e. have a different RIP port than the VIP port
- Administration of the load balancers is via any active IP address

When using a clustered pair of load balancers in one-arm DR mode all load balanced services must be configured on a floating IP to enable failover to the slave unit to occur.

Network diagram: Two-Arm - NAT Network Address Translation (clustered pair)



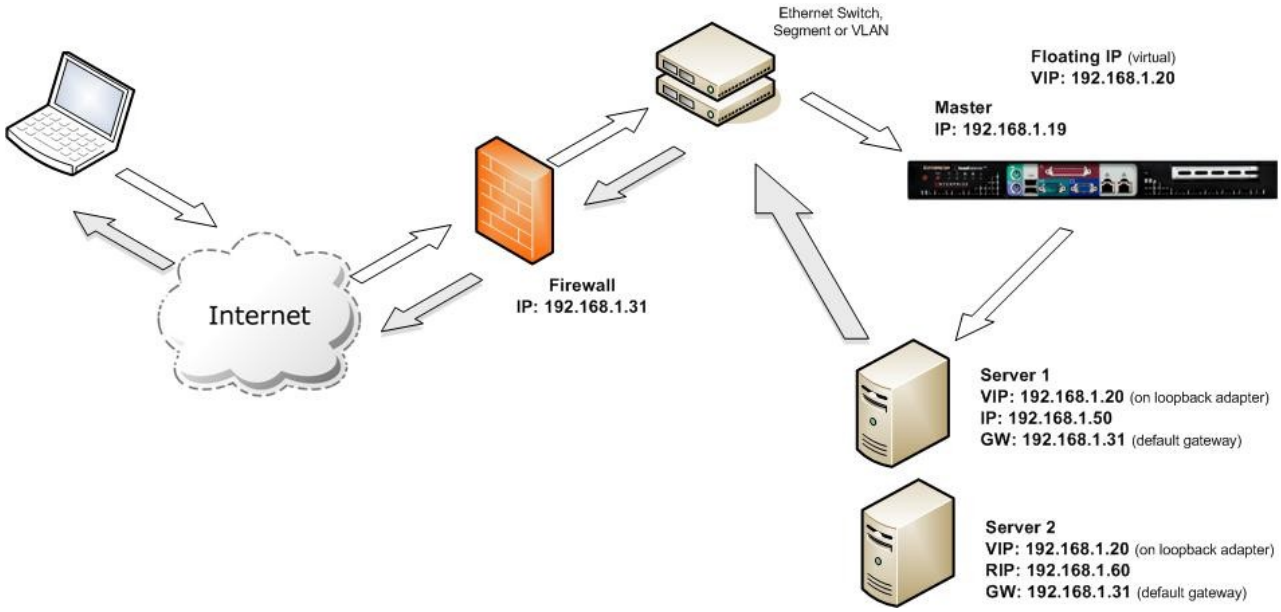
Notes:

- In two-arm NAT mode the load balancer translates all requests from the external virtual server to the internal real servers
- It is a good idea to use *eth1* as your external network and *eth0* as internal, otherwise you will need to change the *autonat* setting in *Edit Configuration* > *Global Settings*
- The real servers *must* have their default gateway configured to point at a floating IP the load balancer
- Real servers are automatically given access to the Internet through the load balancer (via autonat)
- A floating IP must be configured for hosting the virtual server (public access)
- Administration of the load balancers is via any active IP address
- If you want real servers to be accessible on their own IP address for non-load balanced services, e.g. SMTP, you will need to set up individual SNAT and DNAT firewall script rules for each real server. Please refer to Advanced NAT Considerations in section E of the administration manual for more details on this

When using a clustered pair of load balancers in two-arm NAT mode all load balanced services must be configured on a floating IP and the real servers must also have their default gateway directed to a floating IP. This enables failover to the slave unit to occur.

You can also configure the load balancers in one-arm NAT mode, but in order to make the servers accessible from the local network you need to change some routing information on the real servers. Section E of the administration manual provides more detail for one-arm NAT mode. The admin manual is available at http://www.loadbalancer.org/pdffiles/loadbalanceradministration.pdf

Network diagram: One-Arm – DR Direct Routing (single unit)



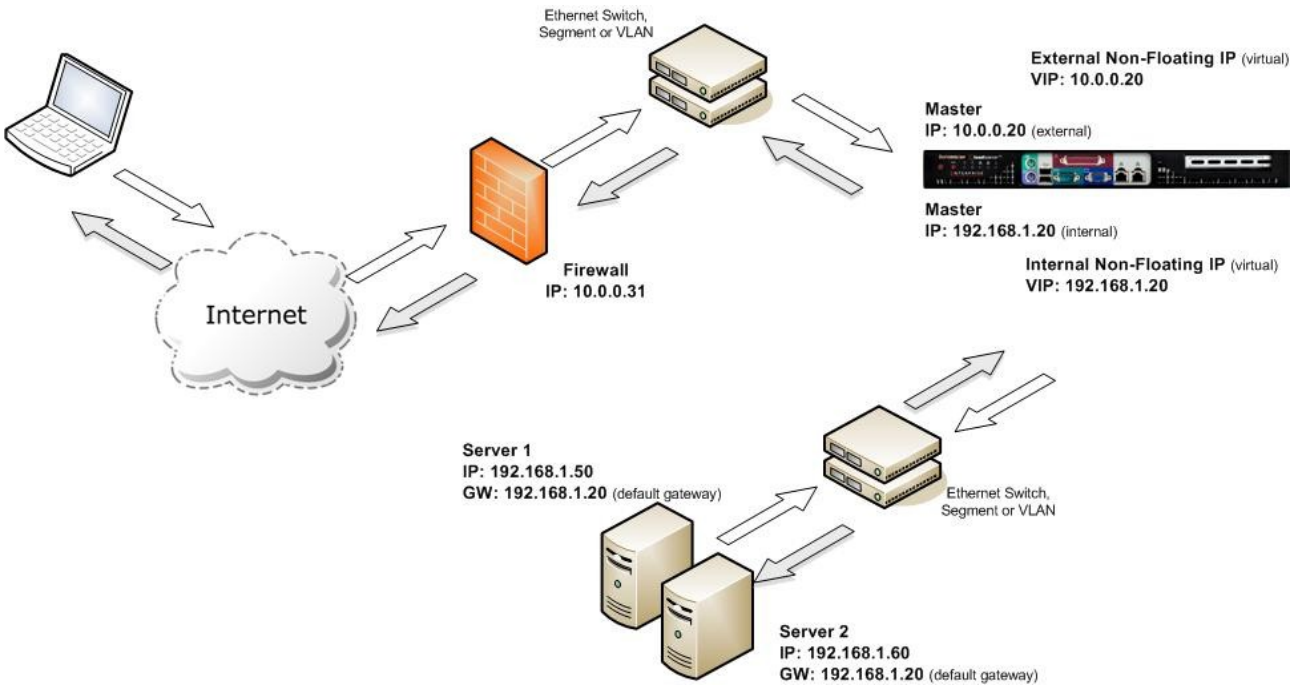
Notes:

- When using a single load balancer only one IP address is required

  NB: Please note however that it is still good practice to use an extra dedicated floating IP to make adding a 2nd unit (to create a clustered pair) much easier should this be needed later
- Direct routing works by changing the destination MAC address of the incoming packet on the fly which is very fast
- However, this means that when the packet reaches the real server it expects it to own the VIP. This means you need to make sure the real server responds to the VIP, but does not respond to ARP requests. Please refer to pages 27 - 35 for more details on resolving the ARP issue
- 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
- Direct routing mode enables servers on a connected network to access either the VIPs or RIPs, no extra subnets or routes are required on the network
- The real server must be configured to respond to its own IP address & the VIP
- Port translation is not possible in DR mode i.e. have a different RIP port than the VIP port
- Administration of the load balancers is via any active IP address

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

Network diagram: Two-Arm - NAT Network Address Translation (single unit)



Notes:

- In two-arm NAT mode the load balancer translates all requests from the external virtual server to the internal real servers
- It is a good idea to use *eth1* as your external network and *eth0* as internal, otherwise you will need to change the *autonat* setting in *Edit Configuration* > *Global Settings*
- The real servers *must* have their default gateway configured to point at a floating IP the load balancer
- Real servers are automatically given access to the Internet through the load balancer (via autonat)
- Administration of the load balancers is via any active IP address
- If you want real servers to be accessible on their own IP address for non-load balanced services, e.g. SMTP, you will need to set up individual SNAT and DNAT firewall script rules for each real server. Please refer to Advanced NAT Considerations in section E of the administration manual for more details on this

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

When using a load balancer in two-arm NAT mode, all load balanced services can be configured on the external IP (eth1). The real servers must also have their default gateway directed to the internal IP. Please note that it is good practice to use extra dedicated floating IP's for the VIP and the gateway to make adding 2nd unit (to create a clustered pair) much easier.

Un-packing and setting up the Loadbalancer.org appliance

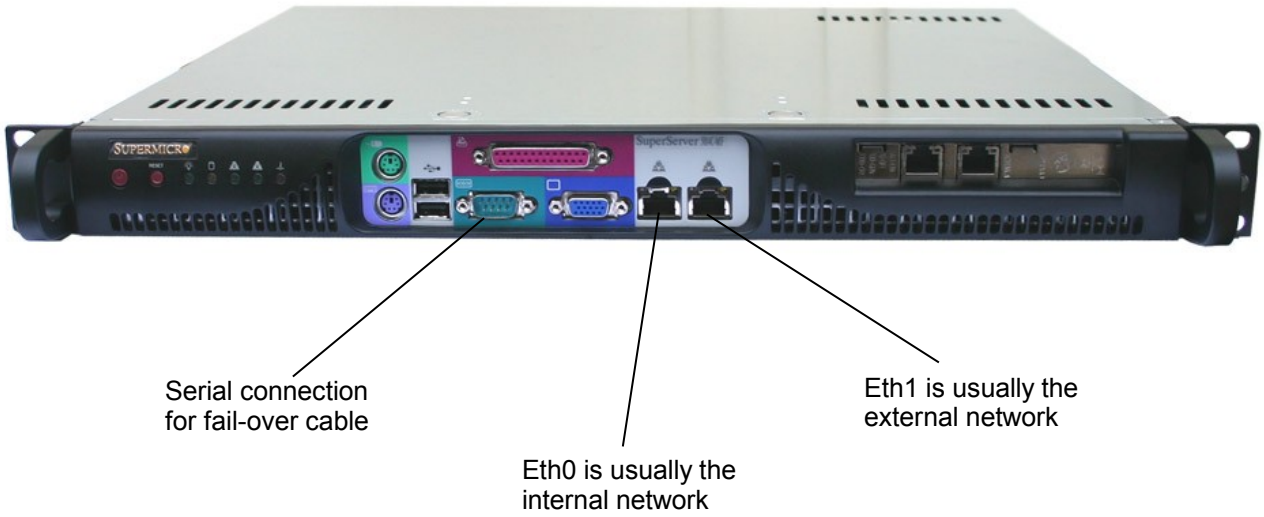
1. Remove all packaging
2. Rack mount the appliance as required
3. The power supply is an auto sensing unit (115v or 230v)
4. Connect the power lead from the power socket to the mains or UPS
5. Connect your network cable from your switch or hub to the internal network port (*eth0*)
6. If using a two-armed configuration connect a second network cable to the external port (*eth1*)

If two load balancers are being used, connect a null modem cable (supplied with the appliance) between the two serial fail-over ports and configure the slave first.

7. Attach a monitor to the VGA port
8. Attach a keyboard to the USB or PS/2 port
9. Check mains power is on
10. Press the power switch on (fans should start)
11. Allow a minute for booting

The next few pages detail the following steps:

12. Configure the load balancer using the web based setup wizard
13. Configure the load balancer using the console wizard
14. Add extra real servers via the web administration interface
15. Configuring the real servers for either NAT or DR mode
16. Testing the load balancer configuration



Configuring the Loadbalancer.org appliance using the web based wizard

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

Network interface configuration

Log in to the console:

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

You can access the web interface either via links at the console or from a web browser on a client connected to the same network (recommended). The default IP address is *192.168.2.21/24.* To change this, at the console use:

```
ifconfig eth0 <IP address> <netmask> up
```

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

Accessing the Web User Interface (WUI)

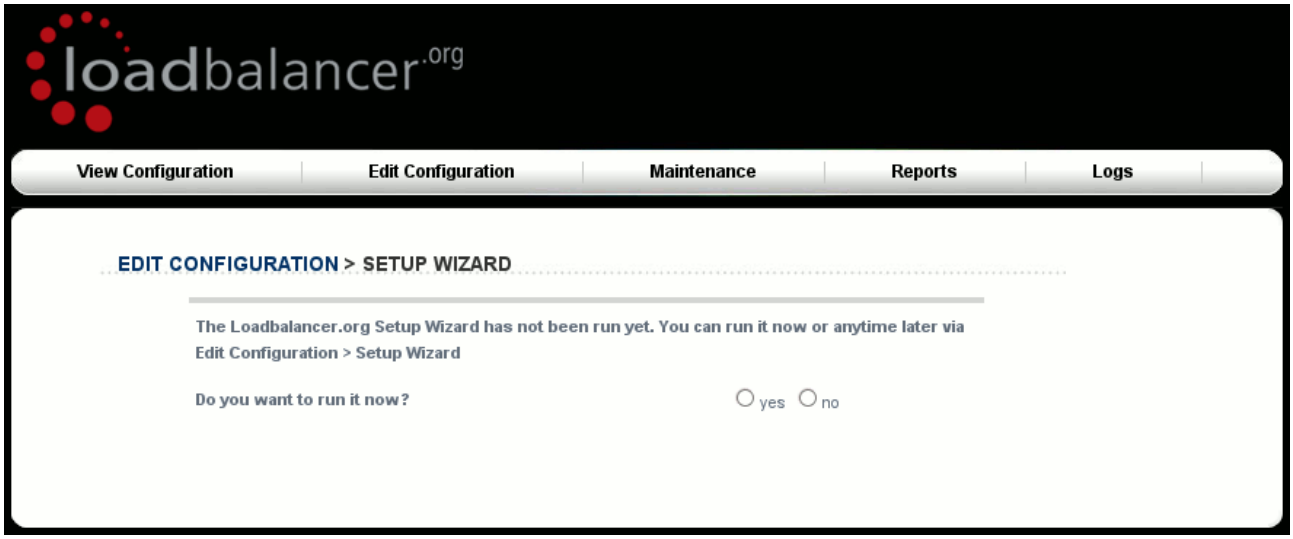
With a web browser, access access the WUI : ***http://192.168.2.21:9080/lbadmin/***

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

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

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

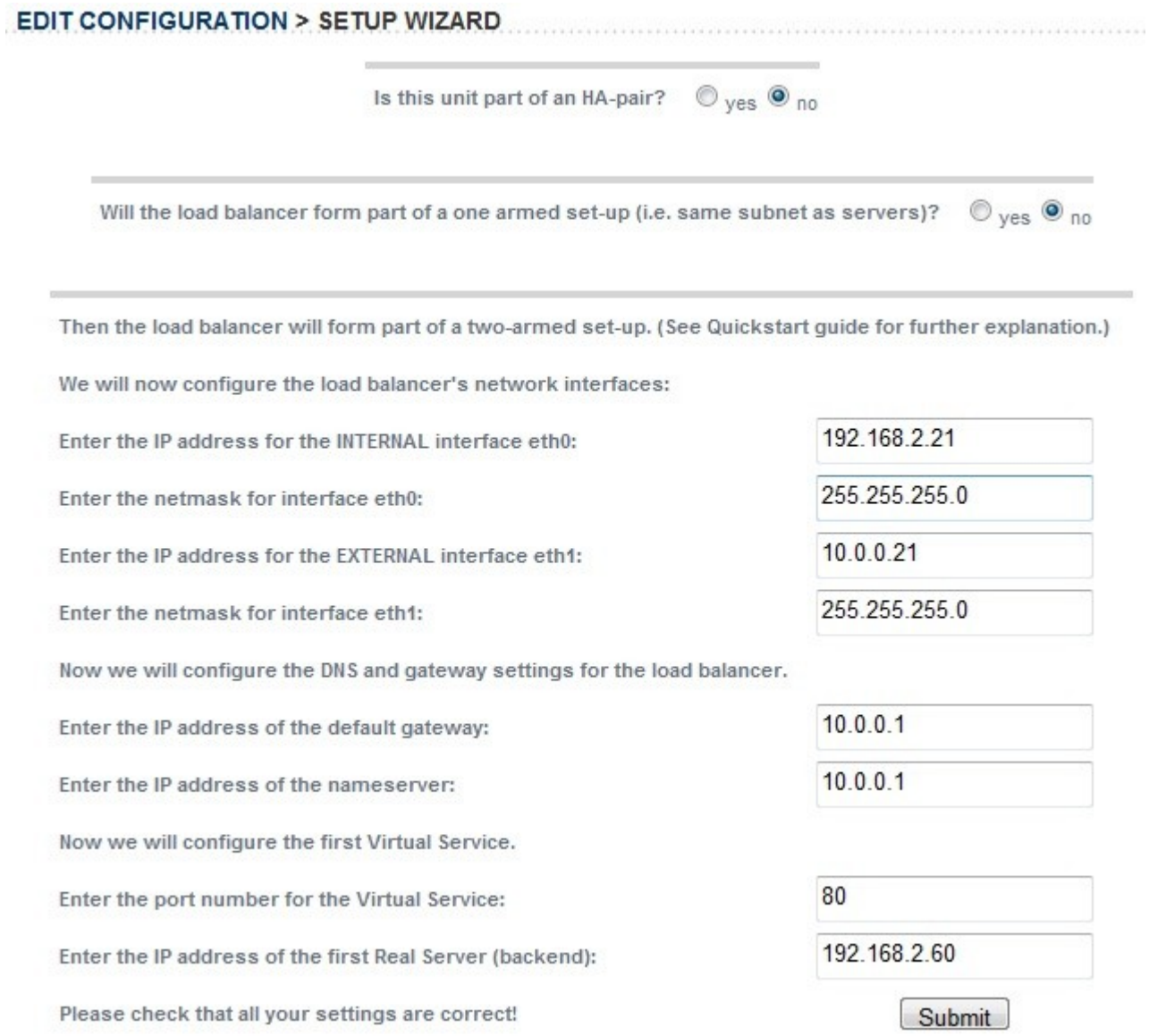
This will take you to the Loadbalancer.org web interface, where the web based configuration wizard will start by default the first time it is accessed. This wizard will ask a series of questions in order to get you started quickly.



All further configuration and administration tasks can then be carried out through the web interface.

Example answers using the wizard for a two-arm NAT configuration

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



Check that your settings are correct and click **Submit**. Once the wizard is complete the load balancer is configured and ready to use. Note that the wizard can also be run via the console by running the command ***lbwizard*** as described on the console welcome screen.

For NAT mode you also need to configure the real server to ensure that it uses the internal IP address of the load balancer as its default gateway. Once this is done you can test the virtual server form the external network. By default, the wizard uses the IP address of the external interface for the first virtual server – 10.0.0.21 in this example.

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

To restore the manufacturer's settings – at the console use the command ***lbrestore*** or in the WUI goto *Maintenance > Disaster Recovery > Restore Manufacturer's Settings.*

Additional Loadbalancer.org configuration (web interface)

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

If you used the web based wizard then you will already be in the web interface. From here all administration tasks can be carried out.

If you chose to use the console wizard then you can now access the web interface either via links at the console or from a web browser on a client connected to the same network (recommended).

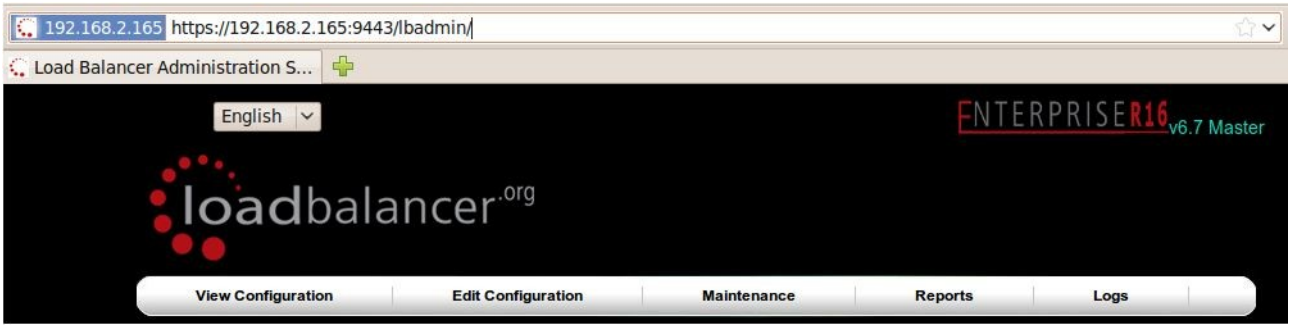
With a web browser access the web interface : ***http://192.168.2.21:9080/lbadmin/***

(replace 192.168.2.21 with the correct address)

Log in to the console:

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

NB. If you prefer you can use the HTTPS administration address : https://192.168.2.21:9443/lbadmin/



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

Additional real servers (web interface)

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

- Use *Edit Configuration* > *Layer 4 Configuration* > *Real Servers* and you should see your logical virtual servers listed, select the one you want and click on **Add a new Real Server**

- You just need to give the IP address and port number of your web server
- Correctly specify your real servers IP address and service port
- Weight defaults to 1 making real servers active immediately
- Leave the minimum & maximum connections as 0 for unrestricted

- The forwarding method will default to NAT if you have a two-arm configuration or DR if you have a one-arm configuration

You have now finished the configuration of both load balancers for the cluster. Now you must configure the back-end web servers to respond to the load balancer's requests.

Real server configuration for NAT mode

If you are using a two-arm NAT load balancing method the real server configuration is a simple case of configuring the load balancer as the default gateway. The real server must also have a valid IP address in the internal subnet behind the load balancer.

Failure to correctly configure the real servers default gateway is the most common problem in NAT configurations. Please refer to section E for more details.

Real server configuration for DR mode (Linux)

If you are using a one-arm DR load balancing method each web server requires the ARP problem to be handled. Every real server must be configured to respond to the VIP address as well as the RIP address. This is because in DR mode load balanced traffic arrives on the VIP address but the load balancer health checks are on the real server's IP address.

Solving for Linux (with iptables)

This is the recommended method for Linux. You can use iptables (netfilter) on the real server to re-direct incoming packets destined for the virtual server IP address. This is a simple case of adding the following command to your start up script (rc.local):

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

i.e. Redirect any incoming packets destined for 10.0.0.21 (virtual server) to my local address.

(Don't forget to change the IP address to be the same as your virtual server)

Solving for Linux – alternative method (with arp_ignore sysctl values)

Each real server needs a loopback IP address to be configured as the VIP. This address needs to be stopped from responding to ARP requests and the web server needs to be configured to respond to this IP address.

With most modern Linux kernels (>2.6) you can alter the ARP behavior allowing you to configure a loopback adapter without worrying about ARP issues. To do this just add the following lines to /etc/sysctl.conf and re-boot, or run /sbin/sysctl.conf -p to reload the file:

```
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
```

Alternatively, the following commands may be used to change the settings interactively during runtime:

```
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
```

Once you have configured your Linux real server so that it won't respond to ARP requests for the loopback adapter you can configure your VIPs as follows:

```
ifconfig lo:0 VIP netmask 255.255.255.255 up
```

To make this permanent and reboot safe you may include this command in rc.firewall or in a equivalent customizable start-up script.

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

Real server configuration for DR mode (Windows)

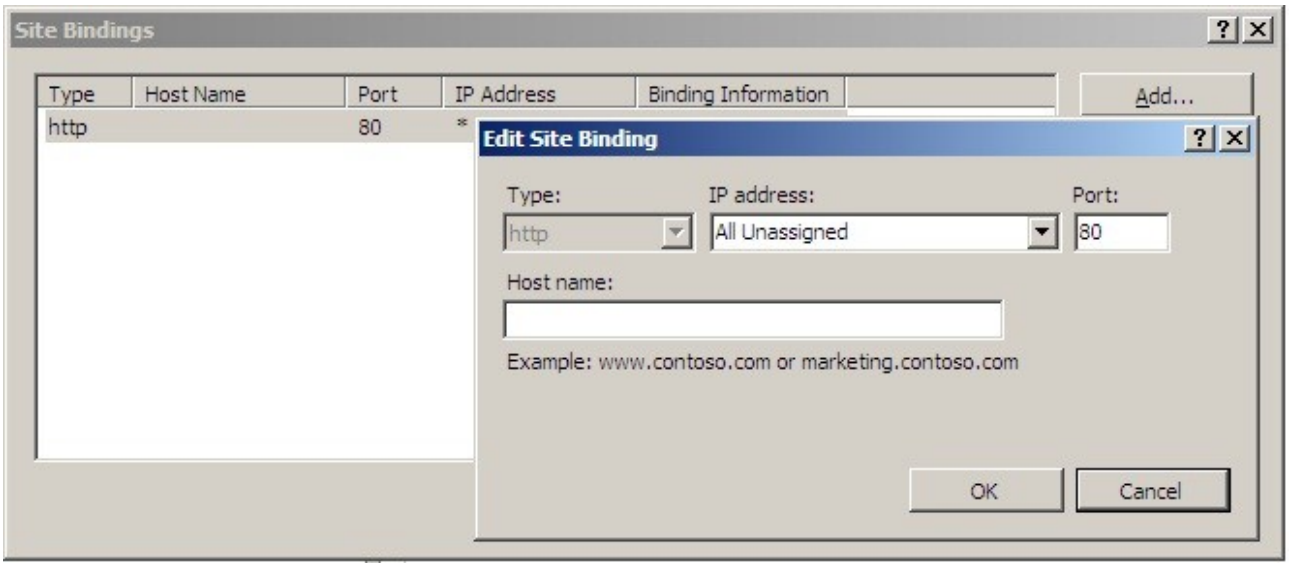
If you are using a one-arm DR load balancing method, each web server requires the ARP problem to be handled:

- Each real server must be configured to respond to the VIP address as well as the RIP address. This is because in DR mode load balanced traffic arrives on the VIP address but the load balancer health checks are on the real IP address
- Each real server must have the MS loopback adapter installed and configured
- The MS loopback adapter must be configured to deal with the ARP problem

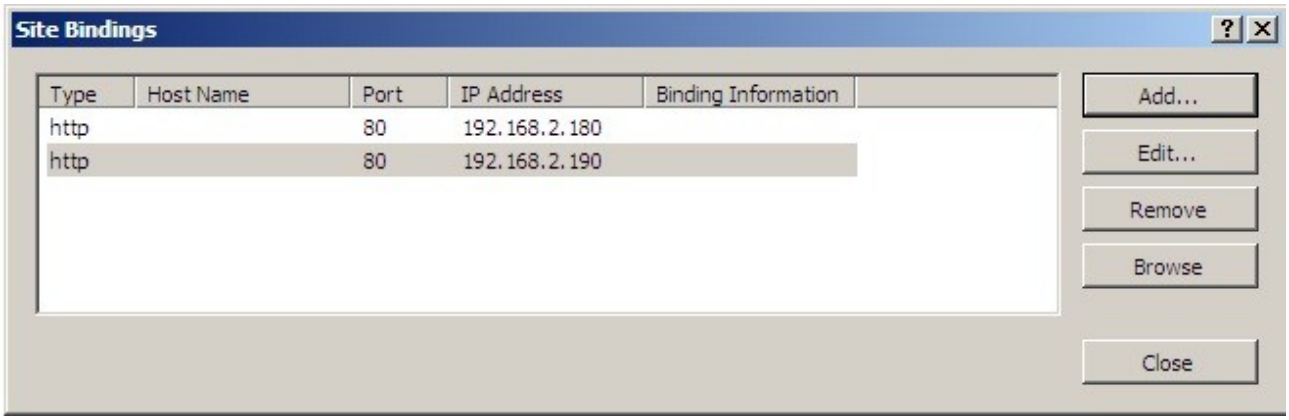
Remember that for all real servers in Direct Routing mode the load balanced application must respond to both the virtual IP as well as the servers real IP. With Windows IIS the IP address must either be set to (All Unassigned) or use the Advanced tab to add a second IP address.

