



## **Appliance Administration Manual**

**v6.7**



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

*Copyright © 2010 Loadbalancer.org Ltd*

# Table of Contents

Section A – Introduction.....	6
Appliance Details.....	7
Initial Configuration.....	7
Further Information.....	7
Section B – Fundamental Load balancing Concepts.....	8
Load balancing Algorithms.....	9
Round Robin.....	9
Weighted Round Robin (Default method).....	9
Least Connection.....	9
Weighted Least Connection.....	9
Destination Hashing.....	9
Source Hashing.....	9
Agent Based.....	9
Section C - Quick Start Guide.....	10
Loadbalancer.org terminology.....	11
What is a virtual IP address?.....	11
What is a floating IP address?.....	11
What are your objectives?.....	12
What is the difference between a one-arm and a two-arm configuration?.....	13
What are the different load balancing methods supported?.....	14
Network diagram: One-Arm – DR Direct Routing (single unit).....	16
Network diagram: Two-Arm - NAT Network Address Translation (single unit).....	17
High-availability configuration of two Loadbalancer.org appliances.....	18
Network diagram: Two-Arm - NAT Network Address Translation (clustered pair).....	18
Network diagram: One-Arm – DR Direct Routing (clustered pair).....	19
Unpacking and setting up the Loadbalancer.org appliance.....	20
Configuring the Loadbalancer.org appliance using the web based wizard.....	21
Network interface configuration.....	21
Configuring the Loadbalancer.org appliance using the console wizard.....	22
Network interface configuration.....	22
Example answers using the wizard for a two-arm NAT configuration.....	23
Additional Loadbalancer.org configuration (web interface).....	24
Additional real servers (web interface).....	25
Real server (back-end) configuration for NAT mode.....	26
Real server (back-end) configuration for DR mode (Linux).....	26
Real server (back-end) configuration for DR mode (Windows).....	27
Resolving ARP issues for Windows server 2000 & 2003 (DR mode only).....	28
Installing the Microsoft loopback adapter.....	28
Configuring the loopback adapter.....	29
Resolving ARP issues for Windows server 2008 (DR mode only).....	31
Weak and strong host behavior in Windows.....	31
Testing the load balancer configuration.....	33
Connection error diagnosis.....	33
Health check diagnosis.....	34
Testing high-availability for a Loadbalancer.org HA-pair.....	34
Does your application cluster correctly handle its own state?.....	34
Replication solutions for shared data.....	35
Solutions for session data.....	35
What do you do if your application is not stateless?.....	36
Loadbalancer.org persistence methods.....	36
Section D – Typical Deployment Examples.....	37
Example 1 – single appliance (web interface).....	38
Network interface configuration.....	38
Example 2 - clustered pair (web interface).....	39
Network interface configuration.....	39
Configuring the Virtual Servers (VIP) in one-arm DR mode.....	40

Layer 4 configuration.....	40
Real Server configuration (RIP) in one-arm DR mode.....	41
Configuring the Virtual Servers (VIP) in two-arm NAT mode.....	42
Layer 4 configuration.....	42
Real server configuration (RIP) in two-arm NAT mode.....	43
Example 3: layer 7 configuration one-arm SNAT mode.....	44
Network diagram for layer 7 SNAT mode (clustered pair).....	44
Network diagram for layer 7 SNAT mode (single unit).....	45
Network Diagram for layer 7 SNAT mode (off site backup).....	46
Virtual Server configuration (layer 7 HAProxy).....	47
Real Server configuration (HAProxy).....	48
SSL Termination (Pound).....	49
Manage this SSL certificate.....	50
<b>Section E – Detailed Configuration Information.....</b>	<b>51</b>
Console configuration methods.....	52
Console access via a serial cable.....	52
Remote configuration methods.....	52
Advanced DR considerations.....	54
What is the ARP problem?.....	54
Solving the ARP problem.....	54
Solving for Linux (with arp_ignore sysctl values).....	54
Solving for Solaris.....	55
Solving for Mac OS X or BSD.....	55
Solving for Windows 2000 / 2003.....	55
Windows 2003 R2 / R1 (With SP1) Firewall Settings.....	58
Solving for Windows 2008.....	58
Windows 2008 R2 Firewall Settings.....	59
Windows 2008 R1 Firewall Settings.....	60
Advanced NAT considerations.....	61
Explaining the RIP & VIP in NAT mode.....	63
Network Diagram: one arm – NAT Network Address Translation (clustered pair) .....	64
Route configuration for Windows Server with one arm NAT mode.....	65
Route configuration for Linux with one arm NAT mode.....	65
Network Interface Configuration.....	66
Heath Monitoring.....	67
Loadbalancer Health.....	67
Heartbeat Configuration.....	67
Serial Cable.....	67
Unicast (ucast).....	67
Broadcast (bcast).....	68
Ping Node.....	68
Real Server Health.....	68
Configuration.....	68
Advanced firewall considerations.....	69
Firewall Marks.....	69
FTP.....	70
Limiting Passive Ports.....	70
For Linux.....	70
For Windows 2008.....	71
For Windows 2003.....	72
For Windows 2000.....	72
FTP Negotiate Health Check.....	73
FTP Recommended persistence settings.....	73
Persistence Considerations.....	74
Persistence > 15 minutes.....	74
Server maintenance when using persistence.....	74
Persistence State Table Replication.....	75
Terminal Server RDP Considerations.....	76
RDP – Layer 4.....	76
Layer 7 (RDP Cookies).....	76
Load balancing based on URL match with Pound.....	77
Load balancing based on URL match with Haproxy.....	78

NIC Bonding and High-Availability.....	79
Example 1: Bonding for bandwidth.....	79
Example 2: Bonding for High-Availability (recommended).....	79
Example 3: Bonding for High-Availability & Bandwidth.....	79
8021q VLAN support.....	80
SNMP Reporting.....	81
SNMP for Layer 4 based services.....	81
SNMP for Layer 7 based services.....	81
Feedback agents.....	82
Installing the Windows agent.....	82
Installing the Linux/Unix agent.....	83
Custom HTTP agent.....	83
Changing the local date, time & time zone.....	84
NTP Configuration.....	84
<b>Section F – Disaster Recovery.....</b>	<b>85</b>
Being Prepared.....	86
Backing up to a remote location.....	86
Backing up to the loadbalancer.....	86
Appliance recovery using a USB memory stick.....	87
Disaster Recovery after Master Failure.....	88
Disaster Recovery after Slave Failure.....	90
<b>Section G – Web User Interface Reference.....</b>	<b>91</b>
View Configuration.....	92
System Overview.....	92
XML.....	92
Layer 4.....	92
Layer 7 (Haproxy).....	92
SSL Termination (Pound).....	92
Network Configuration.....	92
Heartbeat Configuration.....	92
Heartbeat Resources.....	92
Routing Table.....	92
Firewall Rules.....	92
Edit Configuration.....	93
Logical Layer 4 Configuration.....	93
Virtual Servers.....	93
Real Servers.....	97
Logical Layer 7 Configuration.....	99
Virtual Servers (HAProxy).....	99
Real Servers (HAProxy).....	100
SSL Termination (Pound).....	101
Manage this SSL certificate.....	102
Adding an Intermediate key to the certificate chain.....	103
Import certificates exported from Windows Server.....	104
Using Windows.....	104
Using UNIX.....	104
Physical Load Balancer Configuration.....	105
Network Interface Configuration.....	105
DNS & Hostname.....	105
Floating IP(s).....	105
Setup Wizard.....	105
Upgrade License Key.....	106
Advanced.....	106
Execute a shell command.....	106
Heartbeat Configuration.....	106
Global Settings .....	107
Maintenance.....	108
Maintain Real Servers.....	108
System Overview.....	108
Take a real server offline or online .....	108
Backup & Recovery.....	109
Configuration Backup.....	109