Configuring IIS to respond to both the RIP and VIP

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



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

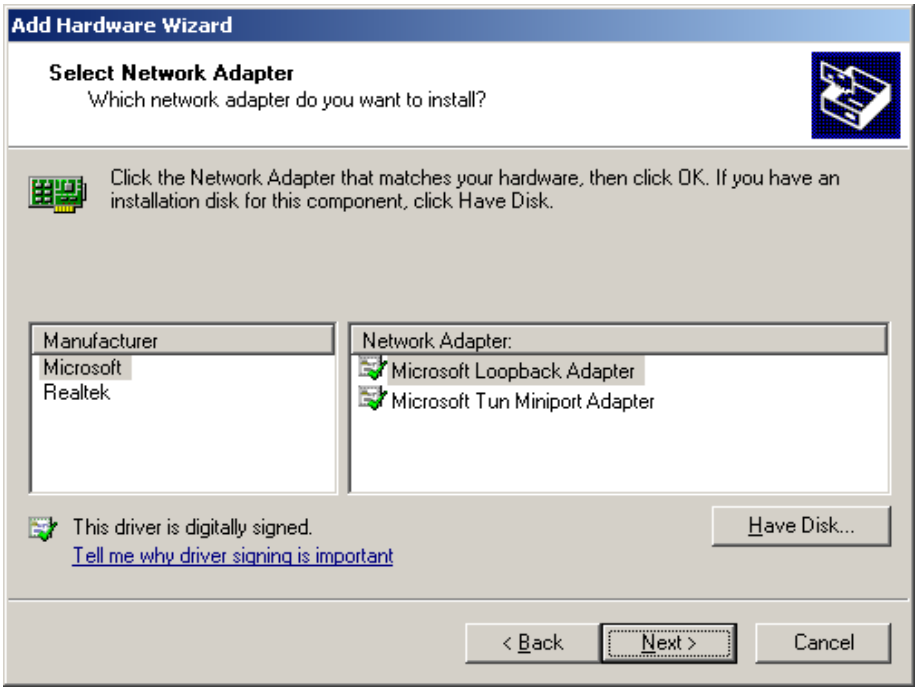


Resolving ARP issues for Windows server 2000 / 2003 (DR mode only)

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

Installing the Microsoft loopback adapter

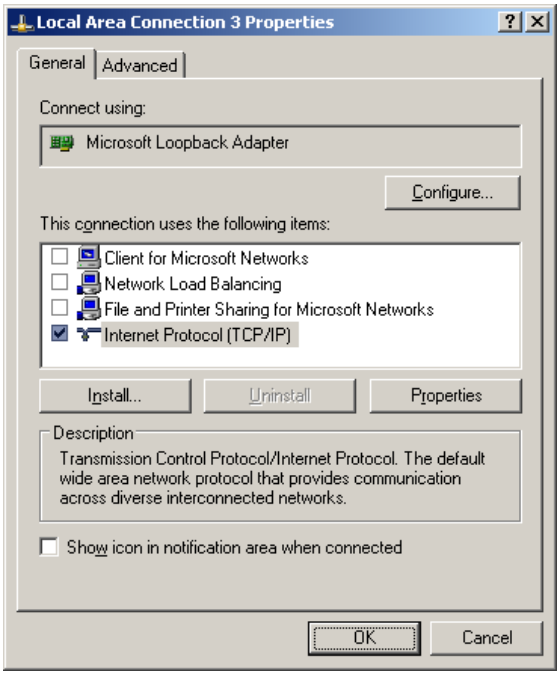
1. Open the Control Panel and double-click Add Hardware
2. Once the Hardware Wizard opens, click Next
3. Select 'Yes, I have already connected the hardware', click Next
4. Scroll to the bottom of the list, select 'Add a new hardware device' and click Next
5. Select 'Install the hardware that I manually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



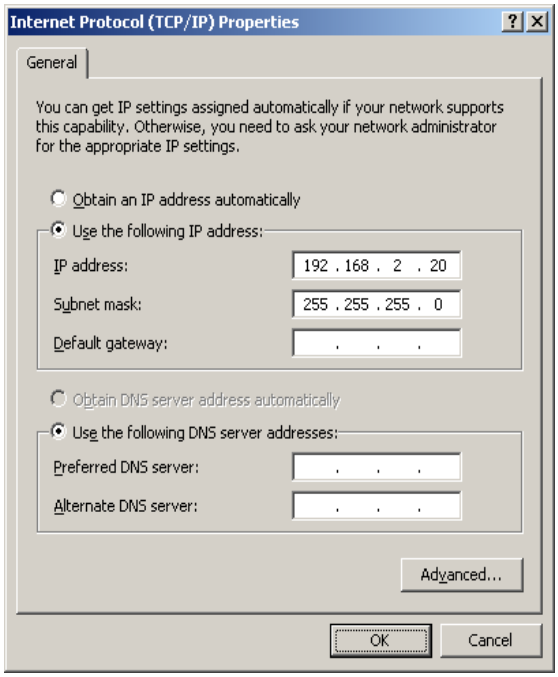
8. Click Next to start the installation, when complete click Finish

Configuring the loopback adapter

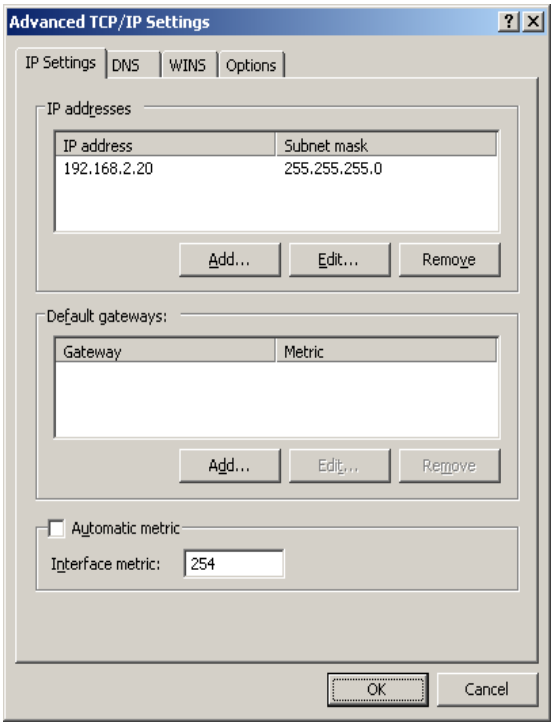
1. Open the Control Panel and double-click Network Connections
2. Right click the new loopback adapter and select properties



3. Un-check all items except Internet Protocol (TCP/IP)
4. Select Internet Protocol (TCP/IP), click Properties and configure the IP address to be the same as the Virtual Server (VIP), e.g. 192.168.2.20/24



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



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

7. Now repeat the above process for all other Windows 2000 / 2003 real servers

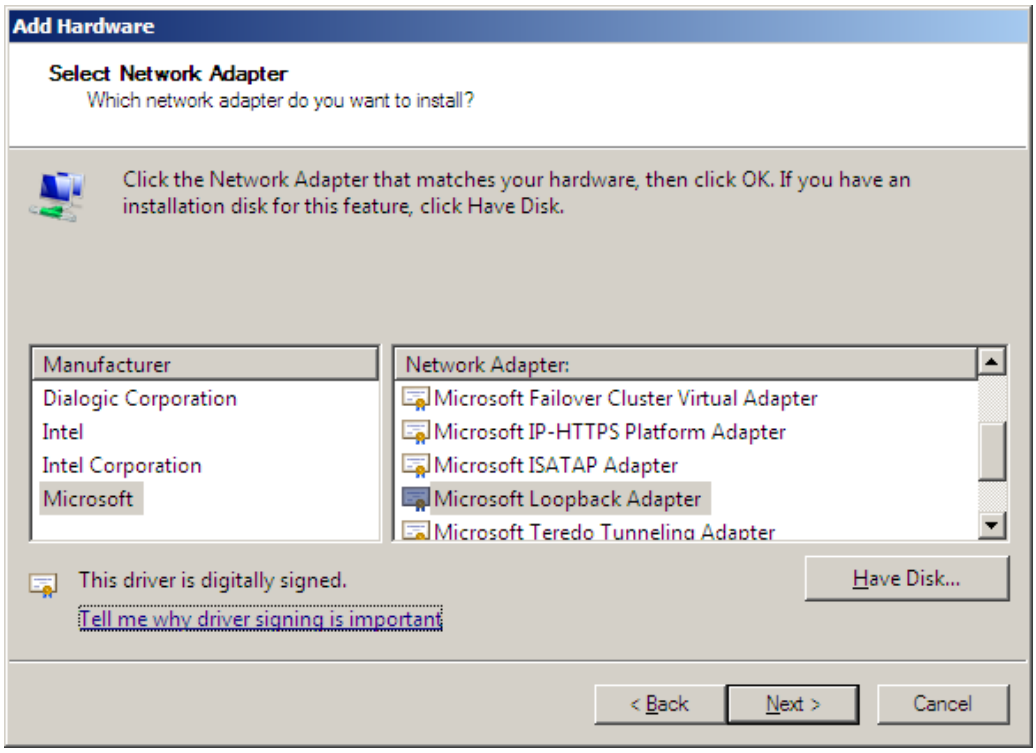
For Windows server 2003 SP1 & above, if you have enabled the built-in firewall, you will need to create an exception to enable access to the web server. This exception by default will allow traffic on both the network and loopback adapters.

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

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

Installing the Microsoft loopback adapter

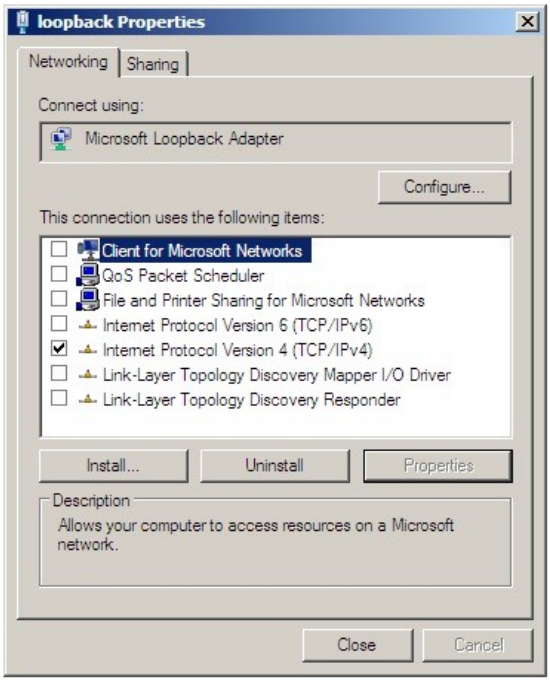
1. Click Start, select Run and enter ***hdwwiz*** to start the Hardware Installation Wizard
2. When the Wizard has started, click Next
3. Select 'Install the hardware that I manually select from a list (Advanced)', click Next
4. Select 'Network adapters', click Next
5. Select 'Microsoft' & 'Microsoft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



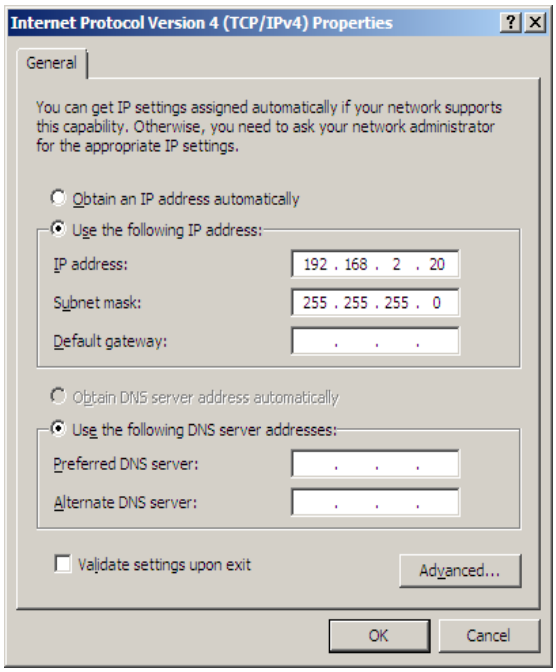
6. Click Next to start the installation, when complete click Finish

Configuring the loopback adapter

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



4. Un-check all items except Internet Protocol Version 4 (TCP/IPv4)
5. Select Internet Protocol Version (TCP/IPv4), click Properties and configure the IP address to be the same as the Virtual Server (VIP), e.g. 192.168.2.20/24



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

7. Now repeat the above process for all other Windows 2008 real servers

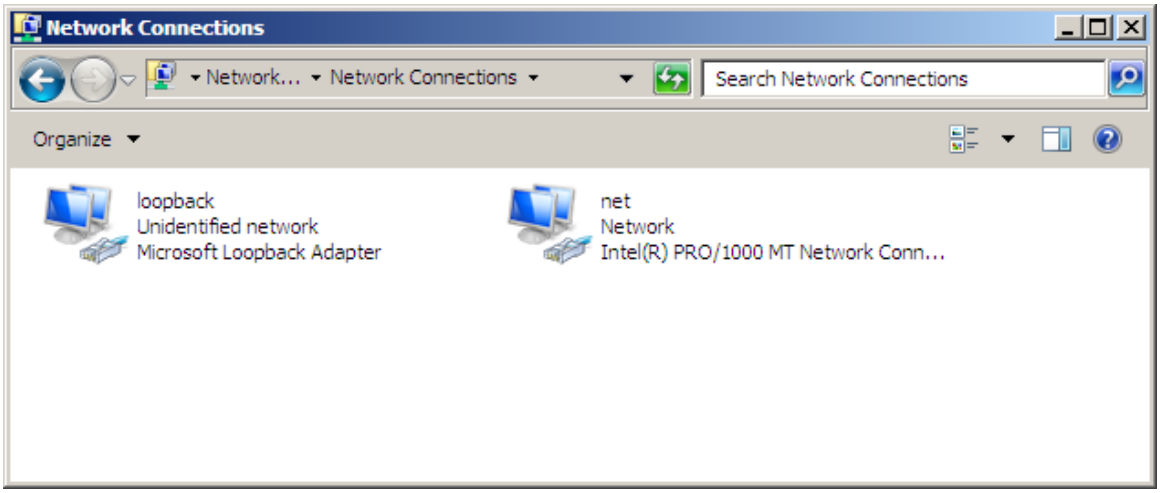
Configuring strong / weak host behavior

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

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

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

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



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

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

```
netsh interface ipv4 show interface
```

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

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

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

Testing the load balancer configuration

For testing add a page to each real web servers root directory e.g. test.html and put the server name on this page.

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

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

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

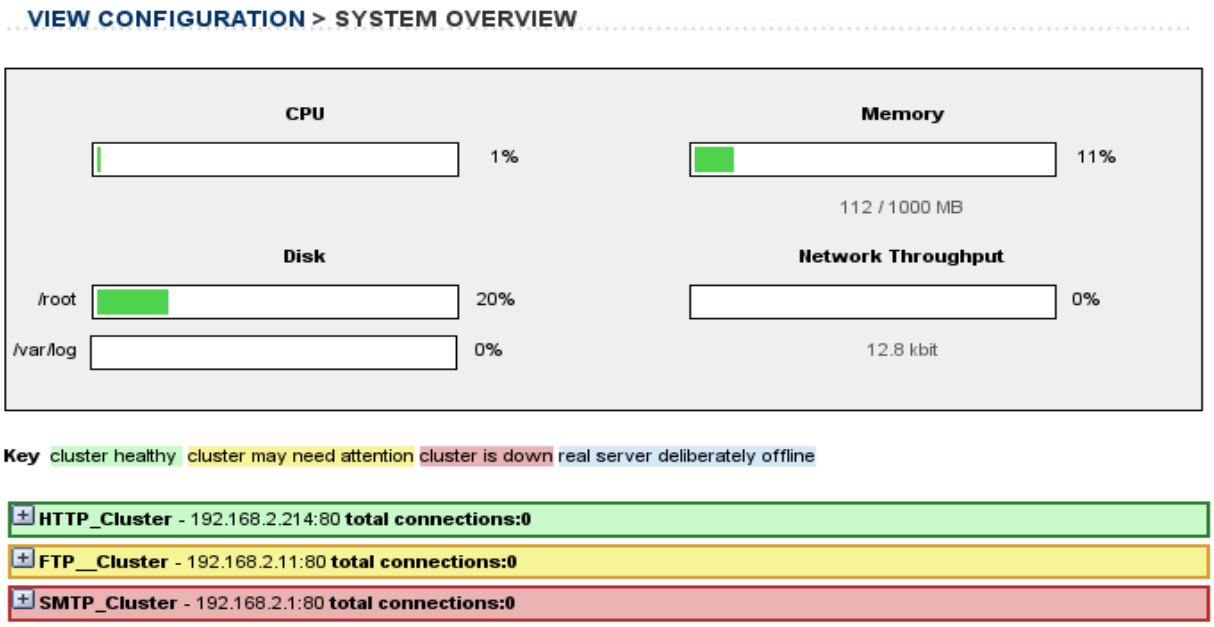
When using a two-arm NAT load balancing method the test client must be in the external subnet.

Connection error diagnosis

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

1. Check *View Configuration* > *Network Configuration* and make sure that the VIP is active on the load balancer, if not check *Logs* > *Heartbeat* for errors.

2. Check *Maintenance* > *System Overview* and make sure none of your VIPs are highlighted in red. If they are, your cluster is down and you should see health check diagnosis (next page). Green indicates a healthy cluster, yellow indicates that your cluster may need attention (one real server may be down), and blue indicates a real server has been deliberately taken offline.



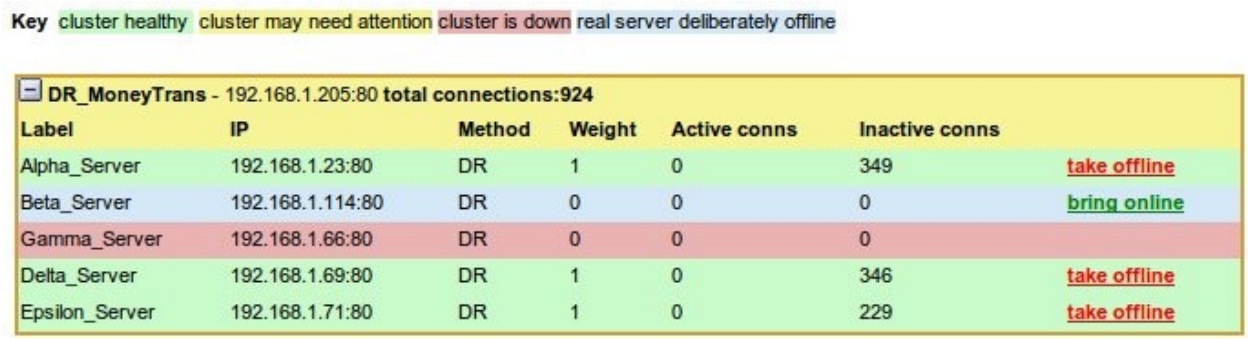
3. If the VIP is still not working then check *Reports* > *Current Connections* to see the current traffic in detail, any packets marked SYN_RECV imply incorrect real server configuration. Check that any NAT mode servers have the correct default gateway and any DR mode servers are responding to the VIP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

Health check diagnosis

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

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

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



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

DR_MoneyTrans - 192.168.1.205:80 **total connections:924**

Label	IP	Method	Weight	Active conns	Inactive conns	
Alpha_Server	192.168.1.23:80	DR	1	0	349	take offline
Beta_Server	192.168.1.114:80	DR	0	0	0	bring online
Gamma_Server	192.168.1.66:80	DR	0	0	0	
Delta_Server	192.168.1.69:80	DR	1	0	346	take offline
Epsilon_Server	192.168.1.71:80	DR	1	0	229	take offline

The example above shows that the requested status of Gamma_Server is down (red). This implies that the real server has failed a health check; you can investigate this using *Logs* > *Ldirectord*. If you know the real server should be active, you may need to increase the health check time-out *Edit Configuration* > *Global Settings*.

The Beta_Server however is blue, this indicates that it is deliberately in maintenance mode. You can use 'bring online' to make it active.

Testing high-availability for a Loadbalancer.org HA-pair

To test fail-over of a clustered pair of load balancers, power down the master and check that the slave unit takes over all the floating IP(s).

If fail-over does not occur correctly check *Logs* > *Heartbeat* on both nodes for any errors.

When testing load balancer fail-over, do not just pull the serial cable and network cable out. These will not cause a fail-over and will invalidate the cluster configuration (split brain). You can configure fail-over on network failure but this is not enabled by default. To enable this, a ping node must be configured under *Edit Configuration* > *Modify Heartbeat Configuration* (see Sections E & G for more details).

Does your application cluster correctly handle its own state?

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

Web based applications are inherently stateless and an ideal candidate for load balancing. However, Do your web servers store persistent information on local drives?

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

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

Replication solutions for shared data

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

Solutions for session data

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

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

What do you do if your application is not stateless?

Some applications require state to be maintained such as:

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

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

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

Loadbalancer.org persistence methods

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

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

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

Section D – Typical Deployment Examples

Example 1 – single appliance (web interface)

This section deals with the process of configuring a single load balancer appliance via the web interface, rather than using the console wizard.

Network interface configuration

- Power up the load balancer
- Log in to the console:

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

You can access the web interface either via links at the console or from a web browser on a client connected to the same network (recommended). The default IP address is *192.168.2.21/24.* To change this, at the console use:

```
ifconfig eth0 <IP address> <netmask> up
```

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

Accessing the Web User Interface (WUI)

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

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

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

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

- Use *Edit Configuration > Network Interface Configuration*
- Specify the IP address, Netmask & Default Gateway
- Use *Edit Configuration > DNS & Hostname*
- Specify the DNS server
- Now refer to page 43 onwards to configure the virtual & real servers

When you are only configuring a single load balancer in two-arm NAT mode you can use the IP address you configure for both administration as well as for the VIP. Please note however that if possible it's good practice to use an extra dedicated floating IP for the VIP to make adding a 2nd paired unit much easier.

If you are using two-arm NAT mode you must also configure an Internal IP on eth0. The Real servers will need to use this as their default gateway.

If you have not used the wizard then the web interface will default the forwarding method for all new virtual and real servers to DR mode, make sure you change this to NAT mode if required in global settings.

Example 2 - clustered pair (web interface)

This section deals with the process of configuring the load balancers as a clustered pair via the web interface, rather than using the console wizard. In this scenario, the slave's network settings must be configured first, followed by the master. This ensures that the master can communicate with the slave and replicate settings as they are configured.

Network interface configuration

- Connect the serial interface cable between the master & slave
- Connect both master & slave to the network
- Power up the **slave** load balancer first
- Log in to the console:

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

You can access the web interface either via links at the console or from a web browser on a client connected to the same network (recommended). The default IP address is *192.168.2.21/24.* To change this, at the console use:

```
ifconfig eth0 <IP address> <netmask> up
```

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

Accessing the Web User Interface (WUI)

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

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

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

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

- Use *Edit Configuration > Network Interface Configuration*
- Specify the IP address, Netmask & Default Gateway
- Use *Edit Configuration > DNS & Hostname*
- Change the hostname from lbmaster to lbslave
- Now power up the master load balancer
- After the master has booted, ensure that its IP address is difference than the slaves IP address – this can be done using the ifconfig command shown above. Then using a browser connect to the web interface using **http://<IP address>:9080/lbadmin/**
- Use *Edit Configuration > Network Interface Configuration*
- Specify the IP address, Netmask & Default Gateway
- Use *Edit Configuration > DNS & Hostname*

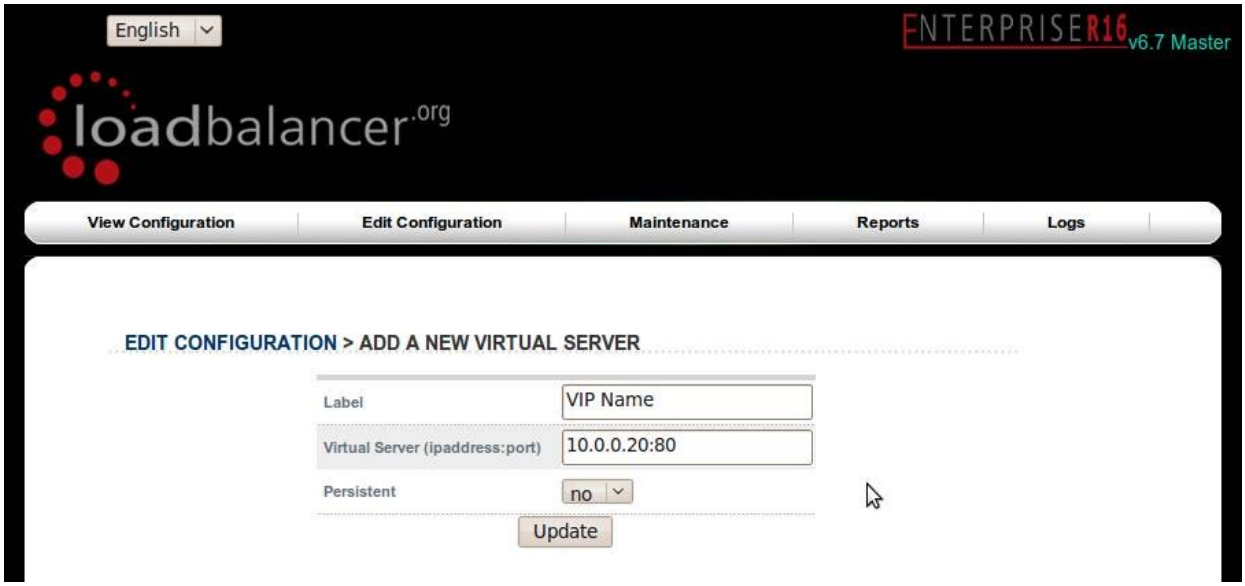
- Specify the IP address of the slave load balancer. Any changes to the configuration of the master load balancer will now be automatically replicated to the slave
- Now refer to the following section and subsequent pages to configure the virtual & real servers

If you are using two-arm NAT mode you must also configure an IP address on eth0. Then configure an internal Floating IP for the Real Servers to use this as the default gateway.