Disaster Recovery.....	109
Services.....	109
Restart HAProxy.....	109
Restart Pound-SSL.....	109
Restart Heartbeat.....	109
Restart Ldirectord.....	109
Power Control.....	110
Shut down and restart server.....	110
Shut down and halt server.....	110
Security & Maintenance.....	110
Online Software Update.....	110
Fallback Page.....	110
Firewall Script.....	111
Firewall Lock Down Wizard.....	111
Scripted Alias configuration (for extra subnets).....	112
Initialize Graphs (rrdtool).....	112
Passwords.....	113
Reports.....	114
System Overview.....	114
Status.....	114
Status (HA Proxy).....	114
Traffic Rate Per Second.....	115
Traffic Qty.....	115
Current Connections.....	115
Current Connections (Resolve Host name).....	115
Graphical Stats Over Time.....	115
Logs.....	116
Ldirectord.....	116
Lbadmin.....	116
Heartbeat.....	116
HaProxy.....	116
Pound (SSL).....	116
Reset all packet counters to zero.....	116
Change the date/time settings.....	116

## **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 customised 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.*

## Further Information

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

## **Section B – Fundamental 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 summarise 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 weighted round robin would be more suitable.

### *Weighted Round Robin (Default method)*

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 assigns 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 runaway 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.

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

*(Also available as a separate download)*

## Loadbalancer.org terminology

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

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

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

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

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

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

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



In order to achieve all three objectives of performance, reliability & maintenance in a web based application, 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:

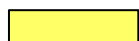
<b>One-Arm</b>	The load balancer has <b>one</b> physical network card connected to <b>one</b> subnet.
<b>Two-Arm</b>	The load balancer has <b>two</b> physical network cards connected to <b>two</b> subnets.

## 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 <b>Requires handling the ARP issue on the real servers</b>	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 - <b>Processor intensive</b>	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 <b>Much slower than Layer 4</b>	1 ARM

### **Key:**



Recommended



Recommended only 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.

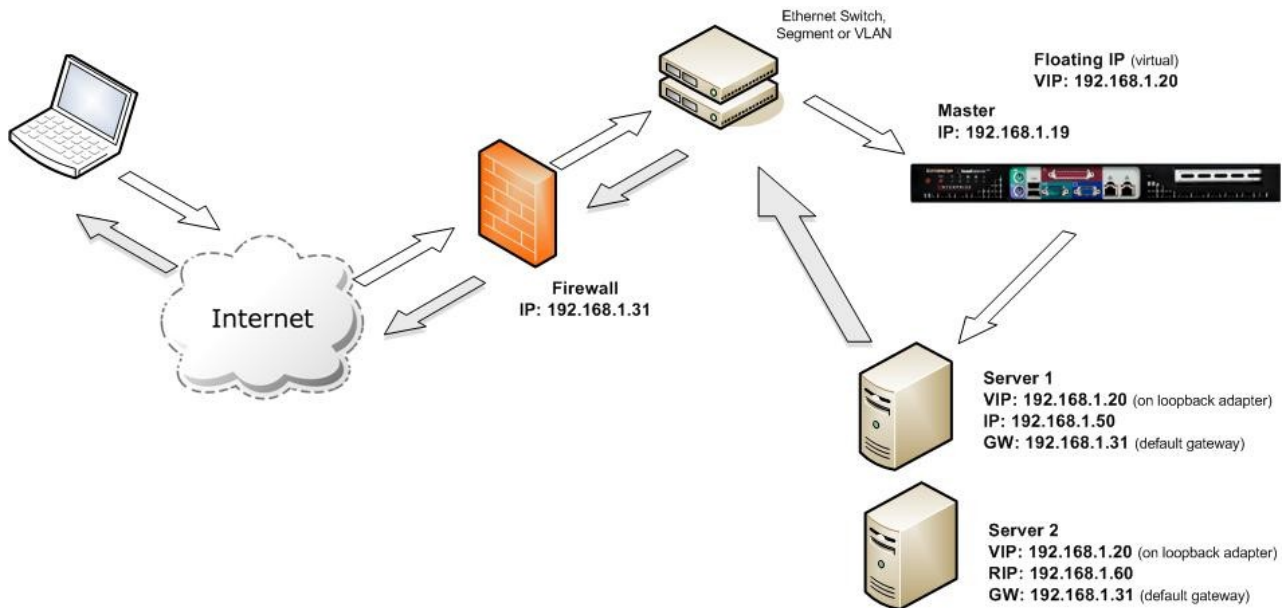
If your application requires that the load balancer handles cookie insertion then you need to use the SNAT configuration (HAProxy). This also has the advantage of a one-arm configuration and does not require any changes to the application servers. However, as the load balancer is acting as a full proxy 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.

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



#### Notes:

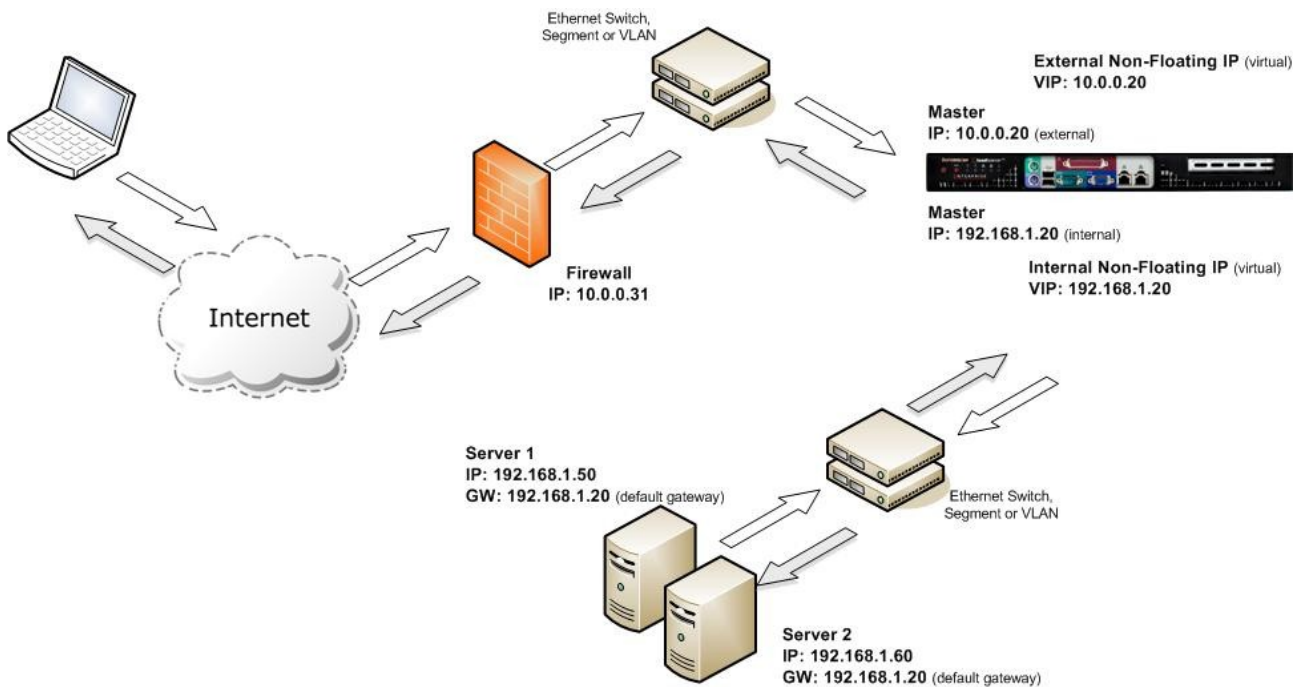
- When using a single load balancer unit only one IP address is required
- Direct routing works by changing the destination MAC address of the incoming packet on the fly which is very fast
- However, it means that when the packet reaches the real server it expects it to own the VIP. This means you need to make sure the real server responds to the VIP, but does not respond to ARP requests. Go to page 16 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 both the VIP & its own IP address
- Port translation is not possible in DR mode i.e. have a different RIP port than the VIP port



When using a load balancer in one-arm DR mode all load balanced services can be configured on the same subnet as the real servers. The real servers must be configured to respond to the virtual server IP address as well as their own IP address.



## 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 global options
- The real servers *must* have their default gateway configured to point at the load balancer
- Real servers are automatically given access to the Internet through the load balancer (*autonat*)
- 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 the advanced NAT considerations section of the administration manual for more details on these two issues



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.



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 D provides more detail for one-arm NAT mode.