Configuring the virtual servers (VIP) in one-arm DR mode

Layer 4 configuration

- You need to tell the master load balancer which service you want to load balance
- Use *Edit Configuration > Logical Layer 4 Configuration > Virtual Servers > Add a new Virtual Server*
- The Virtual Servers are added in the following format *IPAddress:Port*. It basically means that any packet arriving at the load balancer with that IP address and that port number will be handled by the real servers associated with this virtual server.

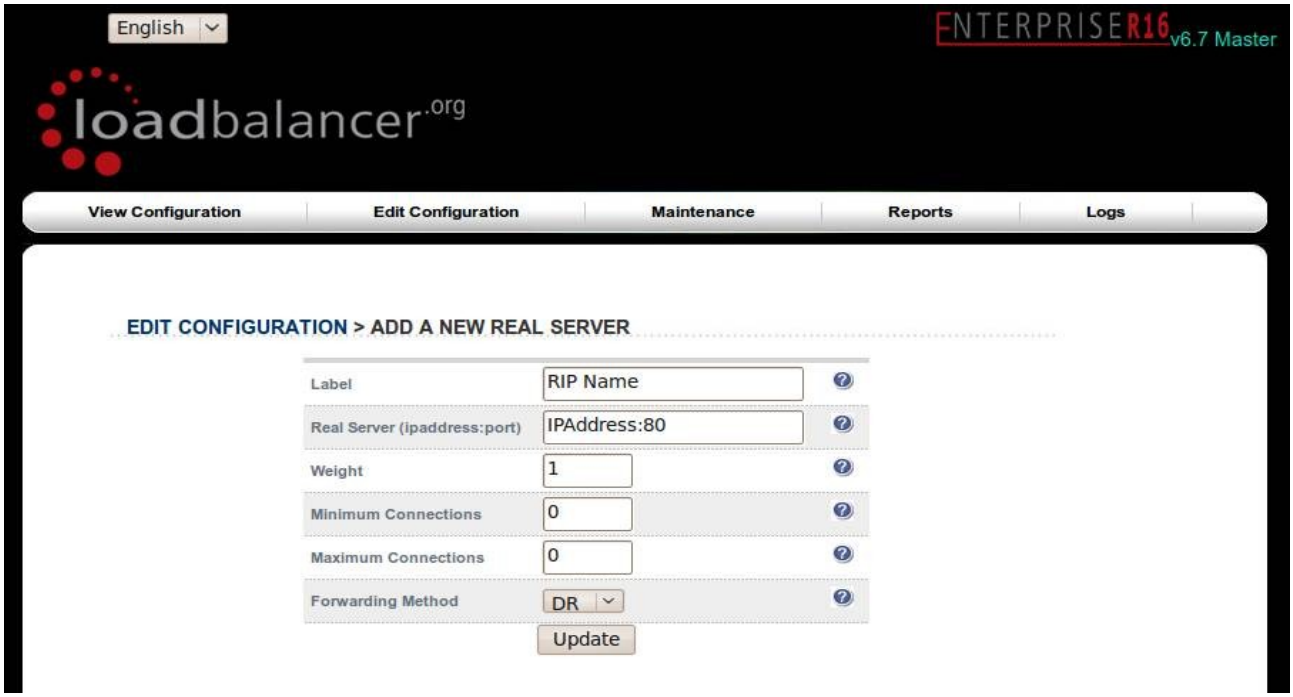


- For this example we are load balancing both HTTP and HTTPS so you need to set up 2 Virtual Servers, e.g. 192.168.1.20:80 and 192.168.1.20:443
- For this example persistence is only recommended for the HTTPS virtual server
- Once you have set up your Virtual Servers you will need to add your Real Servers to the cluster

Real server configuration (RIP)

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

- Use either *Edit Configuration* > *Layer 4 Configuration* > *Real Servers* > *Add a new Real Server* or, click the [**Real Servers**] link on the *Virtual Servers* configuration page
- Against the relevant Virtual server, click *Add a new Real Server*
- You just need to give the IP Address and Port number of your real server



- For the HTTP Virtual Server add the real servers as 192.168.1.50:80 & 192.168.1.60:80
- For the HTTPS Virtual Server add the real servers as 192.168.1.50:443 & 192.168.1.60:443
- Weight defaults to 1 making real servers active immediately
- Leave the Minimum & Maximum connections as 0 for unrestricted
- The Forwarding Method should default to DR if you have a one-arm configuration

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

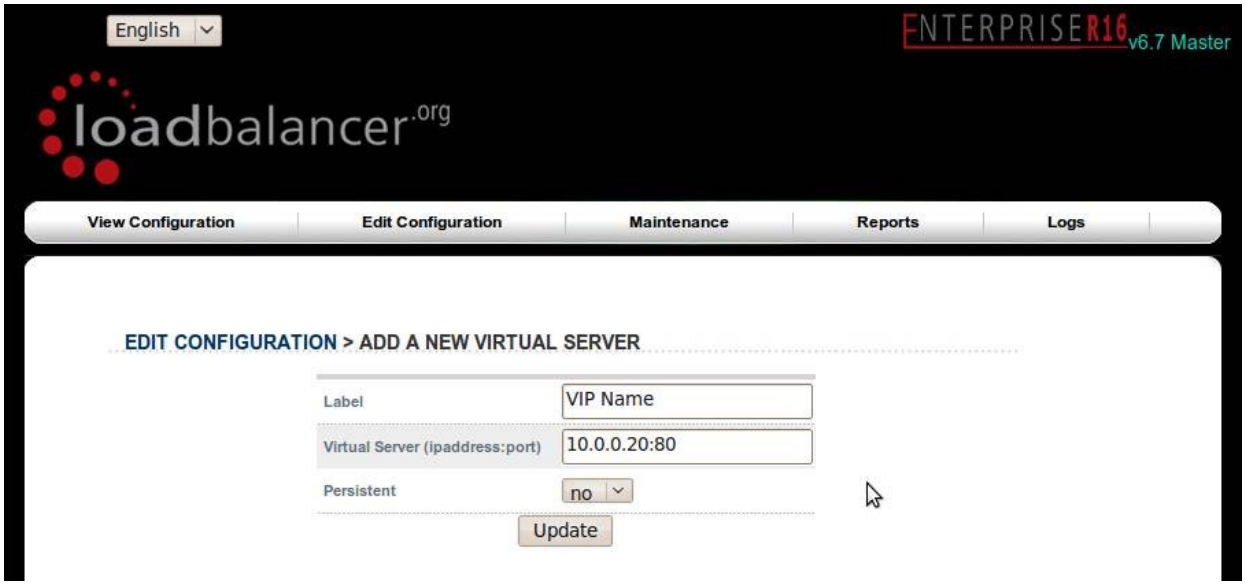
- Each server must be configured to respond to the VIP address as well as the RIP address
- Each Windows server must have the MS Loopback Adapter installed and configured
- The MS Loopback Adapter must be configured to deal with the ARP problem

Failure to correctly configure the real servers to handle the ARP problem is the most common problem in DR configurations. Please refer to pages 27-35 & 59-67 for more details.

Configuring the virtual servers (VIP) in two-arm NAT mode

Layer 4 configuration

- You need to tell the master load balancer which service you want to load balance
- Use *Edit Configuration* > *Logical Layer 4 Configuration* > *Virtual Servers*
- The Virtual Servers are added in the following format *IPAddress:Port*. It basically means that any packet arriving at the load balancer with that IP address and that port number will be handled by the real servers associated with this virtual server

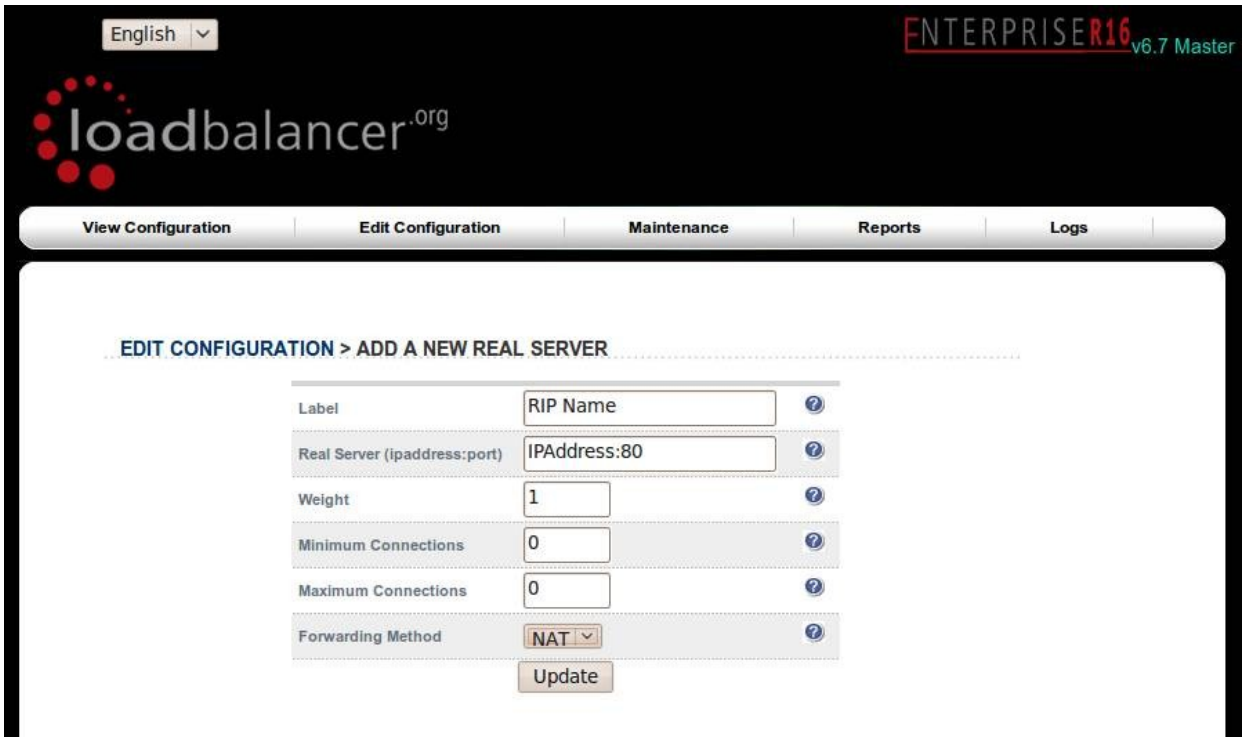


- For this example we are load balancing both HTTP and HTTPS so you need to set up 2 Virtual Servers, for example 10.0.0.20:80 and 10.0.0.20:443
- For this example persistence is only recommended for the HTTPS virtual server
- Once you have set up your Virtual Servers you will need to add your Real Servers to the cluster

Real server configuration (RIP)

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

- Use either *Edit Configuration* > *Layer 4 Configuration* > *Real Servers* > *Add a new Real Server* or, click the [**Real Servers**] link on the *Virtual Servers* configuration page



- You just need to give the IP address and port number of your web server
- For the HTTP Virtual Server add the real servers as 192.168.1.50:80 & 192.168.1.60:80
- For the HTTPS Virtual Server add the real servers as 192.168.1.50:443 & 192.168.1.60:443
- Weight defaults to 1 making real servers active immediately
- Leave the Minimum & Maximum connections as 0 for unrestricted
- The Forwarding Method should default to NAT if you have a two-arm configuration

When using a two-arm NAT load balancing method each web server has to be in the same subnet as the internal load balancer and the default gateway must point at the load balancer.

Example 3: layer 7 configuration one-arm SNAT mode (HAProxy)

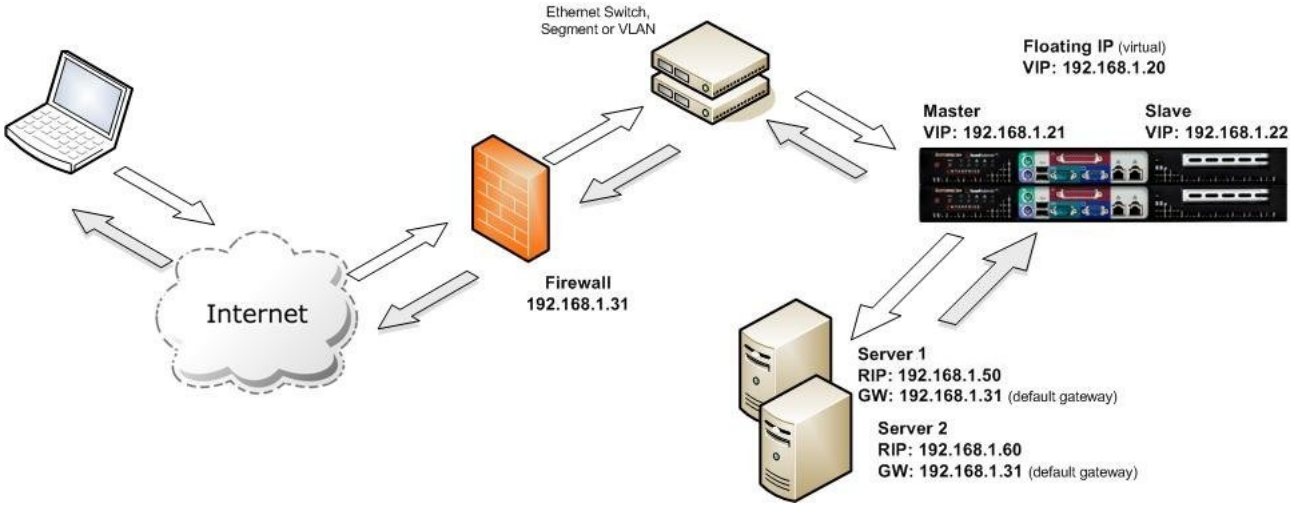
For this example we are going to assume that the e-commerce application does not support persistence. We are going to decrypt the SSL traffic on the load balancer, insert or read the session cookies as required and pass the traffic the the real servers in plain unencrypted HTTP.

1. The Firewall will translate all traffic for the web sites public IP address and the the load balancers floating VIP (192.168.1.20).

2. The load balancer (Pound) will terminate SSL traffic to 192.168.1.20:443 and re-direct it to 192.168.1.20:80 using a valid uploaded SSL certificate.

3. The load balancer (HAProxy) will handle cookie insertion/reading on all traffic through 192.168.1.20:80 and ensure each client goes to the correct server.

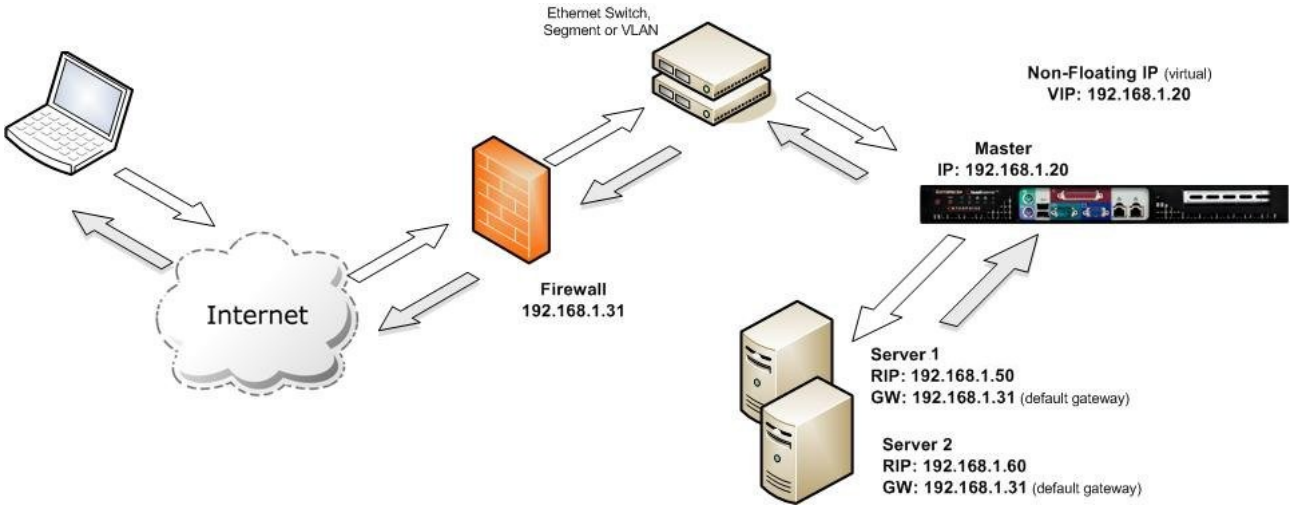
Network diagram for layer 7 SNAT mode (clustered pair)



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

NB. You can configure your web server logs to parse the X-Forwarded-For header to find the client source IP, or see the section on TPROXY in Section G.

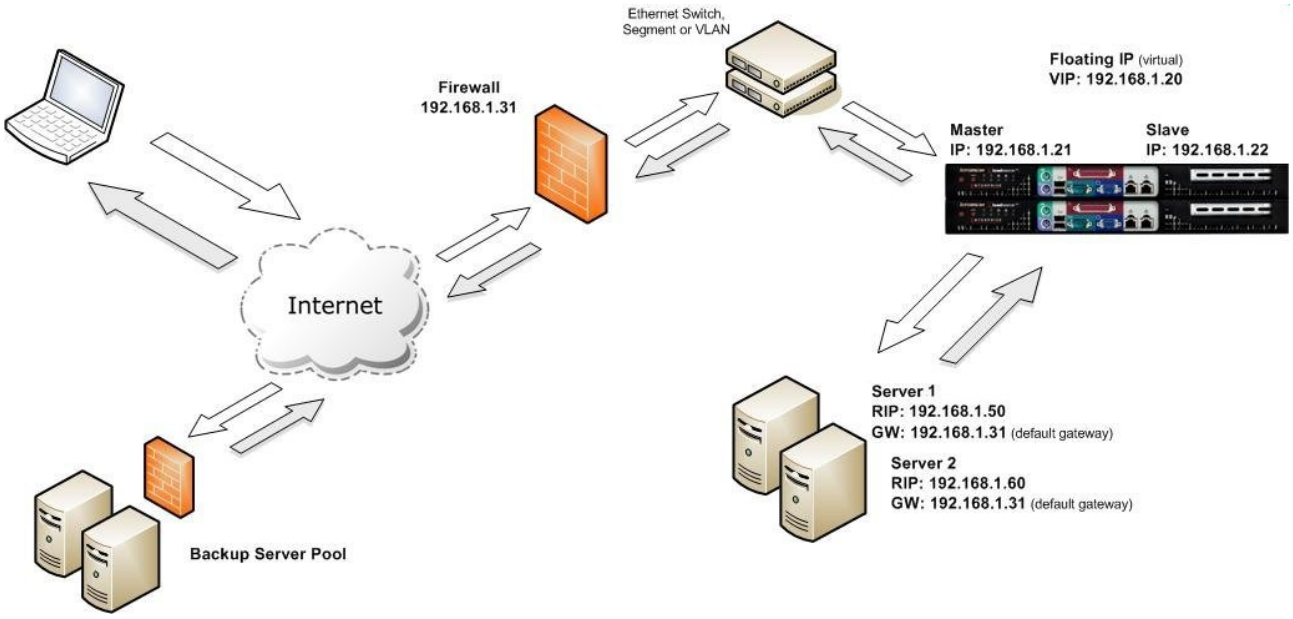
Network diagram for layer 7 SNAT mode (single unit)



Notes:

- As with other modes a single unit does not require a Floating IP. Please note however that it is still good practice to use an extra dedicated floating IP to make adding a 2nd unit (to form a clustered pair) much easier
- SNAT is a full proxy and therefore load balanced servers do not need to be changed in any way

Network Diagram for layer 7 SNAT mode (off site backup)



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

Virtual server configuration

- You need to tell the master load balancer which service you want to load balance. Go to *Edit Configuration > Virtual Servers (HAProxy)*
- Add a new virtual server. The virtual server is added in the following format *ipaddress:port*. It basically means that any packet arriving at the load balancer with that IP address and that port number will be handled by the Real Servers associated with this Virtual Server

- Configure the Virtual Server as 192.168.1.31:80
- Set persistence to 'Cookies'
- Set the Fall-back server as required, this is where requests go if all servers in the cluster are down
- Click the button to add the new Virtual Server to the HAProxy configuration file
- 192.168.1.31 will be automatically added as a Floating IP and activated

NB. HAProxy won't be activated until you add your first back-end server.

Real server configuration

- Use *Edit Configuration* > *Real Servers (HAProxy)* and you should see your Virtual Servers listed, select the one you want and click on **Add a new Real Server**

- You just need to give the IPA*ddress:Port* of your web server and specify a relative weight

- The HAProxy service will be activated as soon as you add the first back-end server
- Add as many real servers as required
- You have now finished configuration of the load balancer

IMPORTANT: The label is used as the cookie so make sure it is different for each server.

SSL termination configuration (Pound)

In order to set up a proxy for the SSL traffic go to Edit Configuration > SSL Termination (Pound).
It is common for SSL traffic to be terminated and then re-directed to port 80 of the same VIP for HAProxy to pick it up insert cookies and load balance it.

- Add a new Virtual Server for SSL termination (Pound)

- Configure the Virtual Server as 192.168.1.20:443
- Configure the Back-end as 192.168.1.20:80
- Click the button to add the new Virtual Server to the Pound configuration file
- Ignore the ciphers setting for now

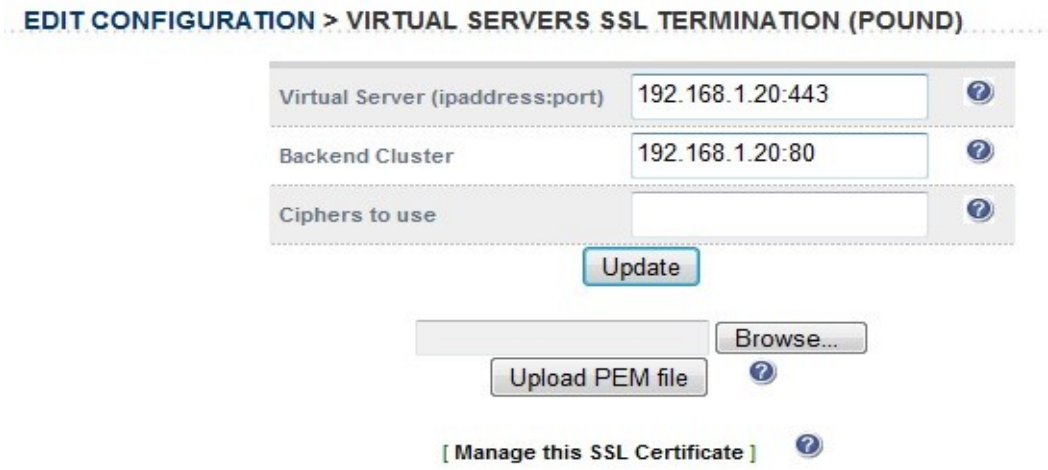
By default a self generated SSL certificate is associated with the new Virtual Server. You can upload your own valid certificate by selecting modify for the Virtual Server. Just browse your local machine for the *cert.pem* file and click the upload button.

IMPORTANT: You must restart the Pound service in order to activate the changes i.e. *Maintenance > Restart Pound SSL*

Manage SSL certificate

In order to get a proper signed certificate from a certificate authority such as Verisign or Thawte you will need to generate a certificate request (CSR).

Go to Edit Configuration > SSL Termination (Pound), then click [**Modify**] next to the relevant Virtual Server, then click [**Manage this SSL Certificat**e].



The following default information is displayed. This should be changed as required.

When you have entered your correct details, click **Manage this SSL Certificate (Server0)**, the CSR is generated for you and the following is displayed:

Upload Signed Certificate

NB. For subsequent Pound VIPS that are created, server0 will become server1 etc.

NB. Make sure you backup, i.e. save to a text file both the CSR & the Private Key.

You can now copy the *Certificate Signing Request* (CSR) from the top pane and provide it to your Certificate Authority. They in turn will then sign the Certificate which you should paste into the second pane of the form and then click upload.

Once the signed key is uploaded you will need to restart Pound-SSL

It is critically important to backup all of these files! You need a copy of the CSR, SIGNEDCERT & PRIVATE KEY.

For more detailed information on SSL termination and Pound configuration, please refer to the SSL Certificates & Pound topics in sections E & G

Section E – Detailed Configuration Information

Console configuration methods

The load balancer can be configured locally from either the bash shell, or using a text based web browser locally such as links.

- Login to the console:

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

SECURITY: It is recommended to change the default password . To do this type *passwd* at the console to change the default root password

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

The following configuration files may be useful:

Physical configuration:	/etc/sysconfig/network-scripts/ifcfg-eth0
Firewall configuration:	/etc/rc.d/rc.firewall
Logical configuration:	/etc/ha.d/conf/loadbalancer.cf
HA-Proxy configuration	/etc/haproxy/haproxy.cfg
Pound SSL configuration	/usr/local/etc/pound.cfg
SSL Certificates	/usr/local/etc/
Fail-over configuration:	/etc/ha.d/ha.cf

For configuration at the console just use: **links 127.0.0.1:9080/lbadmin**

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

You will be prompted for a password and username, the default for both is 'loadbalancer'.