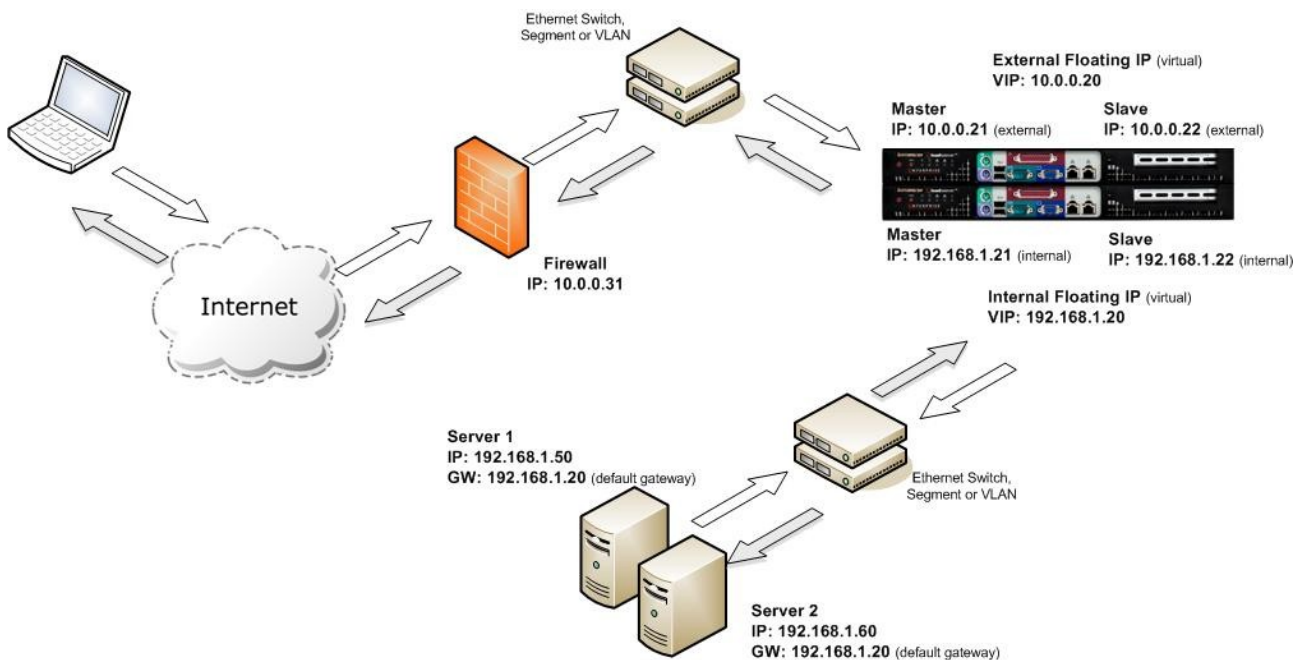
## High-availability configuration of two Loadbalancer.org appliances

When you have a pair of load balancers in a high-availability configuration they must communicate via a heartbeat to ensure that the master node stays active. Should the master node suffer a failure, the slave will immediately take over any resources hosted on the shared floating IP address.

A two-arm configuration requires two floating IP(s):

- One for the external virtual server
- One for the internal real server default gateway

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



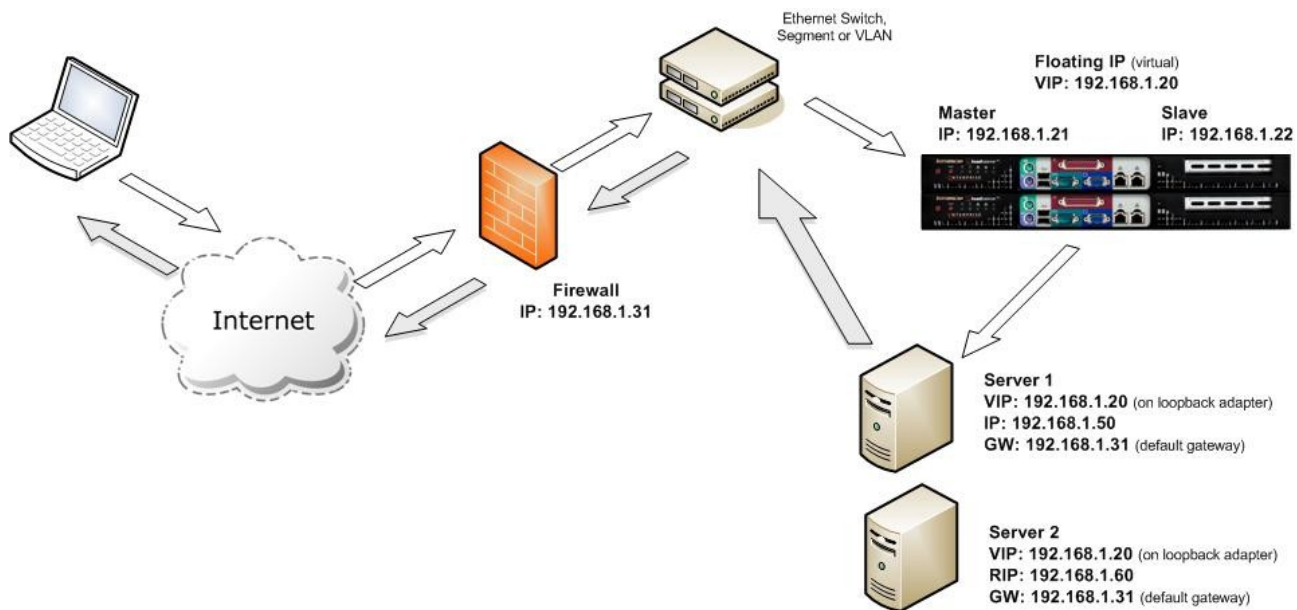
### Notes:

- Administration of the load balancers is via any active IP address
- The internal floating IP must be configured for the real servers to use as a default gateway
- The external floating IP must be configured for hosting the virtual server (public access)
- Using the setup wizard will prompt for both the internal and external floating IP
- Don't forget to plug the serial fail-over cable into each serial port



When using a clustered pair of load balancers in two-arm NAT mode all load balanced services must be configured on an external floating IP. The real servers must also have their default gateway directed to an internal floating IP.

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



#### Notes:

- Administration of the load balancers is via any active IP address
- A floating IP must be configured for hosting the virtual server
- Don't forget to plug the serial fail-over cable in to each serial port
- Don't forget to handle the ARP issue on the real servers



When using a clustered pair of load balancers in one-arm DR mode all load balanced services must be configured on a floating IP.

## Unpacking 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*)

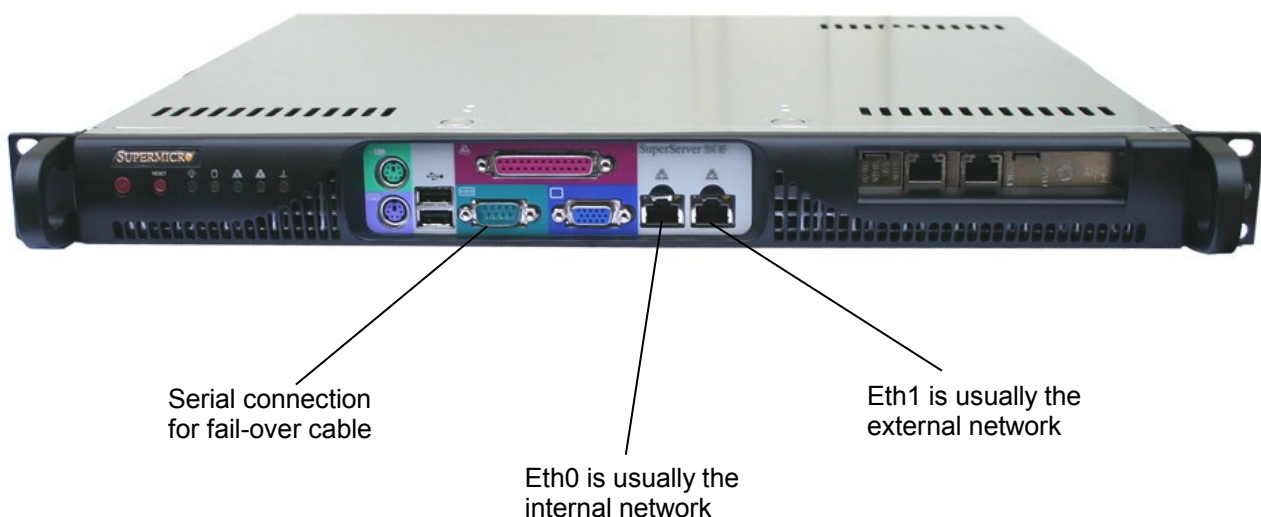


If two load balancers are being used, connect a null modem cable between the two serial fail-over ports and configure the slave first.