Usually you would just use links to navigate to *Edit Configuration* > *Network Interface Configuration* and then change the IP address on the primary interface for easy access from your client web browser.

Or you could just use the following command:

```
ifconfig eth0 192.168.1.21 netmask 255.255.255.0 broadcast 192.168.1.255 up
```

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

Console access via a serial cable

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

Remote configuration methods

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

- OpenSSH or PuTTy — Secure Shell Access
- OpenSCP or WinSCP — Secure File Transfer
- HTTP or HTTPS — Web based Administration

NB. The default IP address for the Loadbalancer.org appliance is 192.168.2.21/255.255.255.0

For SSH and SCP login as *root* using the password: *loadbalancer.* The Web based Administration interface uses a different set of user accounts and passwords based on the simple .htaccess files. This allows you to set up users in three groups, configuration, maintenance and reports.

To access the web based administration interface use : ***http://ApplianceIPaddress:9080/lbadmin/***

You will be prompted for a password and username, the default for both is 'loadbalancer'.

Which should bring up the following screen:

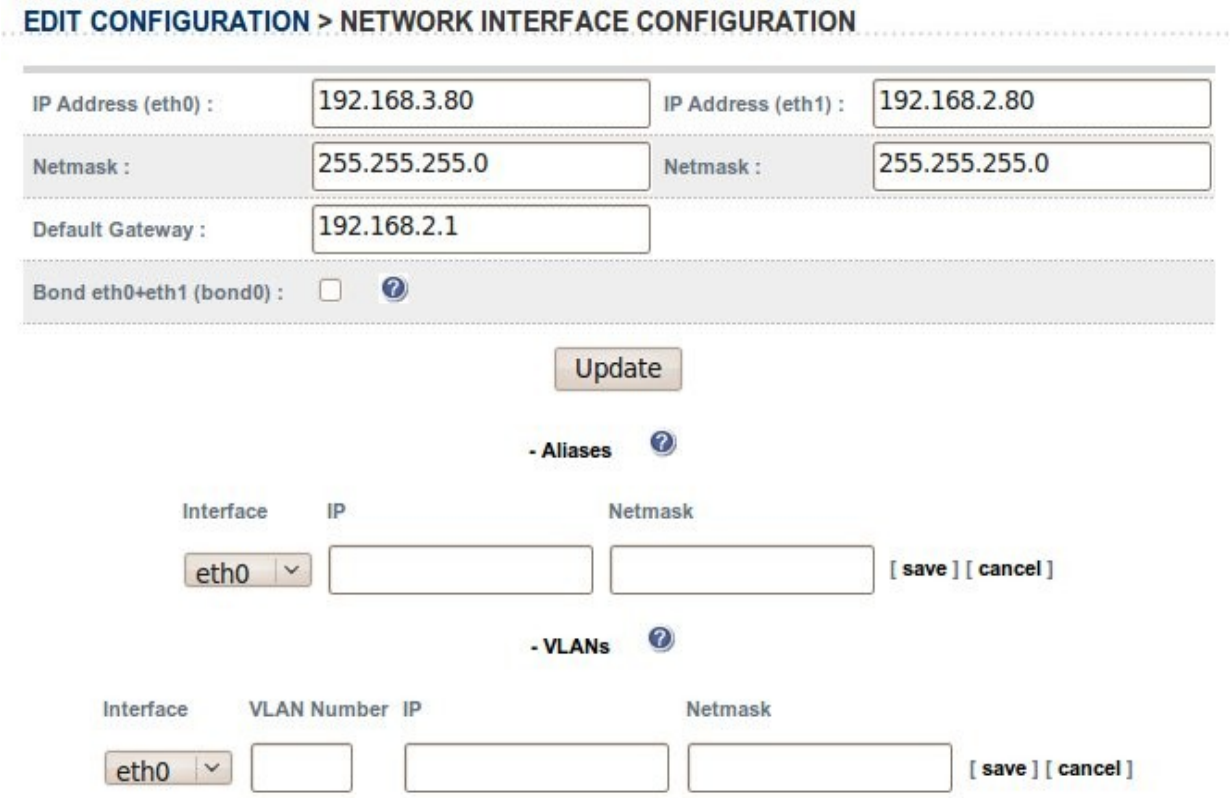
You can then select an option from one of the main menus. The menu options are as follows:

- **View Configuration :** View the network & load balancer configuration
- **Edit Configuration :** Set up or modify the physical and virtual configuration
- **Maintenance :** Take servers offline or bring them back online
- **Reports:** View the actual live status of the load balancer or historical statistics
- **Logs:** View Ldirectord, Lbadmin, Heartbeat, HAProxy and Pound (SSL)

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

Network interface configuration

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



Unlike other appliances on the market you can use any interface for any purpose, eth0 does not always have to be the internal interface (but it usually is). In a standard one arm configuration you would just need to configure eth0, the netmask and the default gateway.

If you tick the *Bond eth0+eth1 (bond0)* check box the system will automatically ensure that both network ports can be used for the eth0 IP address i.e. master / slave bonded network for high-availability.

For a two arm configuration you would normally configure eth0, the netmask and the default gateway + eth1 and its netmask.

NB. If you are configuring NAT mode manually don't forget to set the autonat interface to eth1 in global options.

If you wish to have more than one IP address / network associated with an interface then simply add an alias to that interface as required.

In the same way you can configure as many VLANs as needed on multiple interfaces (see Section G for more details).

The Floating IP addresses must have a physical interface within the same network in order to bind correctly. So if you want a Floating IP in a new VLAN you must configure the physical IP first.

Advanced DR considerations

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

What is the ARP problem?

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

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

Simple DR configuration examples are available in section C - the Quick Start Guide, at the start of this manual.

Solving the ARP problem

Solving for Linux (with iptables)

This is the recommended method for Linux. You can use iptables (netfilter) on the real server to re-direct incoming packets destined for the virtual server IP address. This is a simple case of adding the following command to your start up script (rc.local):

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

i.e. Redirect any incoming packets destined for 10.0.0.21 (virtual server) to my local address.

(Don't forget to change the IP address to be the same as your virtual server)

Solving for Linux – althernative method (with arp_ignore sysctl values)

Each real server needs a loopback IP address to be configured as the VIP. This address needs to be stopped from responding to ARP requests and the web server needs to be configured to respond to this IP address.

With most modern Linux kernels (>2.6) you can alter the ARP behavior allowing you to configure a loopback adapter without worrying about ARP issues. To do this just add the following lines to /etc/sysctl.conf and re-boot, or run /sbin/sysctl.conf -p to reload the file:

```
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
```

Alternatively, the following commands may be used to change the settings interactively during runtime:

```
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
```

Once you have configured your Linux real server so that it won't respond to ARP requests for the loopback adapter you can configure your VIPs as follows:

```
ifconfig lo:0 VIP netmask 255.255.255.255 up
```

To make this permanent and reboot safe you may include this command in *rc.firewall* or in a equivalent customizable start-up script.

Solving for Solaris

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

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

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

Solving for Mac OS X or BSD

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

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

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

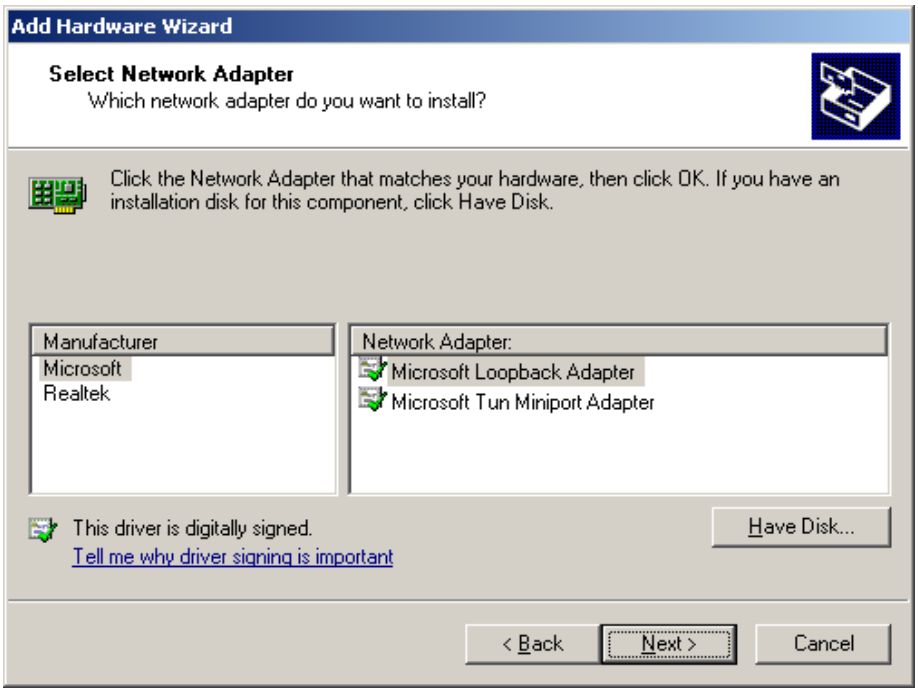
Solving for Windows 2000 / 2003

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

Installing the Microsoft loopback adapter

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

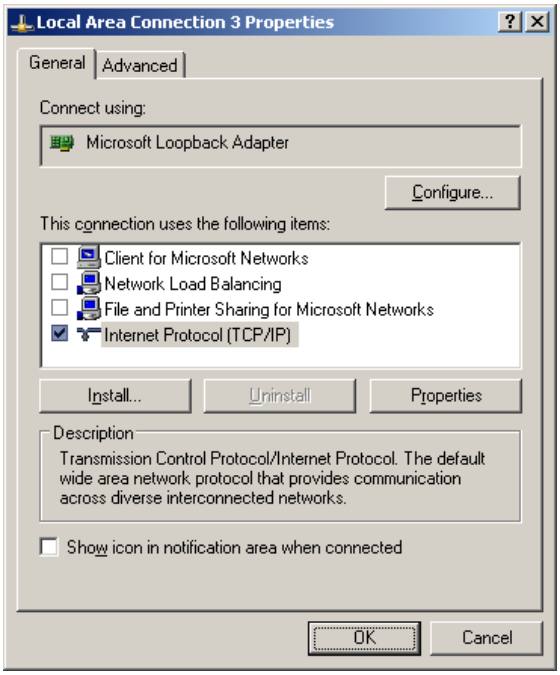
6. Select 'Network adapters', click Next
7. Select 'Microsoft' & 'Microsoft Loopback Adapter', click Next



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

Configuring the loopback adapter

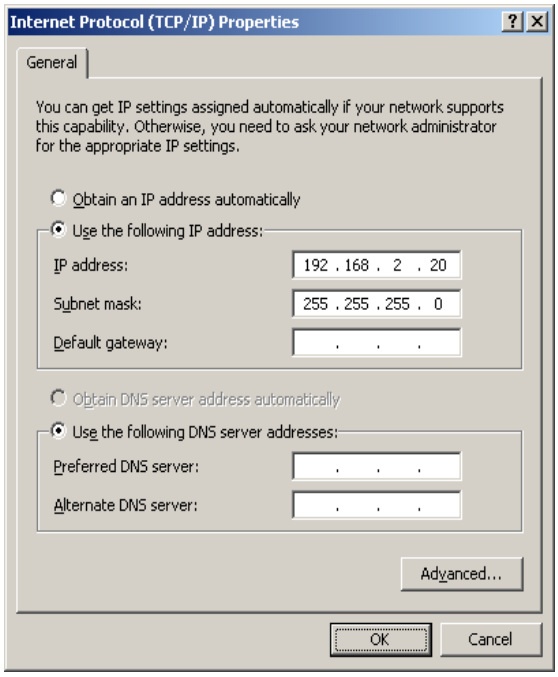
1. Open the Control Panel and double-click Network Connections



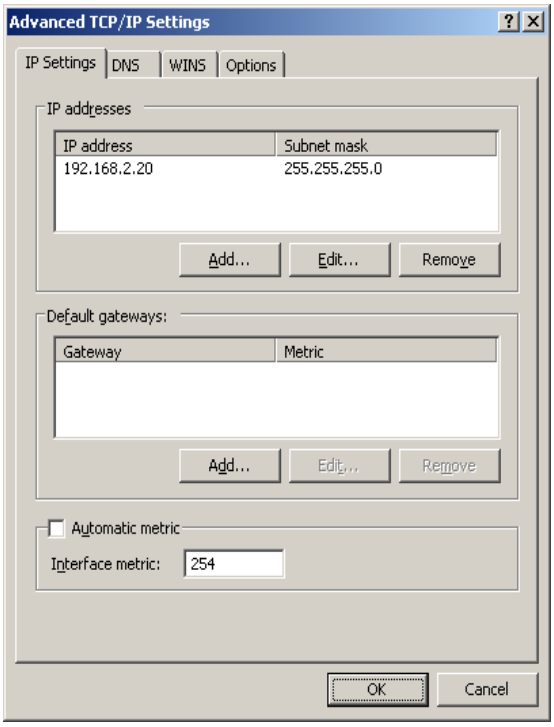
2. Right click the new loopback adapter and select properties

3. Un-check all items except Internet Protocol (TCP/IP)

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



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



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

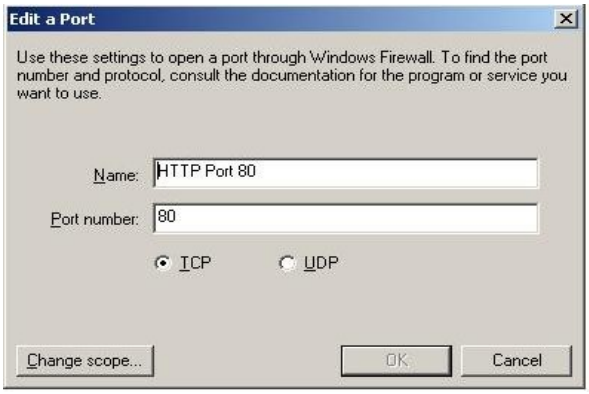
7. Now repeat the above process for all other Windows 2000 / 2003 real servers

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

Windows 2003 R2 / R1 (With SP1) Firewall Settings

Windows 2003 only allows control of inbound connections.

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

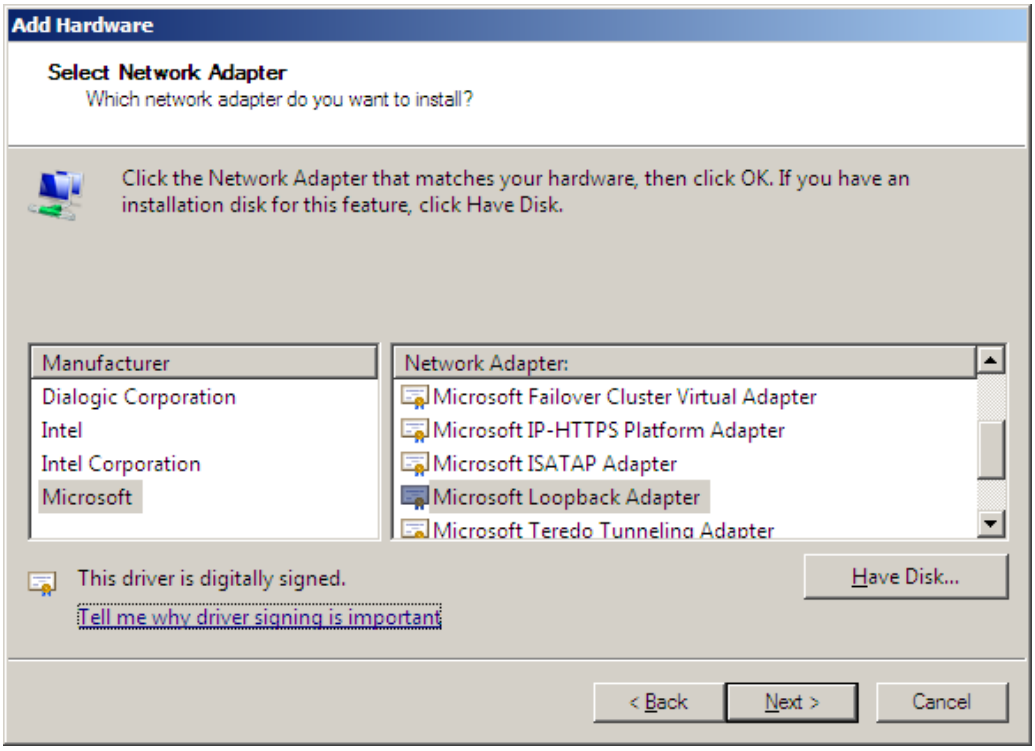


Solving for Windows 2008

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

Installing the Microsoft loopback adapter

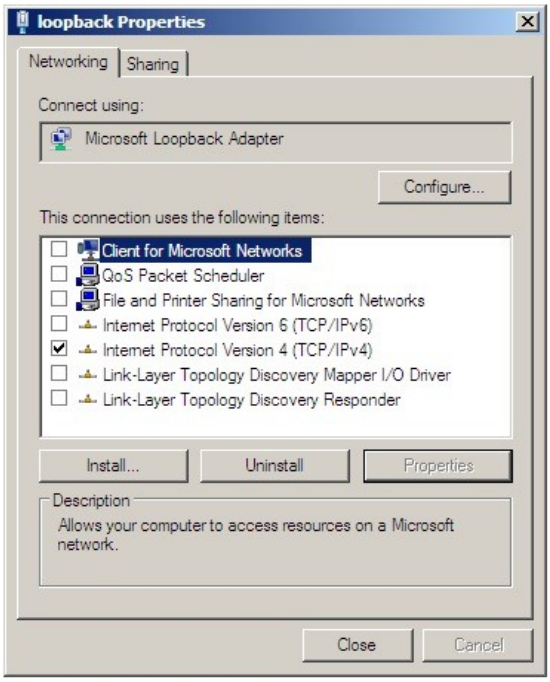
1. Click Start, select Run and enter ***hdwwiz*** to start the Hardware Installation Wizard
2. When the Wizard has started, click Next
3. Select 'Install the hardware that I manually select from a list (Advanced)', click Next
4. Select 'Network adapters', click Next
5. Select 'Microsoft' & 'Microsoft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



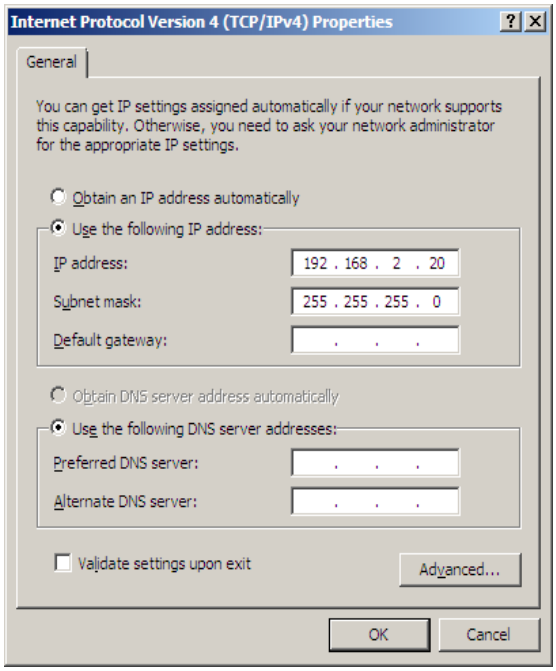
6. Click Next to start the installation, when complete click Finish

Configuring the loopback adapter

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



4. Un-check all items except Internet Protocol Version 4 (TCP/IPv4)
5. Select Internet Protocol Version (TCP/IPv4), click Properties and configure the IP address to be the same as the Virtual Server (VIP), e.g. 192.168.2.20/24



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

7. Now repeat the above process for all other Windows 2008 real servers

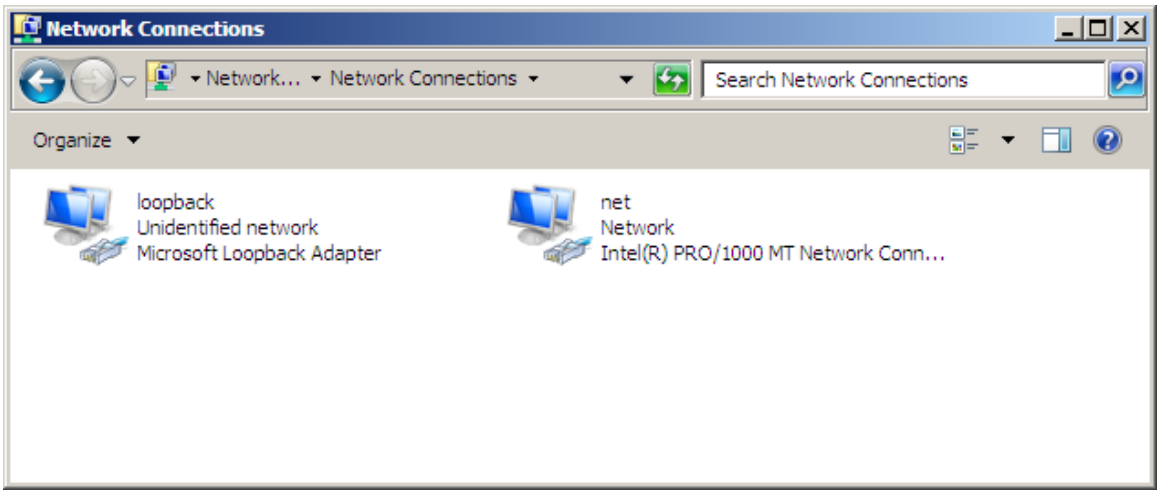
Configuring strong / weak host behavior

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

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

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

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



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

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

```
netsh interface ipv4 show interface
```

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

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

Windows 2008 R2 Firewall Settings

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

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

Windows 2008 R1 Firewall Settings

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

Advanced NAT considerations

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

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

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

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

BUT :

1. Your real servers won't be able to access the Internet through the new default gateway (except when replying to requests made through the external VIP)
2. External (non-load balanced) services such as FTP or SMTP will not be accessible because you haven't exposed any public IP addresses.

To solve problem #1

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

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

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

NB. From v6.7 you can specify which outgoing interface is to be masqueraded in global options.

To solve problem #2

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

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

Or

b) Set up individual public IPs for the services required with individual SNATs and DNATs for each service required i.e.

```
# SNAT & DNAT all traffic from EXT_MAIL to INT_MAIL
INT_MAIL="192.168.1.13"
EXT_MAIL="234.23.45.236"
# MAIL
iptables -t nat -A POSTROUTING -o $EXT_IFACE -p tcp -s $INT_MAIL -j SNAT --to-source $EXT_MAIL
iptables -t nat -A PREROUTING -i $EXT_IFACE -p tcp -d $EXT_MAIL -j DNAT --to-destination $INT_MAIL
```

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

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

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

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

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

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

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

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

Explaining the RIP & VIP in NAT mode

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

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

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

Protocol	VIP	Port	RIP	Port
TCP	10.0.0.20	80	192.168.2.50	80

All traffic destined for IP address 10.0.0.20 Port 80 is load-balanced over real IP address 192.168.1.50 Port 80.

Packet rewriting works as follows.

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

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

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

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

Replies get back to the load balancer as:

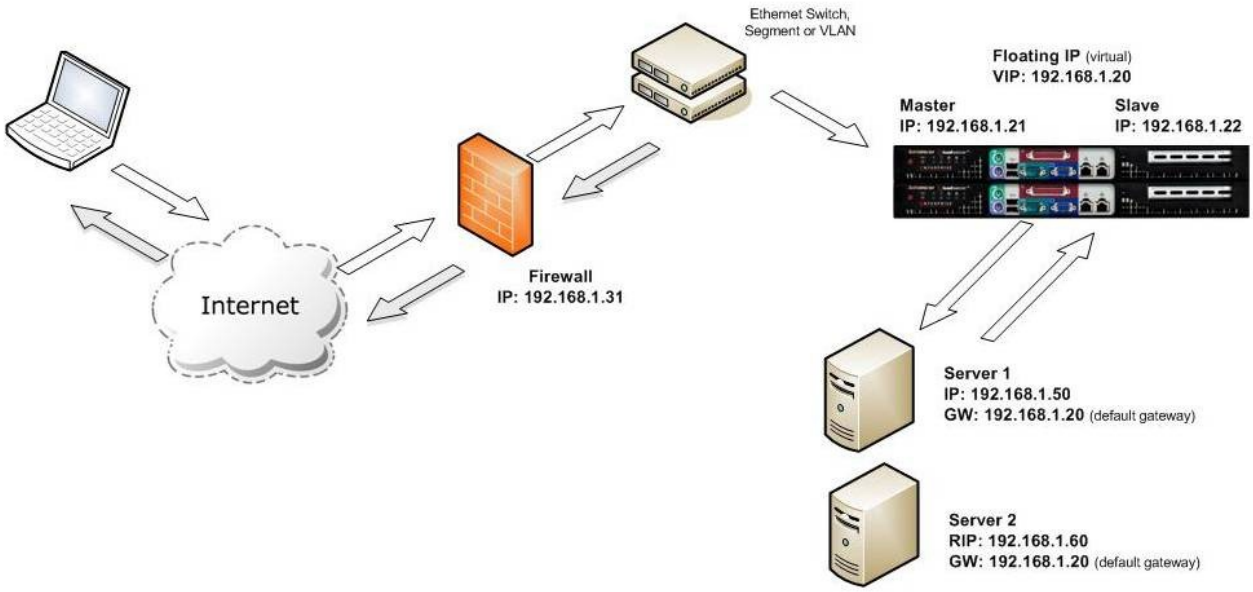
SOURCE	192.168.1.50:80	DEST	x.x.x.x:3456

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

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

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

Network Diagram: one arm – NAT Network Address Translation (clustered pair)



Notes:

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

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

Route configuration for Windows Server with one arm NAT mode

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

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

This is a simple case of adding a permanent route:

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

NB. Replace 192.168.1.0 with your local subnet address.

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

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

Route configuration for Linux with one arm NAT mode

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

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

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

NB. Replace 192.168.1.0 with your local subnet address.

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

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

NB. Replace 192.168.1.21 with your load balancer gateway

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

Advanced Layer 7 considerations

Load balancing based on URL match with HAProxy

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

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

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

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

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

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

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

In this example we have 'test_acl1' <-- just a label, 'path_beg' <--- i.e. match path beginning with... 'test1'. And similarly for test_acl2. There are numerous matching options available. For more details refer to: http://haproxy.1wt.eu/download/1.4/doc/configuration.txt (do a search for 'path_beg' on that page).