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.

### *Network interface configuration*

- Power up the load balancer
- Log in with:
  - Username: *root*
  - Password: *loadbalancer*

You can access the web interface (and subsequently the web based wizard) 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 i.e. ***http://192.168.2.21:9080/lbadmin/***

*(replace 192.168.2.21 with the correct address – this is displayed on the console at boot time)*

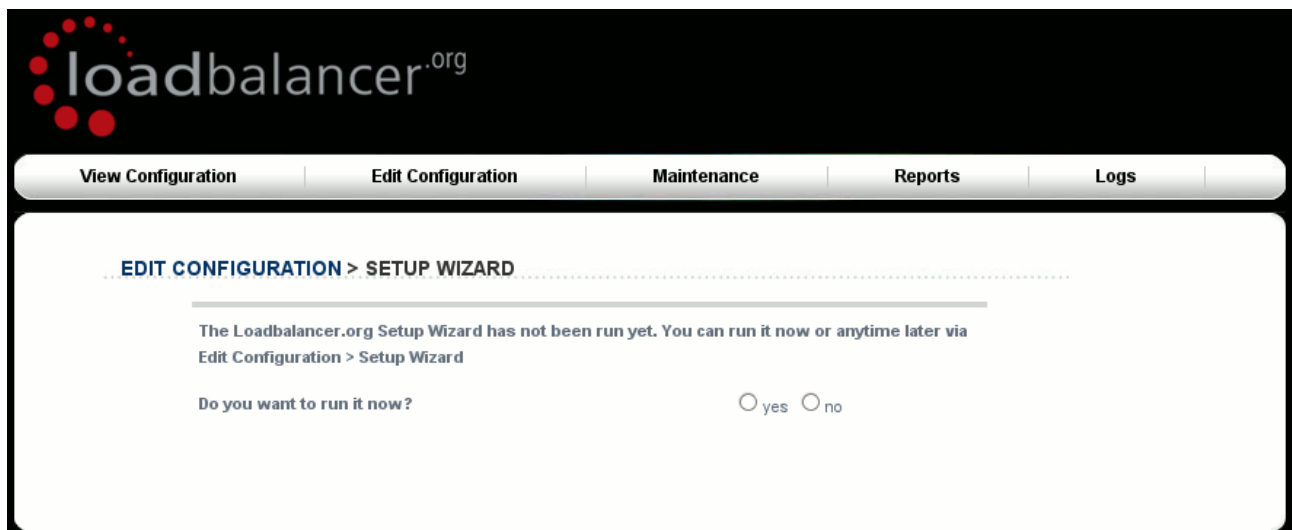
Log in to the web interface:

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.

## Configuring the Loadbalancer.org appliance using the console wizard

This section deals with the process of configuring a single load balancer appliance via the console. The console wizard enables you to configure a complete working configuration with one virtual server and one real server. You can then use the web interface to make modifications to this basic configuration.

*NB. For full configuration using only the web interface please see the administration manual*

### *Network interface configuration*

- Power up the load balancer
- Log in with:
  - Username: *root*
  - Password: *loadbalancer*
- Type the following command:
  - *lbwizard*

This activates the console based configuration wizard which will ask a series of questions in order to get you started quickly.

```
[root@lbmaster ~]# lbwizard

*****
Welcome to the Loadbalancer.org Setup Wizard. In combination with your
Quickstart guide, this Wizard will help you initialize your load balancer.

You can exit the Wizard at any time by pressing Ctrl-C, and all changes
will be undone. Values within square brackets are defaults and will be used
where you have provided no alternative input.
*****

Press Enter to continue, or Ctrl-C to exit.[]
```

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

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

*Question:* Is this unit part of an HA-Pair? (y/n)

*Answer:* n (We are just using one load balancer)

*Question:* Will the load balancer(s) form part of a one-armed set-up (i.e. same subnet as servers) ? (y/n)