Handling Manual Changes to the HAProxy configuration file

Since Haproxy supports a very wide range of configuration options, under certain circumstances it may be required to manually add options to the /etc/haproxy/haproxy.cfg. The problem here is that when servers are later taken offline using System Overview, the file is rewritten according to the configuration in the WUI, which does not include the manual changes, and these are therefore overwritten. A way around this is to use the following commands via the console or a terminal window to control Haproxy:

To take a server offline:

echo "disable server VIP_Name/rip1" | socat unix-connect:/var/run/haproxy.stat stdio

To bring a server online:

echo "enable server VIP_Name/rip1" | socat unix-connect:/var/run/haproxy.stat stdio

HAProxy error codes

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

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

Complete detailed information for HAProxy configuration is available here:

http://haproxy.1wt.eu/download/1.4/doc/configuration.txt

SSL Certificates & Pound

If required, SSL can be offloaded to the load balancer. Pound is used to terminate SSL sessions and requires that the SSL certificate be deployed directly on the appliance. Http traffic will then be passed unencrypted to the real servers.

SSL termination concepts

Layer 7

At Layer 7 HAProxy must be configured to listen on port 80 of the virtual server, and then forward requests to port 80 of the real server. Pound is then configured to listen on port 443 of the virtual server, and forward the decrypted requests on to HAProxy to load balance.

Layer 4

For layer 4, Pound must be configured in a similar way, but instead of forwarding requests to HAProxy, requests are forwarded to a Layer 4 virtual server configured to operate in NAT mode.

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

For more details on SSL & Pound configuration steps, please refer to the relevant sections in Section G.

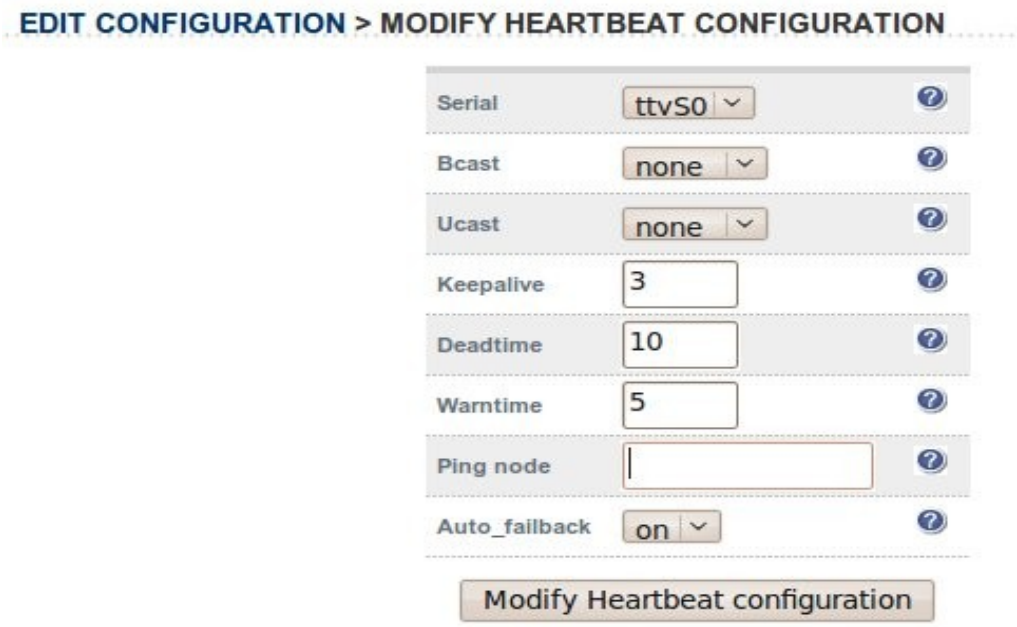
Health monitoring

The loadbalancer.org appliance supports both real server and load balancer health checks.

Load balancer health

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

Heartbeat Configuration



Serial Cable

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

The VMware appliance defaults to using the network (ucast) for its heartbeat.

Unicast (ucast)

When you enable the Unicast (network) based heartbeat you must specify the interface to use (especially important when you activate bonding on interfaces). Ucast is preferred for heartbeat if Serial can not be used for any reason.

Broadcast (bcast)

Enable a Broadcast based heartbeat and choose the interface. When you enable the Broadcast (network) based heartbeat you must specify the interface to use (especially important when you activate bonding on interfaces). You can not have several load pairs of balancers within the same network having broadcast enabled for heartbeat. With the introduction of Ucast we recommend to use Ucast.

Ping Node

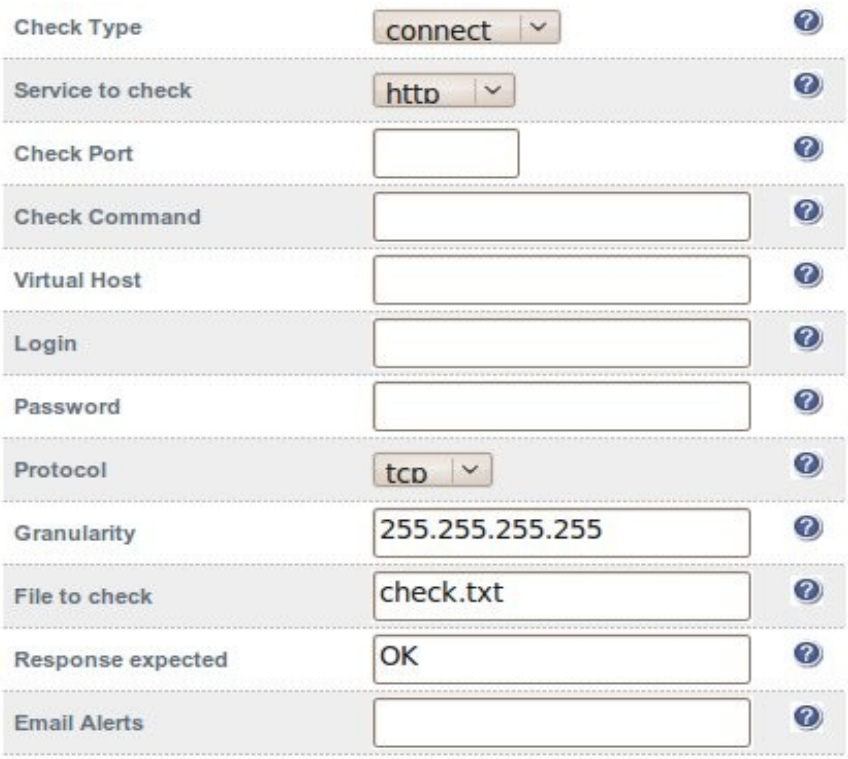
Specify a mutually accessible IP address to test network availability. A good ping node to specify is the IP address of a router that both the master and slave can access (e.g. the default gateway). If the active appliance looses access to the ping node, but the passive appliance still has access, then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

Real server health

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

Configuration – Layer 4

Edit Configuration > Virtual Servers > Modify



Check Types

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

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

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

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

Off - All real servers are off

On - All real servers are on (no checking)

5 - Do 5 connect checks and then 1 negotiate

10 - Do 10 connect checks and then 1 negotiate

Service to check

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

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

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

ftp – use ftp as the negotiate protocol (optional username/password, filename in the default folder)

imap – use imap as the negotiate protocol (requires username/password)

pop – use pop as the negotiate protocol (requires username/password)

ldap – use ldap as the negotiate protocol

smtp – use smtp as the negotiate protocol

nntp – use nntp as the negotiate protocol

dns – use dns as the negotiate protocol

mysql – use mysql as the negotiate protocol

sip – use sip as the negotiate protocol

telnet – use telnet as the negotiate protocol

none

Check Port

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

Check Command

Location of the custom check file relative to root. For use with check type = external.

Virtual Host

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

Login

The login name to use for IMAP,POP3 or FTP accounts (used for negotiate check)

Password

The password to use for negotiate checks.

File to Check

Specify the name of the file to check if you are using 'negotiate' for the check type.

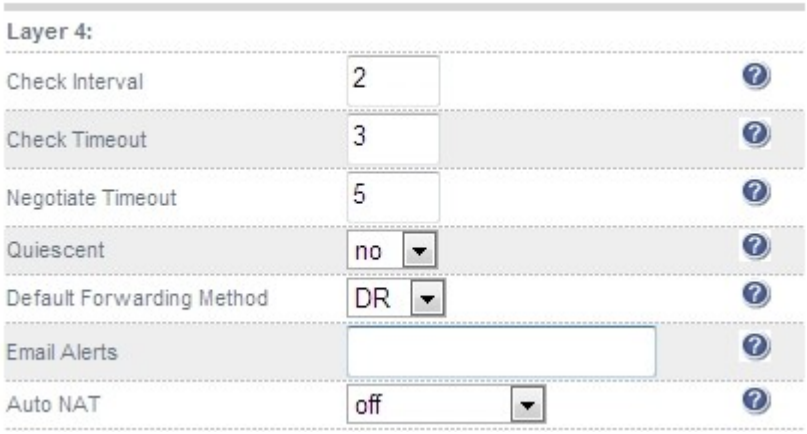
Response Expected

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

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

Additional health check settings

Edit Configuration > Global Settings



Check Interval

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

Check Timeout

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

Negotiate Timeout

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

Quiescent

Yes - When a real server is determined to be down, its not actually removed from the kernel's LVS table. Instead, the weight is set to zero which means that no new connections will be accepted.

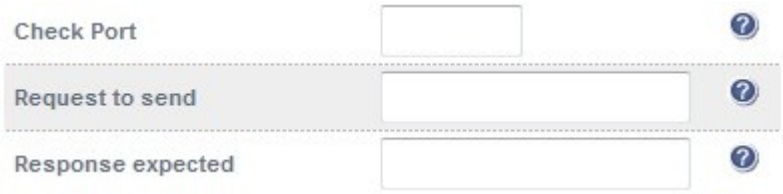
No – When a real server is determined to be down, the real server will be removed from the kernel's LVS table.

Email Alerts

The is the default global setting for email alerts and is used if not specified at the individual VIP level.

Configuration – Layer 7

Edit Configuration > Virtual Servers (HA Proxy) > Modify



Check Port

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

Request to send

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

Response expected

The content expected for a valid health check on the specified file. The response expected can be any valid regular expression statement.

Advanced firewall considerations

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

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

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

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

Firewall marks

You can use the modify firewall script option to group certain protocols together in one cluster. So, rather than specifying VIP as *IPAddress:Port*, you can specify it as '1' i.e. Firewall mark 1.

If you first create a standard VIP (i.e. IPAddress:Port), then change it to a firewall mark (e.g. change IPAddress:Port to '1'), the required floating IP will already exist. If you create the firewall mark directly, you MUST also manually create a corresponding floating IP. This can be done in the WUI under *Edit Configuration > Floating IP(s).*

For Firewall Marks the protocol in the Virtual Server must be set to “fwm”. if a VIP is created first, then modified to a Firewall Mark, this will be handled automatically, if the Firewall Mark is created manually then the protocol must be set manually.

E.g. to configure firewall mark '1', a standard Virtual Server as shown below:

would be changed to:

Then, any incoming packets marked with a '1' will be associated with that VIP. This is especially useful if you need persistence as clients move from HTTP to HTTPS , e.g. an e-commerce web site without a proper back end database for session state.

NB. The firewall script also needs to be changed – see example below

```
# This example marks HTTP & HTTPS connections only
VIP1="192.168.2.165"
 iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1
 iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 443 -j MARK --set-mark 1
```

NB. The default firewall script can be un-commented and modified as required. In the WUI, goto Maintenance > Firewall Script.

Firewall marks are also useful for setting up a very large number of VIPs in a test environment.

NB. When you add the real servers to a firewall mark based virtual server it doesn't matter which port you put. All firewall mark based services are passed through to the same port they tried to connect to. So if the customer requests port 80 they will be sent to port 80 on the real server.

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

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

--dport 1024:1090

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

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

FTP

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

active 20,21
passive 21,high_port

Most firewalls handle this insecure protocol by stateful inspection of the traffic in order to open up the required data port on demand. LVS has a built in helper module (that loads on demand) in order to handle the correct port translation when in MASQ/NAT mode. Therefore, if you set up a Virtual Server on port 21 in MASQ/NAT configuration it should work without a hitch.

However, in DR mode the load balancer cannot see the return packets. One of the simplest ways of dealing with this is to allow your real server to have outgoing FTP access for return traffic to the client from it's RIP and configure only the incoming traffic on the load balancer. So set up a VIP on port 21 for the incoming traffic and allow the server to do the rest of the communication directly with the client. *NB. Your firewall will need to allow FTP connections to all the RIPs as well as the VIP.*

The second direct routing method is to effectively open up all ports and group them together to allow the connections to always talk to the same server. This is best done with a Firewall Mark:

```
# This example marks groups the active FTP ports
VIP1="192.168.0.66"
# First two rule are for Active connections
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 21 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 20 -j MARK --set-mark 1
# Third additional rule for passive
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 1024: -j MARK --set-mark 1
```

NB. Your firewall will either need the same rules or preferably stateful inspection for FTP access to the VIP.

DR mode requires the use of firewall marks for both passive and active FTP. NAT mode works in both FTP modes without any additional firewall marks.

NAT mode works in both FTP modes without any additional firewall marks. If an alternative port to the standard port 21 is required, then follow the steps in the section below.

Changing the FTP Port in NAT Mode

This procedure details how to configure the load balancer to respond to FTP on port 2180 rather than the standard port 21 (this only applies when using NAT mode).

Edit the modprobe.conf configuration file:

```
nano /etc/modprobe.conf
```

At the end of the file, on a new line, add the following:

options ip_vs_ftp ports=2180

now save the file (Ctrl-O, Enter) and exit (Ctrl-X).

To apply the changes to the running system, the ftp module will need to be reloaded. This will only need to be done to test the change - on future system reboots, the option will be applied automatically. On the command line, enter:

```
modprobe -r ip_vs_ftp
```

Check that the module has been unloaded successfully:

```
lsmod | grep ip_vs_ftp
```

there should be no output from the command above.

Then load the module again:

```
modprobe ip_vs_ftp
```

Repeat the check:

```
lsmod | grep ip_vs_ftp
```

This time, you should see output similar to the following:

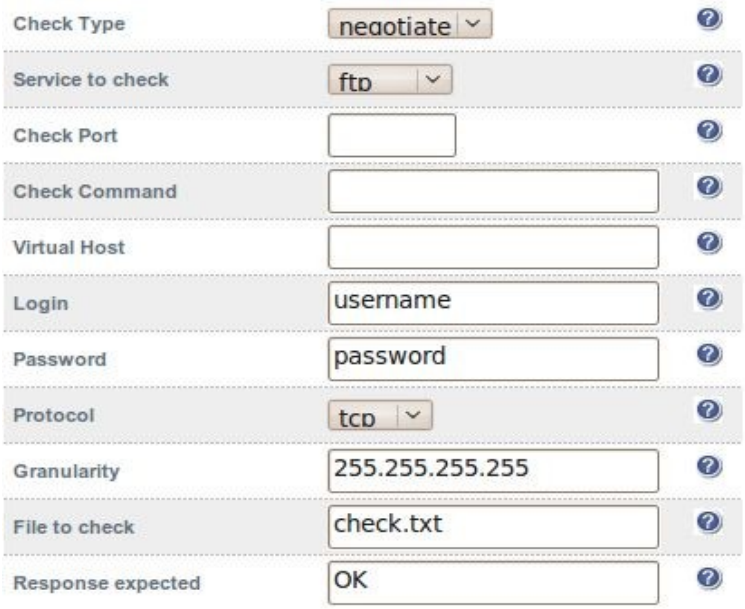
ip_vs_ftp 6297 0

ip_vs 105965 4 ip_vs_ftp

The system is now configured to respond to FTP on port 2180 instead of port 21.

FTP negotiate health check

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



- Change the c*heck type* to negotiate
- Make sure the *service to check* is FTP
- Specify a *login* and *password*
- Specify the *file to check* (defaults to the root directory)
- The file is parsed for the *Response expected* that you specify

FTP recommended persistence settings

When you start using multiple FTP servers in a cluster you need to be aware of the effects of a client switching server. For sites that are download only, you generally don't need any special settings on the load balancer as the connection will usually stay on the same server for the length of the connection. You may however wish to force persistence to something sensible like 15mins (If you go higher remember to change the global TCP timeouts).

If you are using the FTP servers for upload it is recommended to use a single FTP server for uploads and then replicate the data to the read only cluster for downloads (or use a clustered file system). For upload it is especially important to use persistence.

Automatically resuming a broken download is no problem even if you switch servers in a cluster on re-connect. This is because the FTP resume functionality is client based and does not need any server session information.

Limiting passive ports

To reduce the number of ports that the load balancer marks, the following command could be used instead of the command mentioned earlier:

```
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 –dport 50000:50100 -j MARK –set-mark 1
```

This would only allow ports 50000-50100.

This can then be limited in the same way on the real server's firewall and also on the FTP server so that passive connect ports proposed by the ftp server are in this range. The way to to limit the passive port ranges on a range of typical systems is shown below:

For Linux

in vsftpd, the following line can be added to the vsftpd.conf file to limit the port range:

pasv_max_port - max is 65535
pasv_min_port - min is 1024

in proftpd, the following line can be added to the proftpd.conf file to limit the port range:

PassivePorts 50000 - 50100

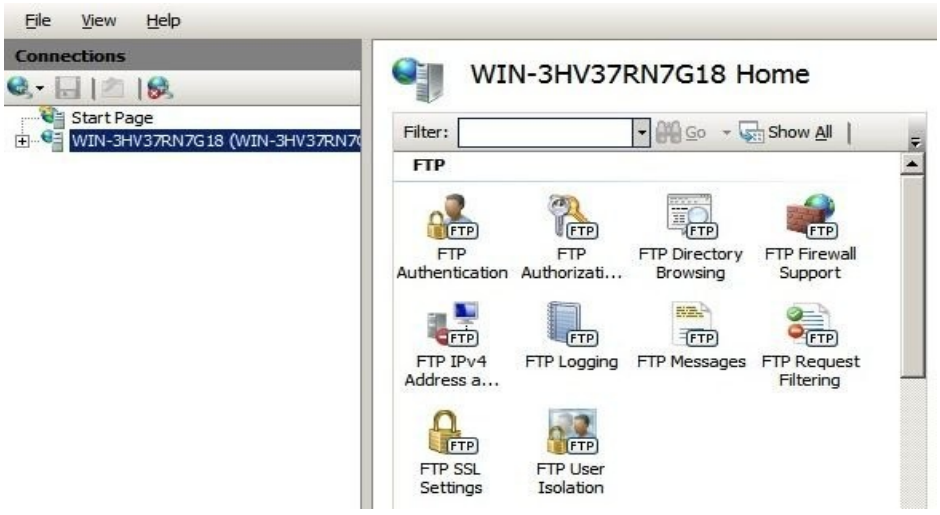
in pureftpd, the following startup switch can be used:

-p --passiveportrange <min port:max port>

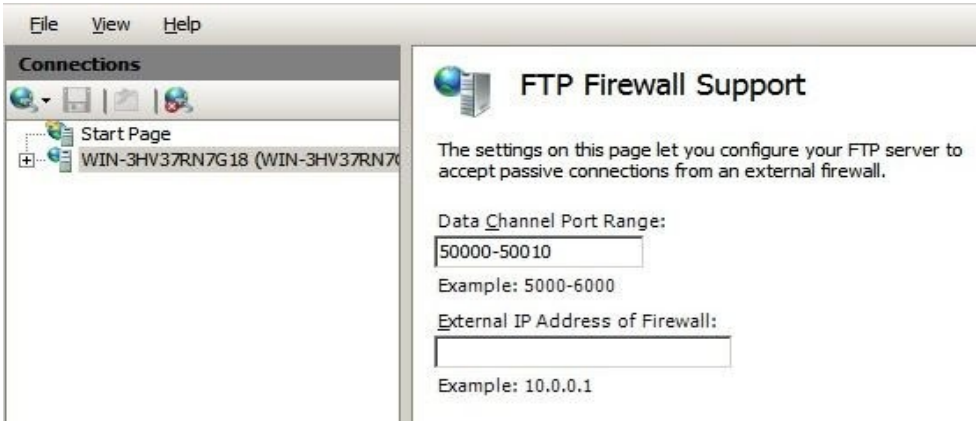
N.B. The firewall should also be configured to limit the ports to the same ranges.

For Windows 2008

Open the IIS Management console, highlight the server node, then double-click the FTP Firewall Support icon.



The following screen will be displayed:



Enter the required port range in the Data Channel Port Range field and apply the changes. These settings will apply to all FTP sites created on the server.

For Windows 2003

a) Enable Direct Metabase Edit
1. Open the IIS Management Console
2. Right-click on the Local Computer node
3. Select **Properties**
4. Make sure the **Enable Direct Metabase Edit** checkbox is checked

b) Configure PassivePortRange via ADSUTIL script
1. Click **Start**, click **Run**, type cmd, and then click **OK**
2. Type cd Inetpub\AdminScripts and then press ENTER
3. Type the following command from a command prompt
   adsutil.vbs set /MSFTPSVC/**PassivePortRange** "50000-50100"
4. Restart the FTP service

For Windows 2000

Configure PassivePortRange via Registry Editor
1. Start Registry Editor (Regedt32.exe)
2. Locate the following registry key:
   HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Msftpsvc\Parameters
3. Add a value named "PassivePortRange" (without the quotation marks) of type REG_SZ
4. Close Registry Editor
5. Restart the FTP service

(SP4 must be installed for this to work)

Note: The range that FTP will validate is from 5001 to 65535

Persistence considerations

Persistence > 15 minutes

Some services such as TELNET, SSH, FTP & Terminal Server (RDP) may require a persistence setting of greater than 15 minutes.

If you require a persistence of greater than 15 minutes then you will need to increase the load balancers TCP time out value. The TCP time out can be set as a command in your firewall script:

```
ipvsadm --set 3600 0 0
```

This example sets the TCP timeout to 1 hour, you should make sure this timeout is the same as your required persistence setting.

NOTE: From v6.7 this timeout setting is created automatically in /etc/rc.d/rc.lvstimeout , and therefore manual changes are not normally required

Server maintenance when using persistence

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

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

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

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

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

Persistence state table replication

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

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

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

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

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

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

```
ipvsadm -Lnc
```

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

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

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

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

Terminal Server RDP considerations

RDP – Layer 4

RDP is a simple TCP based service usually on port 3389. Because of the nature of a Terminal Server you will always want the clients to connect to the same server so that you maintain the session. The common setting to use with Terminal Server is *persistence=900* (15 minutes). This means that if a client is idle for more than 15 minutes then the load balancer will treat the next connection as a new connection and possibly take them to a different server.

By default a Terminal Server connection that is not minimized will perform a keepalive ping every 60 seconds and therefore the client will stay persistent indefinitely. You would normally make your Terminal Server policy to reap idle clients at 15 minutes matching the load balancers persistence setting.

Layer 7 (RDP Cookies)

In some instances Layer 4 source IP persistence can result in uneven load balancing. This would normally happen if you have a large number of users coming through a corporate firewall or proxy. If a large number of users have the same source IP address they will all hit the same back end server.

If you have this issue the Load balancers have a special layer 7 RDP persistence mode that keeps track of the users login and matches it to a back end server. Using this method even if a user connects from a completely different computer they will still hit the same back end server and session.

Simply configure a layer 7 HAProxy Virtual Server on port 3389 and select the persistence mode of RDP cookie. Then add you back end real servers as required with the same port 3389. The default persistence time out is 1 hour which should be ideal for most implementations.

If your back-end servers are running Windows 2008 (R1) / Terminal Services, make sure that the Security Layer setting of the RDP connection properties are set to *RDP Security Layer*, otherwise the RDP Cookie may be encrypted and will not be readable causing persistence to break.

NIC bonding and high-availability

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

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

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

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

Example 1: Bonding for bandwidth

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

Are you really doing 1Gb/s+?

Example 2: Bonding for high-availability (recommended)

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

This works with any switch.

Example 3: Bonding for high-availability & bandwidth

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

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

SNMP reporting

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

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

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

The dedicated load balancing mib oid is: 1.3.6.1.4.1.8225.4711

SNMP for layer 4 based services

You can test if everything works by invoking:

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

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

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

SNMP for layer 7 based services

Front end stats are returned by invoking:

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

Back end stats are returned by invoking:

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

Feedback agents

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

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

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

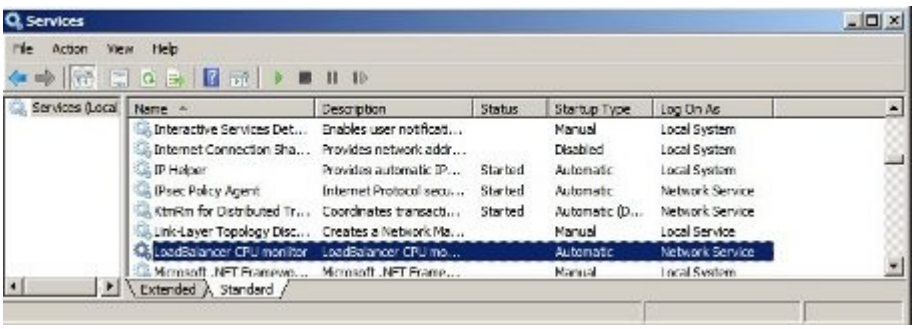
Installing the Windows agent

Download the agent from:

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

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

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



Installing the Linux/Unix agent

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

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

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

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

/etc/init.d/xinetd restart

Testing:
telnet 127.0.0.1 3333

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

Custom HTTP agent

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

Changing the local date, time & time zone

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

Logs > Change the date/time settings

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

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

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

```
hwclock --systohc
```

NTP configuration

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

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

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

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

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

Restoring Manufacturer's settings

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

From the console

```
lbrestore
```

From the WUI

Use *Maintenance > Disaster Recovery > Restore Manufacturers Settings*

Force master/slave take-over in a clustered pair

Force the Master to become passive

On the master:

```
/usr/lib64/heartbeat/hb_standby
```

or

On the slave:

```
/usr/lib64/heartbeat/hb_takeover
```

Force the Master to become active

On the master:

```
/usr/lib64/heartbeat/hb_takeover
```

or

On the slave:

```
/usr/lib64/heartbeat/hb_standby
```

Active / Active load balancer configuration

Normally load balancer clustered pairs are deployed in an active / passive configuration. In this mode only one device hosts the Virtual Services (VIPs) at any one time. However, it is possible to configure the cluster so that the VIPs are shared between both devices at the same time (i.e. an active / active configuration). To achieve this, the heartbeat configuration file /etc/ha.d/haresources must be edited on the master, then copied over to the slave.

An example of a typical active / passive configuration file:

lbmaster 192.168.36.34 192.168.36.35 192.168.36.36 192.168.36.37 ldirectord haproxy pound

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

lbmaster 192.168.36.34 192.168.36.35 ldirectord haproxy pound

lbslave 192.168.36.36 192.168.36.37 ldirectord haproxy pound

In this file, the format is :

<preferred device> <ip address> <resources>

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

If one of the nodes becomes inactive, its VIPs are transferred to the active node, until such time as the inactive node becomes active again.

The configuration file would also need to be copied over to the slave by running the following command on the master:

```
scp /etc/ha.d/haresources root@lbslave:/etc/ha.d/
```

A configuration change is also needed to make sure that Layer 7 services are able to start up when they are not able to bind to a local VIP. This is done by executing the following command (on both nodes):

```
echo 1 >/proc/sys/net/ipv4/ip_nonlocal_bind
```

If you wish to make this change permanent then the following line should be added to /etc/sysctl.conf

Allow binding to non local addresses

net.ipv4.ip_nonlocal_bind = 1

Section F – Disaster Recovery

Being prepared

To be able to quickly recover your appliance when a disaster occurs it is important that you create a backup of the XML configuration file and keep it stored in a safe location off the load balancer. Ideally you should keep a backup of both the master and slave configurations. This can easily be done by following the steps below:

Backing up to a remote location

Login to the web interface:

Username: loadbalancer
Password: loadbalancer

Go to *Maintenance* > *Disaster Recovery*

- Select *Download XML configuration file*
- Select *save* and enter a secure location

Also for the firewall configuration:

NB. If you have not made changes to the firewall script (Maintenance > Firewall Script), this is not necessary.

- Select *Download Firewall Script*
- Select *save* and enter a secure location

Backing up to the load balancer

To create a backup that is stored on the load balancer itself, follow these steps:

Log in to the web interface:

Username: loadbalancer
Password: loadbalancer

Go to *Maintenance*

- Select *Configuration Backup*
- A copy of lb_config.xml will be stored in /etc/loadbalancer.org/userbkup

Appliance recovery using a USB memory stick

The following instructions detail how to recover a Loadbalancer.org appliance to the latest version using a USB stick (1GB or more in capacity).

This will only work on 64Bit hardware. From version 6.0 onwards, all appliances are 64Bit. If you are running an older version, this may or may not be possible depending on the hardware.

If you are running v5 and wish to determine whether your appliance is 64Bit, then enter the following command:

```
grep flags /proc/cpuinfo
```

If **lm** (long mode) is present in the output then the CPU is 64Bit and you can proceed. If not then your appliance is 32Bit and you are limited to the latest v5 software.

NB. Loadbalancer.org are continuing to support v5.

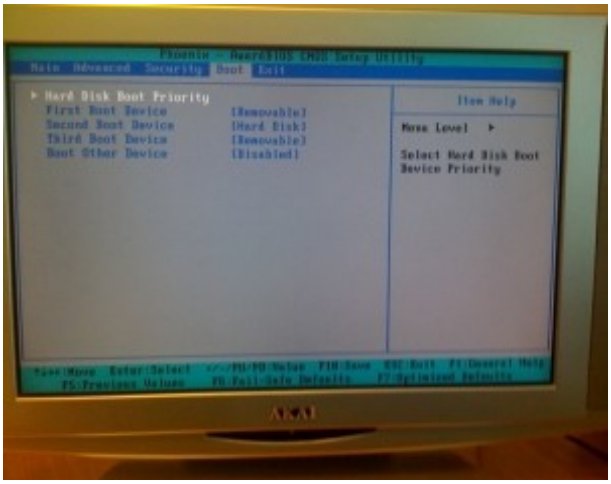
The latest images require a high speed 4Gb IDE DOM / Flash. If you are already running v6 then you will already have this and should be able to simply re-image your current IDE DOM / Flash. If you are upgrading from v5 you will need to purchase a 4Gb IDE DOM / flash card and then use the following procedure to build it from the USB stick.

If you are already running v6 then you can keep your current configuration by backing up the XML file on BOTH the master and the slave. This can then be uploaded to the new appliance once the following steps have been completed.

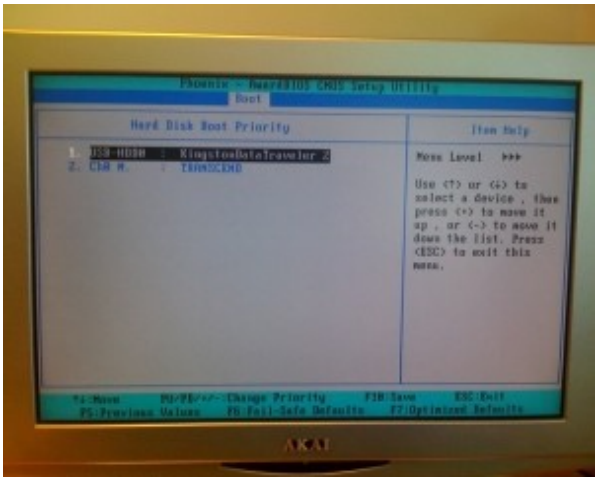
The latest version of the ISO file is available at http://www.loadbalancer.org/download/

You can use UNetBootIn (Windows or Linux) to transfer the ISO onto the USB stick.

Make sure you change the server BIOS to boot from the USB first (stick must be plugged in at that stage).



First boot device (Removable)



Hard disk boot priority (USB)

When it boots choose:

Default image

Once the **root@lbmaster:/ #** prompt appears, enter the following commands:

```
# cd /etc/recovery
# ./clone-dsk.sh
```

at the first prompt, press <ENTER>
at the second prompt, select option 1
at the third prompt, select option 1
at “Is the disk /dev/hda already formatted (manually) [Y/N]?” type 'N'
at “do you want to reformat /dev/hda [Y/N]?” type 'Y'
then Yes to all other prompts

The image transfers onto any IDE HD or IDE DOM / Flash module. This takes around 5 mins to complete.

Once complete, remove the USB stick and reboot the appliance

******* You now have a fully functioning appliance *******

Now continue with the relevant slave / master recovery steps.

Disaster recovery after master failure

For a correctly configured clustered pair, if the master fails, the slave will take over. To restore the master load balancer's configuration, a backup copy of the lb_config.xml file is used. This backup should be created using the steps on page 98.

NB: If a backup copy from the master is not available, It's possible to use the lb_config.xml from the slave instead. If there is no current backup of this, then take a copy of the lb_config.xml from /etc/loadbalancer.org on the slave. A couple of changes need to be done so that the file represents the master unit rather than the slave as shown below.

Steps to modify a copy of the configuration file from the slave, for use on the master:

Change:

```
<network>
    <hostname>lbslave</hostname>
    <slave></slave>
    <fullsync></fullsync>
```

To:

```
<network>
    <hostname>lbmaster</hostname> (change to 'lbmaster')
    <slave>192.168.2.165</slave> (specify the IP of your slave unit)
    <fullsync>on</fullsync> (change to 'on')
```

Change:

```
<eth0>
    <ip>192.168.2.165</ip>
    <netmask>255.255.255.0</netmask>
```

To:

```
<eth0>
    <ip>192.168.2.164</ip> (change to the IP of your master unit)
    <netmask>255.255.255.0</netmask>
```

NB. If you use eth1, eth2 (Ent. MAX only) or eth3 (Ent. MAX only) these should be changed in the same way

To perform the recovery:

- *Locate your copy of lb_config.xml (either the backup from the master, or the modified slave copy)*
- If the failed master is still on, power it down
- Disconnect the Heartbeat (serial) cable and the network cable
- Repair the problems you are having with the master
- Connect a mouse, monitor and keyboard & power up the master
- Restore the master from the Load balancer ISO image using a USB stick by following the steps on page 99-100

  N.B. During the aforementioned restore at no point reconnect the cables!
- Log onto the console of the master appliance as:

  Username: root
  Password: loadbalancer
- At the terminal stop the heartbeat service with the following command:

  # service heartbeat stop
- On a workstation, open http://www.loadbalancer.org/download/recoveryscripts/ , right-click lbrecover-cli, select Save Target As.. / Save Link As and save the file (with no extension)
- Copy your backup lb_config.xml file and the lbrecover-cli file onto the USB stick
- Insert the USB stick into the master unit
- To determine the system reference for the USB stick type the following command:

  # fdisk -l *(lowercase 'L')*
- Mount the USB device (normally named sda1):

  # mount /dev/sda1 /mnt
- Run the following command:

  # php /mnt/lbrecover-cli /mnt/lb_config.xml

  NB. Ignore any errors relating to connecting to the slave, this is expected since the master is not currently connected to the network
- Once the script has completed and the # prompt has reappeared, shutdown the master unit using the following command:

  # shutdown -h now
- Now disconnect the USB stick and reconnect the Heartbeat (serial) cable and the network cable
- Power-on the master unit
- Now logon to the WUI of the master:

  Username: loadbalancer
  Password: loadbalancer

- After a minute or so restart heartbeat by going to *Maintenance* > *Restart Heartbeat*

NB: Its important to remember that when heartbeat is restarted, virtual services will be temporarily interrupted and all existing client sessions will be closed. Therefore at least the previous step should be completed during a maintenance window.

- Now check the heartbeat settings using *Edit Configuration* > *Heartbeat Configuration.* If Auto Fail-back is set to 'Off' you'll need to manually force the master to become active using the following command on the master:

```
# /usr/lib64/heartbeat/hb_takeover
```

After a minute or so your cluster should be restored with the master unit as the active appliance and the slave as the passive appliance.

To confirm this:

Layer 4 Services

Go to *Reports* > *Current Connections* , if you do this on both the master and the slave you should see all connections being routed through the master and none to the slave.

Also, go to *View Configuration* > *Network Configuration* on the master and verify that the Virtual Server's floating IP's are active (you should see eth0:0, eth0:1 etc.). If visible, then the master is the active unit.

Layer 7 Services

At layer 7, go to *View Configuration* > *System Overview,* if the layer 7 Virtual Servers are listed at the bottom of the screen then the unit is active, if not, then its passive.

Disaster recovery after slave failure

If the slave unit has failed, the master will continue to provide load balancing functions as normal. However it is important to recover the slave unit as soon as possible to restore the clustered pair to normal.

To restore the load balancer's configuration, a backup copy of the lb_config.xml file is used. This backup should be created using the steps on page 98.

NB: If a backup copy from the slave is not available, It's possible to use the lb_config.xml from the master instead. If there is no current backup of this file, then take a copy of the lb_config.xml from /etc/loadbalancer.org on the master. A couple of changes need to be done so that the file represents the slave unit rather than the master as shown below.

Steps to modify a copy of the configuration file from the master, for use on the slave:

Change:

```
<network>
    <hostname>lbmaster</hostname>
    <slave>192.168.2.165</slave>
    <fullsync>on</fullsync>
```

To:

```
<network>
    <hostname>lbslave</hostname> (change to 'lbslave')
    <slave></slave> (remove the IP of the slave)
    <fullsync></fullsync> (remove fullsync 'on')
```

Change:

```
<eth0>
    <ip>192.168.2.164</ip>
    <netmask>255.255.255.0</netmask>
```

To:

```
<eth0>
    <ip>192.168.2.165</ip> (change to the IP address of the slave)
    <netmask>255.255.255.0</netmask>
```

NB. If you use eth1, eth2 (Ent. MAX only) or eth3 (Ent. MAX only) these should be changed in the same way

To perform the recovery:

- *Locate your copy of lb_config.xml (either the backup from the slave, or the modified master copy)*
- If the slave is still on, power it down
- Disconnect the Heartbeat (serial) cable and the network cable
- Repair the problems you are having with the slave
- Connect a mouse, monitor and keyboard & power up the slave
- Restore the slave from the Load balancer ISO image using a USB stick by following the steps on page 99-100

  N.B. During the aforementioned restore at no point reconnect the cables!

- Log onto the console of slave appliance as:

  Username: root
  Password: loadbalancer

- At the terminal stop the heartbeat service with the following command:

  # service heartbeat stop

- On a workstation, open http://www.loadbalancer.org/download/recoveryscripts/ , right-click lbrecover-cli, select Save Target As.. / Save Link As and save the file (with no extension)
- Copy your backup lb_config.xml file and the lbrecover-cli file onto the USB stick
- Insert the USB stick into the slave unit
- To determine the system reference for the USB stick type the following command:

  # fdisk -l *(lowercase 'L')*

- Mount the USB device (normally named sda1):

  # mount /dev/sda1 /mnt

- Run the following command:

  # php /mnt/lbrecover-cli /mnt/lb_config.xml

  NB. Ignore any errors relating to unknown interfaces / devices.

- Once the script has completed and the # prompt has reappeared, shutdown the slave unit using the following command:

  # shutdown -h now

- Now disconnect the USB stick and reconnect the Heartbeat (serial) cable and the network cable
- Power-on the slave unit
- Now logon to the WUI of the master:

  Username: loadbalancer
  Password: loadbalancer

- After a minute or so, restart heartbeat by going to *Maintenance* > *Restart Heartbeat*

NB: Its important to remember that when heartbeat is restarted, Virtual services will be temporarily interrupted and all existing client sessions will be closed. Therefore at least the previous step should be completed during a maintenance window.

After a minute or so, your cluster should be restored with the master unit as the active appliance and the slave as the passive appliance.

To confirm this:

Layer 4 Services

Go to *Reports* > *Current Connections* , if you do this on both the master and the slave you should see all connections being routed through the master and none to the slave.

Also, go to *View Configuration* > *Network Configuration* on the master and verify that the Virtual Server's floating IP's are active (you should see eth0:0, eth0:1 etc.). If visible, then the master is the active unit.

Layer 7 Services

At layer 7, go to *View Configuration* > *System Overview,* if the layer 7 Virtual Servers are listed at the bottom of the screen then the unit is active, if not, then its passive.
.

Section G – Web User Interface Reference

View Configuration

System Overview

View an overview of system performance and cluster status.

XML

View the lb_config.xml configuration file. This details the main configuration for the appliance.

Layer 4

View the layer 4 configuration file.

Layer 7 (HAProxy)

View the haproxy.cfg configuration file.

SSL Termination (Pound)

View the pound.cfg configuration file.

Network Configuration

View the running configuration of the network of the load balancer.

Heartbeat Configuration

View the ha.cfg configuration file.

Heartbeat Resources

Displays the contents of the /etc/ha.d/conf/haresources file.

Routing Table

View the routing table of the appliance.

Firewall Rules

View all firewall rules configured on the appliance.

Edit Configuration

Set up or modify the physical and virtual configuration of the load balancer appliance.

Logical Layer 4 Configuration

The Logical Layer 4 Configuration controls how the incoming traffic is handled for Virtual Servers and Real Servers.

Virtual Servers

This menu option allows you to add, remove or modify virtual servers from your cluster.

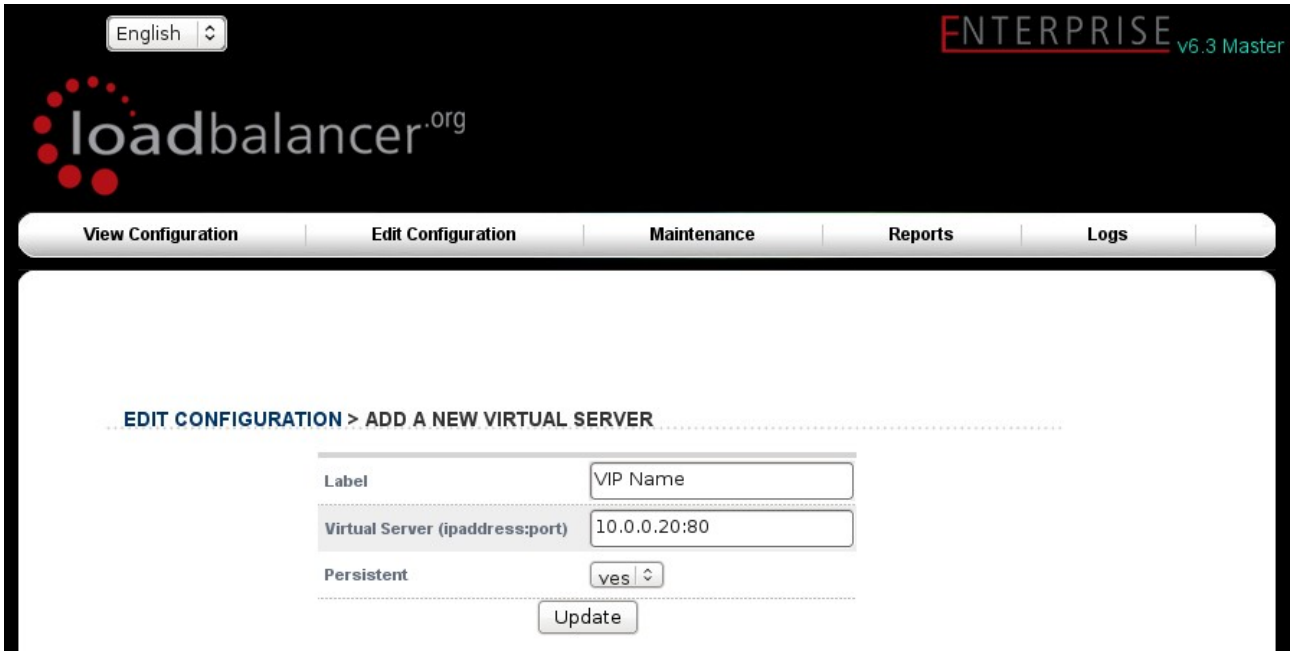
Each Virtual Server has a number of real servers, for example one Virtual Server can have any number of Real Servers in its cluster.

You need one Virtual Server for each distinct cluster *AND* protocol that you wish to load balance.
So if you want to serve both HTTP and HTTPS then you will need two virtual servers:

192.168.1.20:80 & 192.168.1.20:443

NB. *Assuming that 192.168.1.20 is the Floating Virtual IP address shared between the master and slave load balancer.*

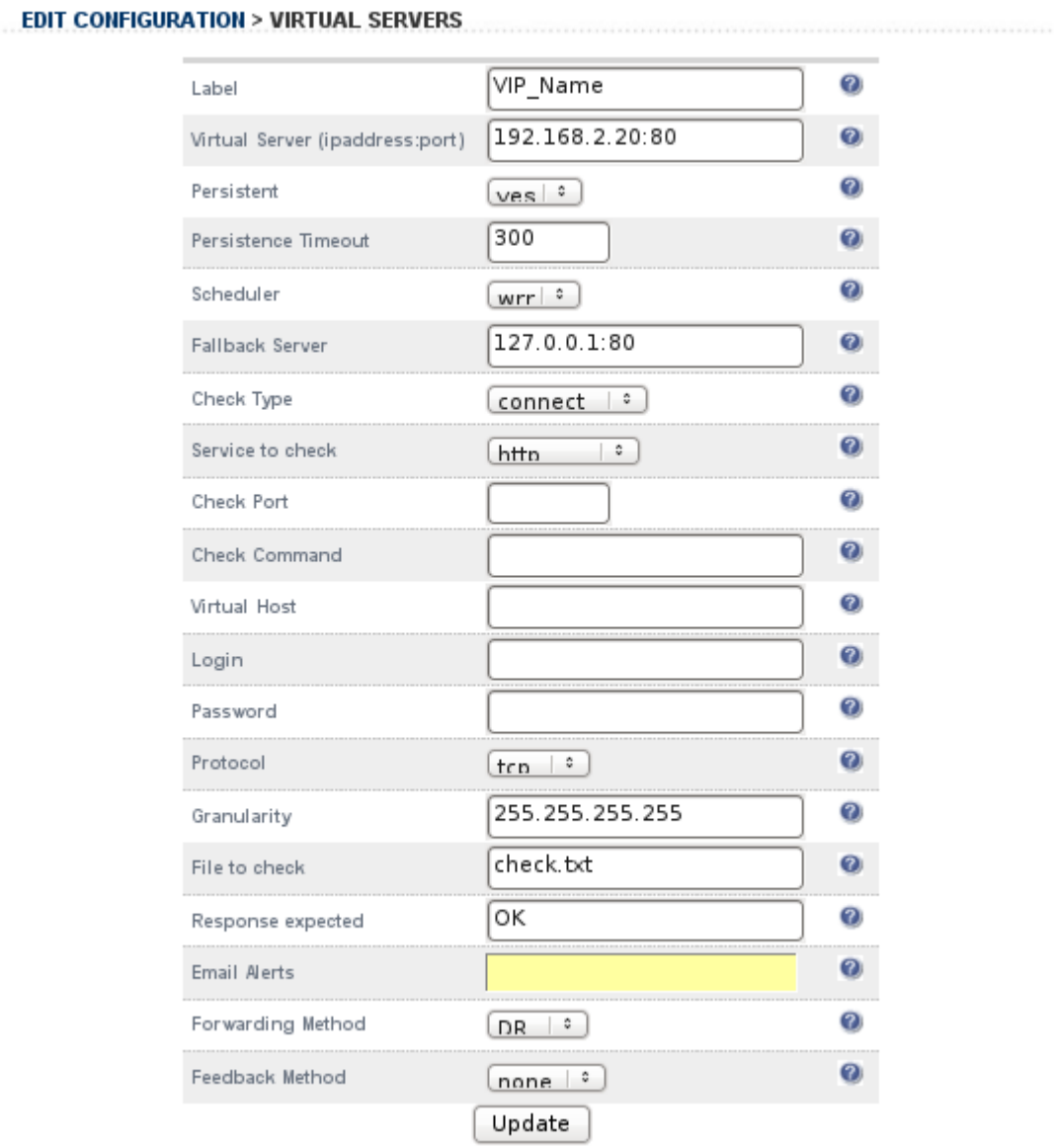
Adding a Virtual Server is a simple case of specifying the IP address & port number. If you require the client connections to stick to the first real server they hit then say *'yes'* to sticky connections. This is recommended for HTTPS to stop clients repeatedly re-negotiating SSL keys.



Persistence is based on source IP address & destination port. The time out is in seconds and each time the client makes a connection the timer is reset so even a 10 minute persistence setting could last for hours if the client is active.

- The load balancer will automatically add the Virtual Server to the pool of Floating IP(s) if required.
- The Floating IP should activate instantly.
- Just check '*View Configuration* > *Network Configuration*' to ensure that the Floating IP address has been activated correctly. It will show up as an alias i.e. eth0:0 etc.

Modify Virtual Server has several more options that have been filled in by default when you added the virtual server.



Here you can modify :

- The virtual IP address and port (or firewall mark).
  - It is important that the virtual IP be on the same subnet as one of the physical IPs under Edit Configuration > Network Interface Configuration.
- Whether you want persistent/sticky connections (source IP persistence)
- How long should the connections persist in seconds (300 should be fine)
- What type of scheduler to use :
  - WLC – Weighted Least Connection (Often used for more even balancing)
  - RR – Round Robin
  - WRR – Weighted Round Robin (This is the default and should be fine)
  - LC – Least Connections
  - DH – Destination Hash
  - SH – Source Hash
- What server to fall back to if ALL the real servers fail (the default is the local maintenance page)

- The local fallback server is an NGINX instance on port 9081
- Use the command lb2ports for layer 4 environments to put NGINX on port 80 & 9081
- Use the command lb1port to move it back to port 9081 only for compatibility with HAProxy on port 80.

- The type of health checks to carry out on the real servers:
  - Connect – This is the default just check that a server is responding correctly
  - Negotiate – Request a specified URL and check that the response is as expected
  - External – Use a custom file for the health check , specify the filename in the Check Command field
  - Off – All real servers are off line
  - On – All real servers are always on line
  - Ping – ICMP Ping check
  - 5 – Do a connect check 5 times then one negotiate then repeat
  - 10 – Do a connect check 10 times then one negotiate then repeat
- Service to check -
  - HTTP – Requires filename and path + Text expected
  - HTTPS – Requires filename and path + Text expected
  - FTP – Requires optional username password & filename to check in default directory
  - IMAP - Requires username password
  - POP - Requires username password
  - LDAP
  - SMTP
  - NNTP
  - DNS
  - MYSQL
  - SIP
  - TELNET
  - NONE
- Protocol
  - TCP – The default
  - FWM – For virtual servers specified by a firewall mark
  - UDP – DNS & SIP
  - OPS - One packet UDP based scheduler
- Check Port - Specify a custom port for health checks
- Virtual Host - Specify a virtual host for the health check as well as real server IP address
  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.
- Login – Specify the login name to use for IMAP, POP3 or FTP accounts (negotiate check)
- Password – Specify the password to use
- File to check - Specify the URL checked if negotiate is the type of health check selected

- Response expected - Specify the string required to be present on the page returned by the URL
- Email Alerts – Specify the email address to send alerts when servers fail health checks
- Forwarding Method
  - DR – The default Direct Server Return
  - TUN – IP encapsulation
  - NAT - NAT (network address translation)
- Feedback Method
  - none – Don't measure the performance of the real servers
  - agent – Loadbalancer.org agent installed on each real server
  - http – Read an HTTP page from the real server on port 3333

Real Servers

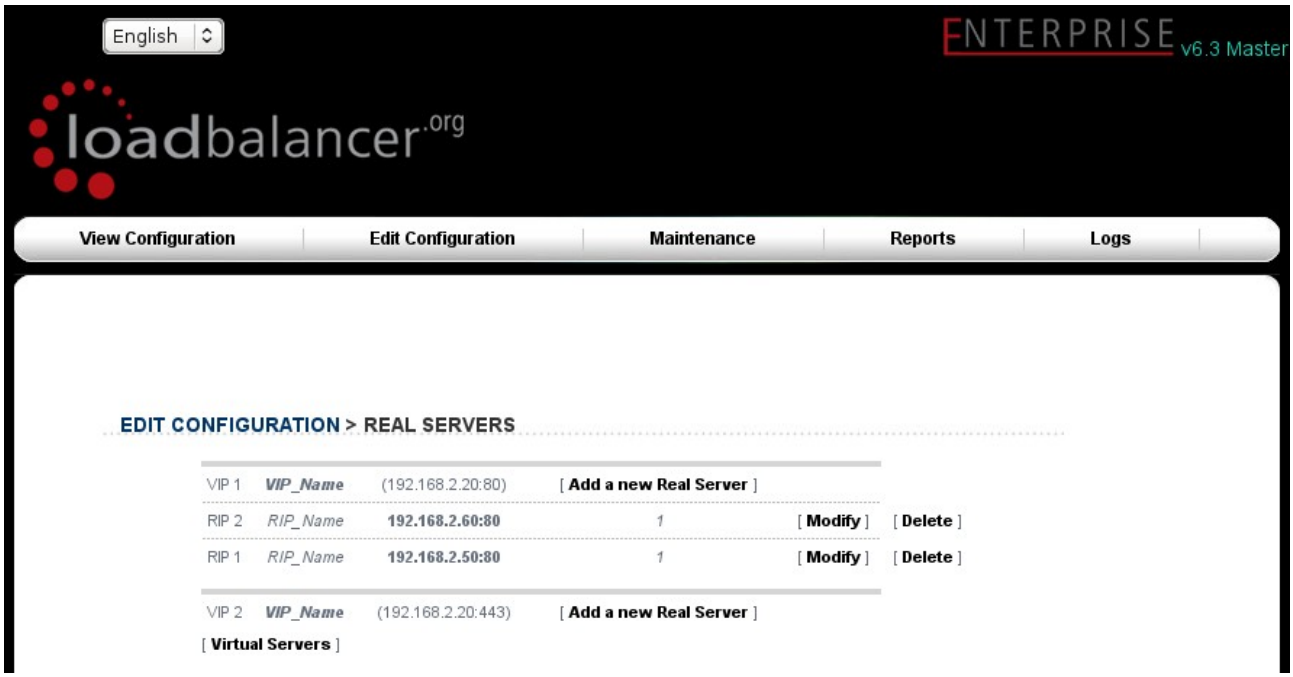
This menu option allows you to add, remove or modify Real Servers from your cluster.

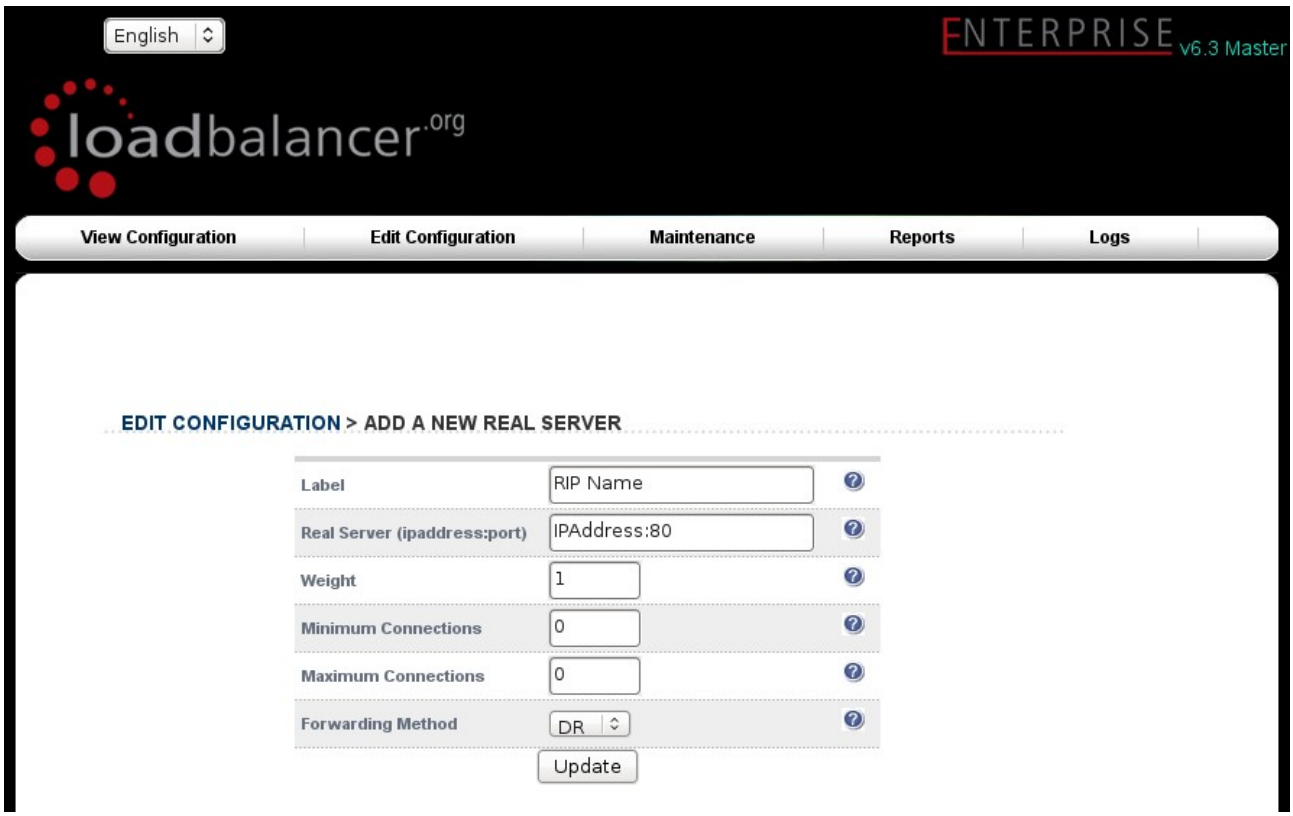
Each Virtual Server has a number of Real Servers. A Virtual Server can have any number of real servers in its cluster.

A real server is a combination of IP address and port number in the following format: *ipaddress:port* i.e. 192.168.1.101:80 for a web server.

NB. The port number is usually the same as the parent virtual server i.e. Virtual port 80 on the virtual IP address goes to real IP address on a real server and real port 80. In fact it must be for DR mode.

From the overview you can see each web server in the cluster, the IP address, port number and the requested relative weight (0 is off line).





Adding a new real server to a cluster is a simple case of specifying IP address, port number and weight.

The forwarding method defaults to that defined for the virtual server and you will normally leave this as DR. NAT can be used when you have two Floating Virtual IP(s) set up (one internal and one external) and TUN can be used to route through a tunnel across the Internet or WAN.

Selecting modify will bring up a similar dialog where you can change the details. This is the normal way that you would change the weight (priority) of a server.

Why would you change the weight of a real server? Say you had a 4 core Xeon web server and a single core Celeron web server, you could increase the weight of the Xeon based server so that it took more of the load.

Logical Layer 7 Configuration

If you require SSL termination or http cookie insertion to be carried out on the load balancer then this is done through the Logical Layer 7 Configuration.

Virtual Servers (HAProxy)

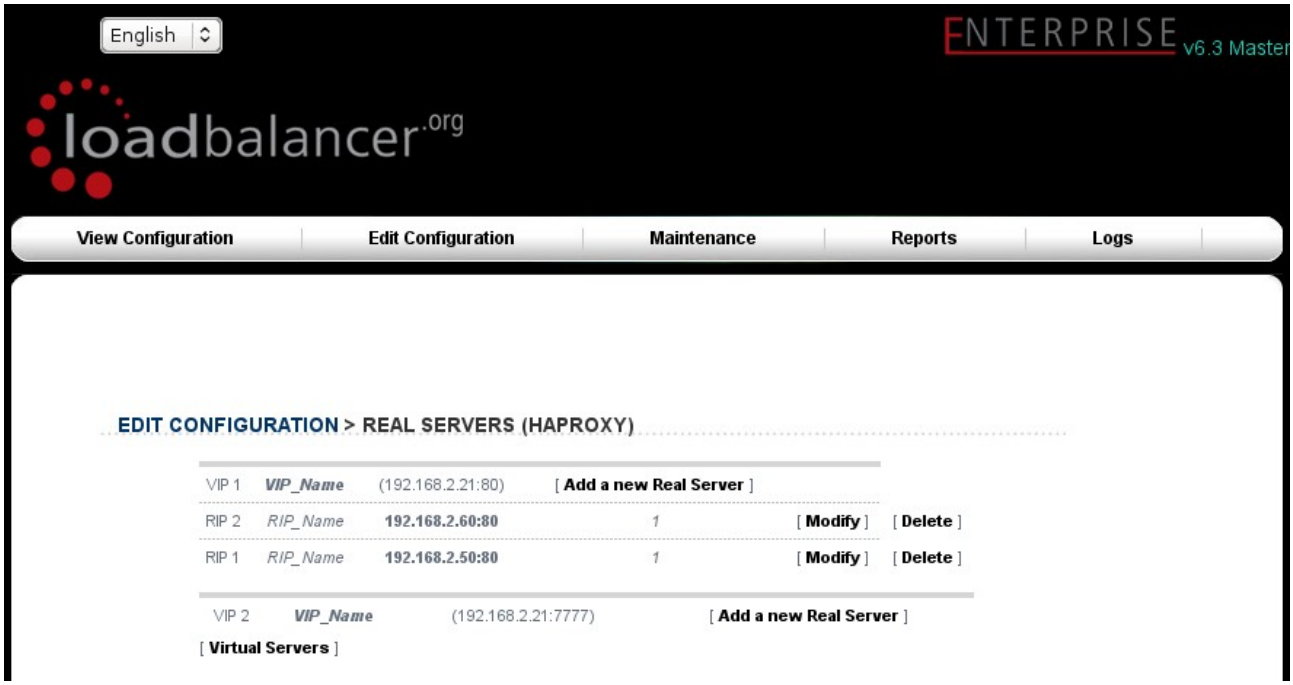
The Layer 7 Virtual Servers are configured separately from the Layer 4 ones because they use the HAProxy engine rather than the LVS engine.

Virtual Server (HAProxy) VIPs are created in the usual way by specifying a Virtual IP address and port for the service. If *persistence=no* then weighted round robin load balancing is performed. If *persistence=yes* and the *mode=tcp* then persistence by source IP is used.

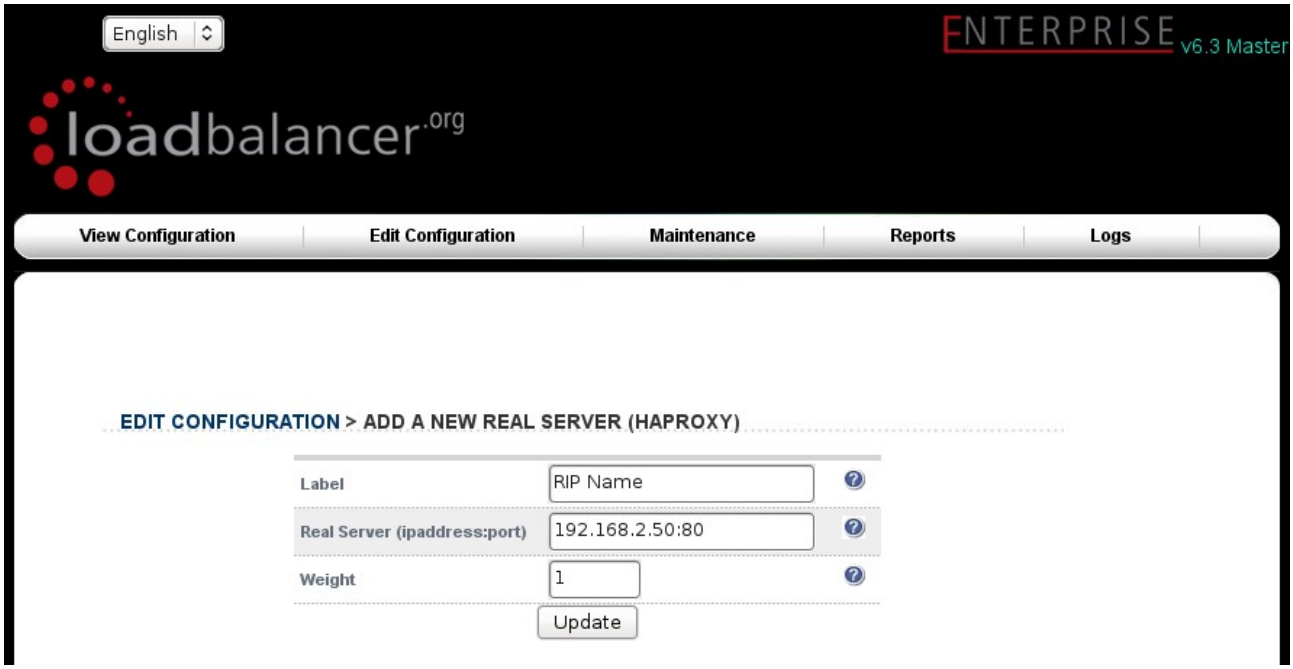
However if *persistence=yes* and the *mode=http* then the load balancer will automatically insert a cookie into each http request with the same name as the original destination server name. Therefore it is important that each real server is given a unique label when using cookie persistence.

Real Servers (HAProxy)

The Real Servers in a Layer 7 configuration can be on any subnet in any network as long as they are accessible from the load balancer.



The Real Servers are specified by *IPAddress:Port*, label and weight. The Real Servers can be a different port and a different subnet because the connections are proxied.

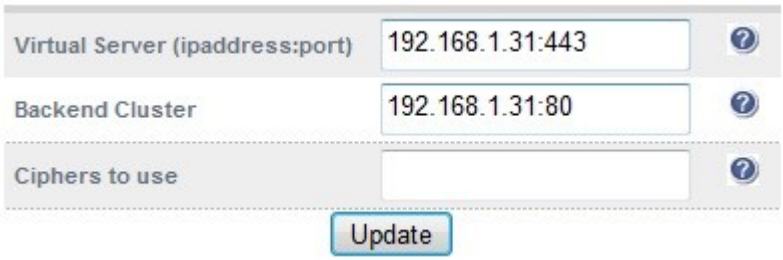


NB. Any changes to the Layer 7 configuration requires a restart of the HAProxy service. Restarting the service causes no downtime because it caches incoming connections while re-starting.

SSL Termination (Pound)

In order to set up a proxy for the SSL traffic go to *Edit Configuration* > *SSL Termination (Pound)*. SSL traffic can be terminated and then re-directed to port 80 of the same VIP for HAProxy to pick it up, insert cookies and load balance.

- Add a new Pound Virtual Server



- Configure the Virtual Server as 192.168.1.31:443
- Configure the Back-end as 192.168.1.31:80
- Click the Update button to add the new Virtual Server to the Pound configuration file.

IMPORTANT: You must now restart Pound in order to activate the changes i.e. *Maintenance* > *Services* > *Restart Pound*

By default a self generated SSL certificate is associated with the new Virtual Server. You can also upload your own certificate by selecting [**Modify**] for the Virtual Server, clicking Browse and navigating on your local machine to the *cert.pem* file (see the 'Create and Upload a PEM file' section in the following few pages) and click the upload button.

Manage this SSL certificate

In order to get a proper signed certificate from a Certificate Authority such as Verisign or Thawte you will need to generate a certificate request. This form will allow you to generate a CSR that is individual to this Virtual Server.

- Go to *Edit Configuration > SSL Termination (Pound)*
- Click [**Modify**] next to the VIP
- Click [**Manage this SSL Certificate**]

- Update the details as required and click **Manage this SSL Certificate (Server0)** (For subsequent Pound Virtual Servers, this will be Server1, Server2 etc.)

The CSR is then generated:

You can now copy the *Certificate Signing Request* (CSR) from the top pane and provide it to your Certificate Authority. They in turn will then sign the Certificate which you should paste into the second pane of the form and then click **Upload Signed Certificate**. Once the signed key is uploaded you will need to restart Pound-SSL (*Maintenance* > *Restart Pound-SSL*).

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

If you have already generated the CSR on your Web Server, you will need to create a PEM file using the Certificate and Private Key, then upload this using the interface (see section below).

Create and Upload a PEM file

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

```
----BEGIN CERTIFICATE-----
MIICsDCCAhmgAwIBAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIEwpTb21ILVN0YXRlMSEwHwYDVQQKExhJbnRlcm5ldCBX
kU6DJupvN6U6PRi7+zcKqd8wUiY8+3CyYKHtJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBNkyHYbQTvU8MeR3ilhe2fw+qVE2pgxWYWaGm8QwTsxQKgbxiG
2bShC2AVC+ZDMNu6bvCdvfySi6EypUcIvEwao7ZbyaAEbcSVympQJdgs6W6ajiLS
SXUWIWb0+k2j2L1z2PszFxwCIwQ=
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwIm3HyoVxjrymusOeIFgZlWyuaebIrreCpIo+iy
pSxEruhpqmdj2tYIpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPLAfmh88bS7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgRUQPiLosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUcIvEwao7ZbyaAEbcSVympQJdgs6W6ajiLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HUlrTqwES4Lun
xC6pIPck+XLBU1qyijfP34xRWcS1obm0momsDxc0RvpKxWO58lxIsQq1dCKrtKfO
SlqR7x/WAQUnFKVxQAMDatpeXSp3FGgXF+mpffusjEw=
-----END RSA PRIVATE KEY-----
```

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

Adding an Intermediate key to the certificate chain

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

NB. Your current signed key is stored in /usr/local/etc/serverX.pem

When you select [**Manage this SSL Certificate**] on a pre-configured certificate it will show a copy of the fully signed PEM file:

Select the whole of the text in the top pane and paste it into a text editor such as notepad (*not* Word!):

```
----BEGIN CERTIFICATE-----
MIICsDCCAhmgAwIBAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIEwpTb21lLVN0YXRlMSEwHwYDVQQKExhJbnRlcm5ldCBX
kU6DJupvN6U6PRi7+zcKqd8wUiY8+3CyYKHtJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBNkyHYbQTvU8MeR3ilhe2fw+qVE2pgxWYWaGm8QwTsxQKgbxiG
2bShC2AVC+ZDMNu6bvCdvfySi6EypUcIvEwao7ZbyaAEbcSVympQJdgs6W6ajiLS
SXUWIWb0+k2j2L1z2PszFxwCIwQ=
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwIm3HyoVxjrymusOeIFgZlWyuaebIrreCpIo+iy
pSxEruhpqmdj2tYlpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPLAfmh88bS7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgRUQPiLosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUcIvEwao7ZbyaAEbcSVympQJdgs6W6ajiLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HUlrTqwES4Lun
xC6plPck+XLBU1qyijfP34xRWcS1obm0momsDxc0RvpKxWO58lxIsQq1dCKrtKfO
SlqR7x/WAQUnFKVxQAMDatpeXSp3FGgXF+mpffusjEw=
-----END RSA PRIVATE KEY-----
```

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

```
----BEGIN CERTIFICATE-----
MIICsDCCAhmgAwIBAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIEwpTb21lLVN0YXRlMSEwHwYDVQQKExhJbnRlcm5ldCBX
kU6DJupvN6U6PRi7+zcKqd8wUiY8+3CyYKHtJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBNkyHYbQTvU8MeR3ilhe2fw+qVE2pgxWYWaGm8QwTsxQKgbxiG
2bShC2AVC+ZDMNu6bvCdvfySi6EypUcIvEwao7ZbyaAEbcSVympQJdgs6W6ajiLS
SXUWIWb0+k2j2L1z2PszFxwCIwQ=
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwIm3HyoVxjrymusOeIFgZlWyuaebIrreCpIo+iy
pSxEruhpqmdj2tYlpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPLAfmh88bS7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgRUQPiLosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUcIvEwao7ZbyaAEbcSVympQJdgs6W6ajiLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HUlrTqwES4Lun
xC6plPck+XLBU1qyijfP34xRWcS1obm0momsDxc0RvpKxWO58lxIsQq1dCKrtKfO
SlqR7x/WAQUnFKVxQAMDatpeXSp3FGgXF+mpffusjEw=
-----END RSA PRIVATE KEY-----
----BEGIN CERTIFICATE-----
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgRUQPiLosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUcIvEwao7ZbyaAEbcSVympQJdgs6W6ajiLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HUlrTqwES4Lun
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgRUQPiLosSmUCZ
ctnhgedfdfwwddddfefN+kugDxlgCT
-----END CERTIFICATE-----
```

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

Windows Servers

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

Windows Server is only able to export a private key file in .pfx format. Thus, we must use the program OpenSSL to perform the conversion for us.

There are two approaches to accomplishing the conversion, and can involve using either Windows or a UNIX-like Operating System.

Using Windows:

OpenSSL is available as a binary package for Windows at:

http://www.slproweb.com/download/Win32OpenSSL-1_0_0a.exe

Please download and install this package. There are no special instructions for this. You will now have an OpenSSL directory located on your file system. Click `START`, `RUN` then type `cmd.exe`. You need to navigate to the path where you installed your OpenSSL binaries. Within this directory chdir to `bin`

To convert .PFX to .PEM

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

To convert .CER file to .PEM format:

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

Using UNIX:

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

To convert .PFX to .PEM

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

To convert .CER file to .PEM format:

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

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

Import certificates exported from Windows Server

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

1) Once the certificate is working correctly on your Windows server, export the certificate from Windows. The format will be .pfx.

2) Now download openssl from : http://www.slproweb.com/download/Win32OpenSSL-1_0_0a.exe and install this on your PC.

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

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

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

4) Now upload this pem file to the Pound Virtual Server on the load balancer:

- goto Edit Configuration > SSL Termination (Pound)
- click [**Modify**] next to the relevant Pound Virtual Server
- click Browse and select the pem file created in step 3
- click Upload PEM file

Now restart Pound (*Maintenance* > *Restart Pound-SSL*)

Converting an encrypted private key to an unencrypted key

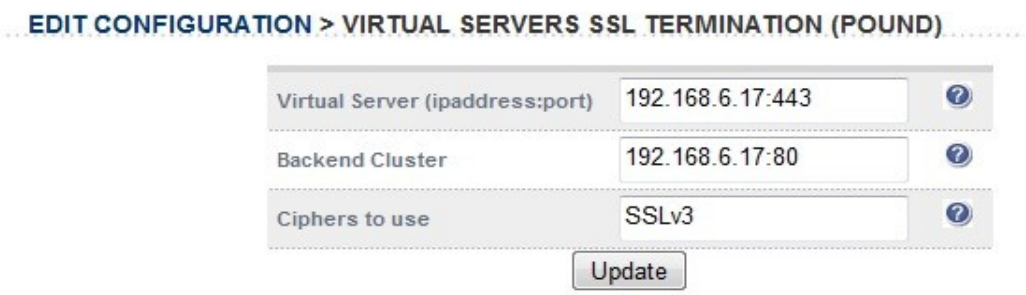
If a password has been included in the private key, this should be removed before it is used with your pem file. This can be done using the following method:

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

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

Limiting Ciphers

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



Physical Load Balancer Configuration

Network Interface Configuration

This form allows you to modify the physical IP address of the load balancer.

WARNING: Obviously it's safer to do this with access to the local console.

The *eth0* interface is for the internal network and is the only network you need for the default Direct Routing configuration. If you want to use NAT mode, then you will also need to configure the external network *eth1*, or define an alias on eth0.

Aliases

Aliases for the load balancers interfaces can be defined here. This may be needed for example when you want to use just one physical interface, but want the load balancer to be dual-homed in two different subnets.

VLANS

802.1q VLANS can be defined here. This is typically required if your real servers are connected to specific VLANS. The exact requirements depend on your infrastructure.

If you have a master and slave, you'll need to configure any Aliases or VLANS on the slave manually since these are not currently replicated automatically

DNS & Hostname

After both lbslave & lbmaster are configured with the correct IP addresses and host names you need to tell lbmaster the slave load balancers IP address. Once this is done all changes will be replicated correctly to the slave load balancer.

Force full slave sync will transfer all settings from the master to the slave (useful if you modified logical settings before setting up the replication).

Entering a *Domain Name Server* will allow any reports that reverse lookup IP address info to work correctly and will also allow on-line updates via the Loadbalancer.org web site.

Floating IP(s)

In order for the load balancer to work the box must physically own the Virtual IP address that the clients are accessing before they get re-directed to a real server in the cluster. The Floating Virtual IP(s) are controlled by heartbeat to ensure that only one of the load balancers (normally the master) owns the Floating Virtual IP(s). You can add as many Floating Virtual IP(s) as you like.

NB. If you are configuring two servers in fail over then it is recommended that you configure the load balancers hostname, then the IP address on both servers, then tell lbmaster the IP address of lbslave. This will let all changes configured on lbmaster to be automatically replicated to lbslave.

Setup Wizard

This option runs the setup wizard.

Upgrade License Key

This option allows a license key to be entered to unlock the R16 restrictions. The key is provided when a Enterprise license is purchased.

Advanced

Execute a shell command

This allows you to remotely execute a shell command as a root user. Useful if you accidentally kill your SSH server or something.

WARNING: You should really know what you are doing if you use this function.

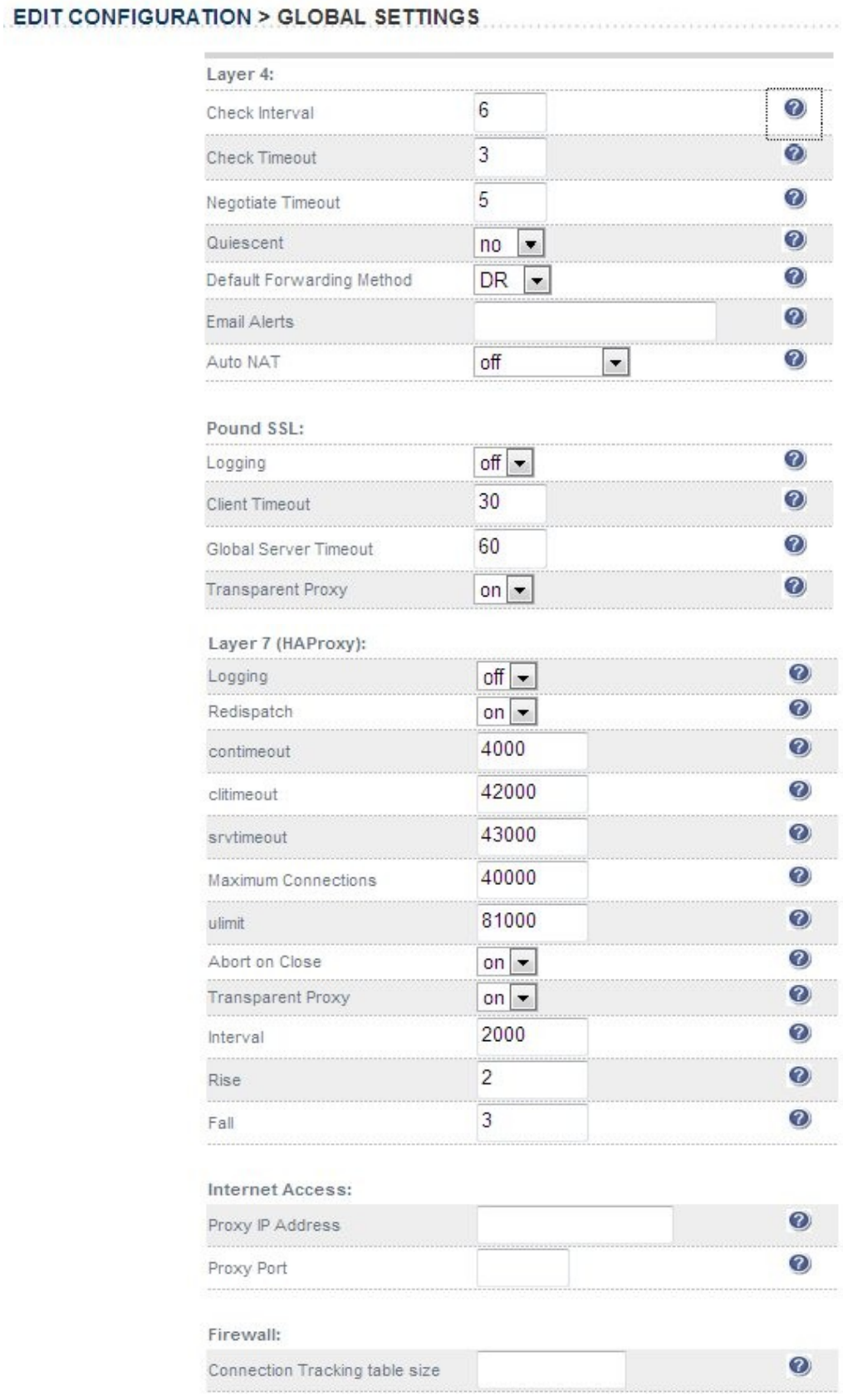
The output of the command will be displayed on screen.

Heartbeat Configuration

This allows you to control the behavior of the HA-Linux implementation:
Here you can specify whether the heartbeat is over serial cable, network cable or both.
NB. If you disable heartbeat over serial cable this automatically enables console re-direction over the serial port.
You can also configure a network ping node, this should be mutually accessible from both the master and slave load balancer, e.g. the default gateway.

For faster fail over times you can reduce the *Keepalive* and *Deadtime* settings.

Global Settings



Layer 4

Check Interval

Layer 4 (Ldirectord) health check interval in seconds. If this setting is too low, you may induce un-expected real server downtime.

Check Timeout

Layer 4 (Ldirectord) health check timeout in seconds. If this setting is too low, you may induce un-expected real server downtime.

Negotiate Timeout

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

Quiescent

When Quiescent='Yes' the real server is set to a weight of 0 on health check failure. When Quiescent='No' the server is completely removed from the load balancing table.

Email Alerts

Specify the global email alert address. The global email alert address is used to send notifications of real server health check failures. This can also be configured on a virtual server level.

Auto NAT

Automatically NAT outbound network connections from internal servers. By default servers behind the load balancer in a NAT configuration will not have access to the outside network. However clients on the outside will be able to access load balanced services. By enabling Auto NAT the internal servers will have their requests automatically mapped to the load balancers external IP address. The default configuration is to map all requests originating from internal network eth0 to the external IP on eth1. If you are using a different interface for external traffic you can select it here. Manual SNAT and DNAT configurations for individual servers can also be configured in the firewall script.

Pound SSL

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 normally require changing.

Global Server Timeout

Configure the global real server response timeout in seconds. This setting should not normally require changing.

Transparent Proxy

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 (see the Haproxy section below) and also adds appropriate rules to the firewall.

One consequence of using transparent proxy with both Pound and HAProxy is that you can no longer access the HAProxy virtual service directly. With transparency turned on HAProxy will only accept traffic from Pound. The way around this is to create two HAProxy virtual services. One will listen on port 80, and be your standard HTTP service. The other will listen on a different port – 81 for example - and will be the destination for traffic from Pound.

Layer 7 HAProxy

Logging

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

Redispatch

Allows HAProxy to break persistence and redistribute to working servers should failure occur. This setting should not normally require changing.

Contimeout

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

Clitimeout

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

srvtimeout

HAProxy real server timeout in milliseconds. This setting should not normally require changing.

Maximum Connections

HAProxy maximum concurrent connections. This setting should not require changing, unless you are running a high volume site. See also Maximum Connections for a virtual Server (HAProxy).

ulimit

The maximum number of file descriptors used for layer 7 load balancing. This value is optional. If no value is given then a default value will be used internally. For simple configurations where each virtual server only listens to one address/port a reasonable value is the sum of:

- 2 times the number of maximum connections (Global Settings Layer 7)
- Number of virtual servers on layer 7 (HAProxy)
- Number of real servers
- Plus 1 for logging purpose

In a more sophisticated environment you should use the number of address/port/proxy tuples instead of the number of virtual servers.

Abort on close

Abort connections when users close their connection. Recommended as the probability for a closed input channel to represent a user hitting the 'STOP' button is close to 100%.

Transparent Proxy

Enable TPROXY support for Layer 7 HAProxy. TPROXY enables 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 (i.e. both internal and external subnets) with the real servers using a floating IP address configured on the load balancer as their default gateway.

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

X-Forwarded-For Headers

Since the load balancer must be in a NAT configuration (i.e. VIPS & RIPS in different subnets) to utilize TPROXY, it is not always an appropriate solution. In situations such as this, it's possible to use the X-forwarded-for header in layer 7 Virtual Servers. Starting in V6.14 is a new check-box for each layer 7 virtual Server that allows the XFF header to be enabled/disabled, by default this is enabled for all layer 7 Virtual Servers. Most web servers can then be configured to record the X-Forwarded-For IP in the log files. For example, with Apache it's simply a change to the log file configuration. For details on how to do this please refer to:

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

With Microsoft IIS a third party application is needed. For more details on this, please refer to:

http://blog.loadbalancer.org/iis-and-x-forwarded-for-header/

Interval

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

Rise

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

Fall

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

Internet Access

Proxy IP and Proxy Port for Internet Access

State your Proxy Server's IP address and port here. For internet access via a proxy. You will need this for an online update if your load balancer is behind a proxy server. Leave both fields empty if you don't use a proxy.

Firewall

Connection tracking table size

High traffic load balancers using NAT mode, or using connection tracking in the firewall script, may see the connection tracking table fill up. Systems experiencing this problem will report the following in the kernel log: ip_conntrack: table full, dropping packet. The table size may be set here, in units of active connections. Each connection entry uses approximately 300 bytes of memory, and the default table size is approximately 30,000 connections.

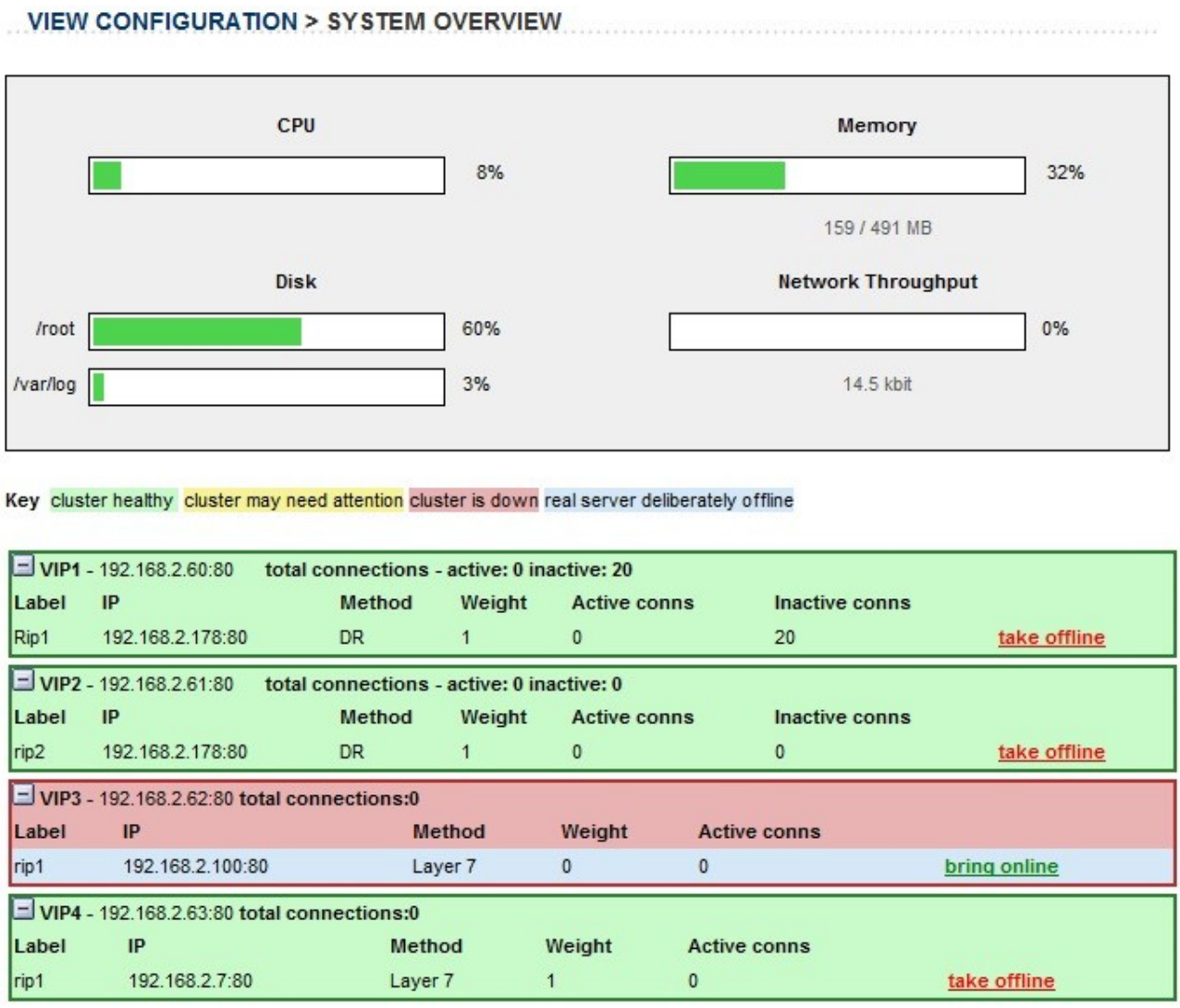
Maintenance

Maintain Real Servers

System Overview

This shows a quick overview of system resources and also the status of all layer 4 & layer 7 virtual servers running on the load balancer.

The take offline and bring online links allow the real servers to be brought online and offline, for example to perform server maintenance.



Take a real server offline or online

This function is now deprecated. Please use the System Overview instead

Backup & Recovery

Your Loadbalancer.org appliance is covered by a full on-site warranty, and re-configuration is simple from the default install BUT it's always nice to have a backup.

Configuration Backup

This option will instantly backup the current configuration to the local disk, this is useful when you want to make a major change and yet have the ability to roll back quickly if it didn't have the desired effect.

Disaster Recovery

This section gives you the following options :

- Download XML configuration file -
- Download firewall script -
- Upload XML Config file – To upload a previous XML file and activate the settings
- Restore last online backup
- Restore manufacturers settings – Handy if you want to start all over again

Services

Restart HAProxy

Any configuration changes to the Layer 7 (HAProxy) configuration, including server weights, will automatically restart HAProxy. If you get a failure to start please check your configuration to ensure you are not binding to ports that are already in use.
Restarts of HAProxy are completely graceful whether they succeed or not.

Restart Pound-SSL

Any configuration changes to the SSL termination configuration or server certificates will require a restart of Pound. If you get a failure to start please check your configuration to ensure you are not binding to ports that are already in use.

Restart Heartbeat

Heartbeat controls the fail over between the master and slave load balancer, if you make any changes to the physical IP address then this will automatically restart heartbeat (*on a properly configured cluster this will also force a heartbeat restart on the slave*).

Restart Ldirectord

It is unlikely that you will ever need to use this function. It just re-loads the health check configuration file. This does not usually result in any down time for the cluster.

Power Control

Shut down and restart server

Restarts the load balancer.

Shut down and halt server

Halts the load balancer.

Security & Maintenance

Online Software Update

If you have a valid software maintenance license for your appliance you can use this form to check for the available online updates and install them.

- You will need a valid authorization code.
- You will need your default gateway & DNS correctly configured.
- You will need HTTP access to www.loadbalancer.org enabled through your firewall.

Updates are also available as a complete downloadable ISO software image if preferred.

NB. You will need to update both the Master & the Slave (normally the slave is updated first followed by the master). In some cases you may need to reboot or do a service httpd restart to get online update to recognize a DNS change.

Fallback Page

This section allows you to view and modify the local holding page on the load balancer. This page will only be shown if ALL of the real servers in a cluster are unavailable. If you have a master and slave load balancer then you must change this on both servers.

You can use any valid HTML for the default page, simply cut and paste from your favorite editor.

The fallback server on the load balancer is an implementation of NGINX.

Layer 7

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

Layer 4

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

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

Holding page on a dedicated server

Set the Fallback Server field of the VIP to the IP address:port of the fallback server, e.g. 192.168.2.10:80

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

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.

Holding page on the Load balancer

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

```
lb2ports
```

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

Firewall Script

Similar to the modify maintenance form this allows you to directly edit /etc/rc.d/rc.firewall.

WARNING: BE CAREFUL! Make a backup before changing this script so that you know you can roll everything back if you cause a problem.

If you wish to clear the firewall tables completely use the following command from the console:

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

This can either be used for belt & braces security; for example to replicate your normal firewall settings onto the load balancer as well for double security. What kind of settings? Well normally you don't want any customers to be able to access the administration IP address on the load balancers, you only want them to have access to, say port 80 & 443 on the VIP interface.

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

If you are planning to use NAT you may also want to use the load balancer as your main firewall which is fine, but we think it is a lot simpler to keep your firewall separate from your load balancer. Especially if you want to set up VPNs etc.

A firewall script would typically only allow the administrator access to the load balancer and allow the traffic for the defined Virtual Services. This can be automated using the firewall lock down wizard.

Firewall Lock Down Wizard

The firewall lock down wizard prompts you for an administration IP address that will be given sole access to the administration ports on the load balancer 80,443,9080,9443 & 22.
If you need to specify an administration network just change the network mask.
The lock down wizard will allow full public access to all the defined VIPs and reply traffic from the defined real servers.

The generated script is stored here: /etc/rc.d/rc.lockdownwizard
This script is activated at the end of the /etc/rc.d/rc.firewall script.
Any changes that you have already made to the /etc/rc.d/rc.firewall script are kept in place.

An example of the script generated:

```
#!/bin/sh
#/etc/rc.d/rc.lockdownwizard
# Auto generated by Loadbalancer.org appliance
# Make sure the default INPUT policy is drop
iptables -P INPUT   DROP
# Allow unlimited traffic on the loopback interface for local administration
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
# Define an administration IP address or subnet
ADMINIP="192.168.1.73"
ADMINSUBNET="255.255.255.255"
# Grant the administration IP address access
iptables -A INPUT -p tcp -s $ADMINIP/$ADMINSUBNET -m multiport --destination-port
80,443,9080,9443,22 -j ACCEPT
# Layer 4 VIPs
iptables -A INPUT -p tcp -d 192.168.1.21 --dport 3389 -j ACCEPT
iptables -A INPUT -p tcp -s 10.0.0.14 --sport 3389 -j ACCEPT
# Layer 7 VIPs
# SSL VIPs
iptables -A INPUT -p tcp -d 192.168.1.21 --dport 81 -j ACCEPT
iptables -A INPUT -p tcp -s 10.0.0.14 --sport 80 -j ACCEPT
```

NB. If you accidentally block your own access to the appliance you will need to clear the current firewall rules and try again.

If you wish to clear the firewall tables completely use the following command from the console:

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

IMPORTANT! The firewall lockdown wizard should only be run once the load balancer is fully configured and tested. Also, if changes are made later to the load balanced services, the wizard should be re-run to ensure these changes are included in the lockdown script.

Initialize Graphs (rrdtool)

Once you have configured all of your virtual and real servers you will probably want to initialize the statistics tracking database. Clicking this menu option will construct a series of RRDTool databases and relevant cron jobs to update those databases using the output from LVSGSP. More cron jobs are then used to generate the daily, weekly, monthly and yearly charts accessible from the reports section.

WARNING: All of your old statistics will be lost when you use this function.

From v6.7 graphs for Network Throughput, CPU stats and Layer 7 HAProxy VIPs are created in addition to the layer 4 stats.

Passwords

This section allows you to manage the user accounts that have access to the web based administration system, any changes you make will need to be done on both lbmaster and lbslave.

The administration account is *loadbalancer* and its default password is *loadbalancer.* This account cannot be deleted but the password should be changed.

When you modify a user you can select its security group from either:

- Conf – Configuration access (same as the loadbalancer account)
- Maint – Maintenance access ability to take servers on and offline only
- Report – Access to the management reports only

NB. These passwords are simple apache .htaccess style password and nothing to do with the local Linux accounts for the root or loadbalancer users.

Resetting your WUI password

If you do forget your WUI password, it is possible to reset it. To to this you'll need root access to the console or terminal session. At a command prompt type:

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

WARNING: Don't forget to change you root password from the console using the *passwd* command!

Reports

System Overview

The System Overview gives a real time look at the status of each load balanced service.

It also shows real time load and performance statistics. You can take individual real servers offline or bring them back online. This is useful if you need to take a server offline for maintenance. When you request a server to go online or offline it will normally take 5 seconds to change the active status.

NB. If you request ALL the real servers to be OFFLINE then the fallback server will NOT be activated.

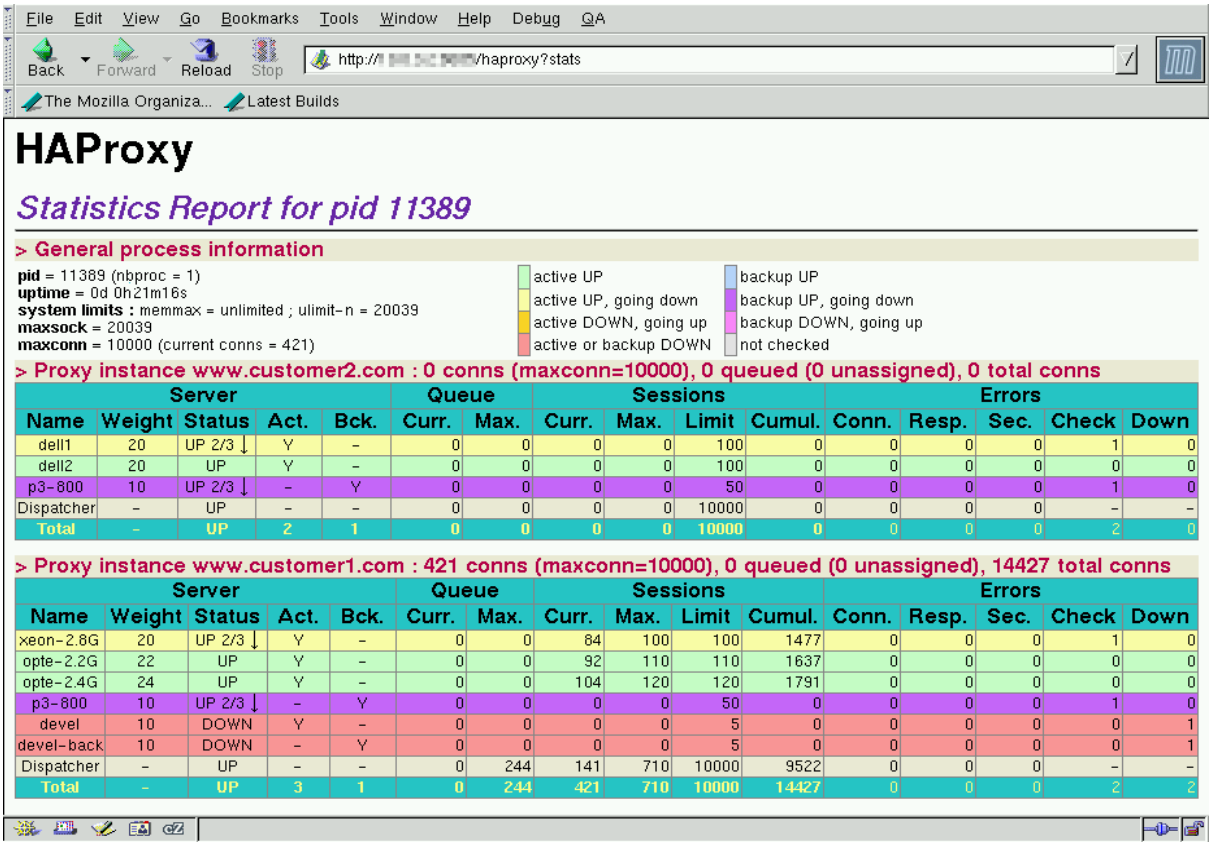
Status

This live report shows the current number of active and inactive connections for each configured real server. It also shows the current weight of each real server. If the weight is 0 or the real server is not showing (QUIESCENT=no) this means that the health checker has failed for that real server, check the ldirectord log file to confirm if required.

NB. The maintenance screen clearly shows the real time status of virtual and real servers using the raw data from this report.

Status (HA Proxy)

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. Login using username: *loadbalancer*, password: *loadbalancer*.



Traffic Rate Per Second

This report shows the current connections per second and bytes per second to each real server.

Traffic Qty

This report shows the volume of traffic to each real server since the counters were last re-set.

Current Connections

The current connections report is very useful for diagnosing issues with routing or ARP related problems.

NB. These reports are generated in real time. Direct Routing is the default load balancing method and you will not see any stats for return packets (as they do not pass through the load balancer). You would however see them if you were in NAT mode.

Current Connections (Resolve Host name)

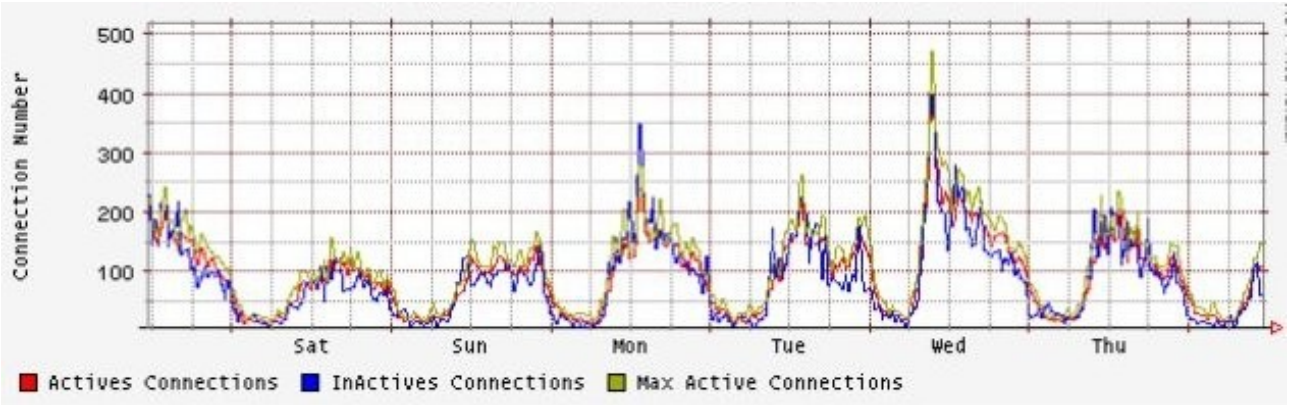
This is the same as the current connections report but is much slower as it looks up the DNS name of each IP address.

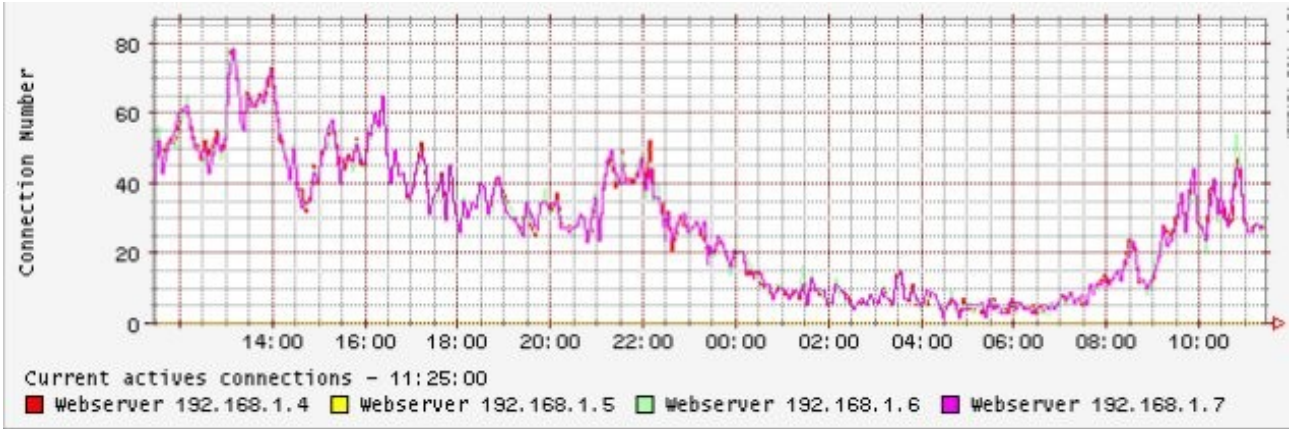
Graphical Stats Over Time

This link goes to a page generated by the Initialize *Graphs (rrdtool)* command. The graphs generated are great for showing management pretty pictures they may understand.

Why does the average activity get lower over time?

There is a good mathematical reason for this, but the graphs now also show max connections as well as average connections.





Logs

Ldirectord

The ldirectord log shows the output from the health checking daemon. This is useful for checking how healthy your real servers are or pinning down any configuration errors. The logging here can be quite verbose but it clearly shows what the health checking daemon is doing.

Lbadmin

The Lbadmin log shows any changes made via the admin system. Useful to prove that someone has changed something that they should not have done.

Heartbeat

The heartbeat log shows the status of the heartbeat between the master and slave nodes. Don't worry about the various memory usage messages in this log they are their to prove that that everything is working fine.

HAProxy

If activated as a global option, this will show the contents of /var/log/haproxy. This is a very detailed log of all transactions through HAProxy.

Pound (SSL)

If activated as a global option, this will show the contents of /var/log/poundssl. This is a very detailed log of all transactions through Pound SSL

Reset all packet counters to zero

As it says this resets the packet counters to zero for the load balancer reports.

Change the date/time settings

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

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

So GMT+X means GMT - X hours and vice-versa.