*Answer:* n (We are going to use two-arm NAT mode)

*Question:* Enter the IP Address for the INTERNAL interface eth0 [192.168.2.21]:

*Answer:* 10.0.0.21 (This is for the internal subnet)

*Question:* Enter the netmask for interface eth0 [255.255.255.0]:

*Answer:* 255.255.255.0

*Question:* Enter the IP address for the EXTERNAL interface eth1 [10.0.0.21]:

*Answer:* 192.168.2.21 (This is for the external subnet)

*Question:* Enter the netmask for interface eth1 [255.255.255.0]:

*Answer:* 255.255.255.0

*Question:* Enter the IP address of the default gateway [10.0.0.1]:

*Answer:* 10.0.0.31 (The default gateway for your external network)

*Question:* Enter the IP address of the nameserver [192.168.2.1]:

*Answer:* 10.0.0.31 (The DNS server for your external network)

*Question:* Enter the port for first Virtual Server [80]:

*Answer:* 80 (The first virtual server will default to the external address i.e. 10.0.0.21:80)

*Question:* Enter the IP address of the first real server (back-end) [10.0.0.100]:

*Answer:* 192.168.2.50 (The real server must be on the internal subnet)

*Question:* To confirm these settings and make the changes permanent, press Enter. To cancel changes and exit, press Ctrl-C.

*Answer:* <Enter> or Ctrl-C.

Once the wizard is complete the load balancer is configured correctly.

Now you need to configure the real server to ensure that it uses the internal IP address of the load balancer as its default gateway. Once this is done you can test the virtual server from the external network.

You can also use the web interface to easily add more virtual or real servers to your configuration.



If you need to restore the manufacturer's settings at any time - just use the command `lbrestore` from the console.

---

## 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 i.e. ***http://192.168.2.21:9080/lbadmin/***

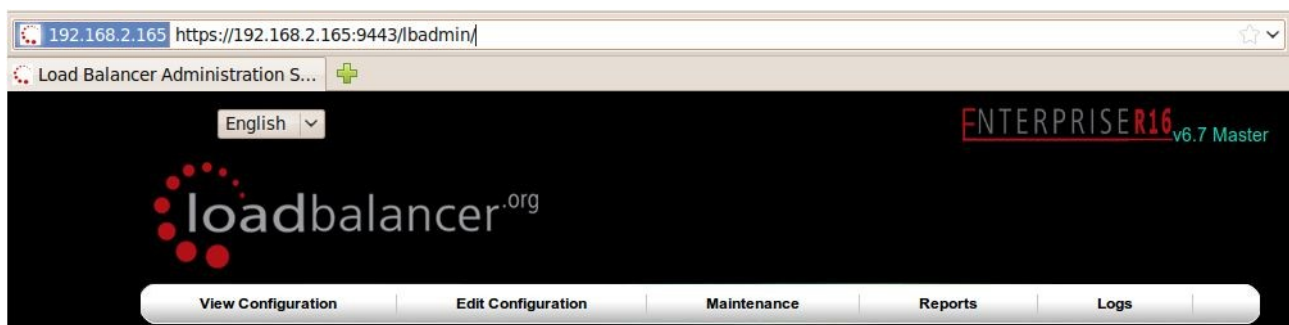
*(replace 192.168.2.21 with the correct address)*

Log in to the web interface:

Username: **loadbalancer**

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**

### EDIT CONFIGURATION > REAL SERVERS

VIP 1	<b>HTTP_Cluster</b>	(192.168.1.23:80)	[ <b>Add a new Real Server</b> ]
[ <b>Virtual Servers</b> ]			

- 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

### EDIT CONFIGURATION > ADD A NEW REAL SERVER

Label	<input type="text" value="WebServer1"/>	?
Real Server (ipaddress:port)	<input type="text" value="192.168.1.50:80"/>	?
Weight	<input type="text" value="1"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?
Forwarding Method	<input type="text" value="DR"/>	?
<input type="button" value="Update"/>		

- 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 (back-end) 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 (back-end) 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.

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)*



---

Failure to correctly configure the real servers to handle the ARP problem is the most common problem in DR configurations. Please refer to Section E for more details.

---

## Real server (back-end) 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 server must have the MS loopback adapter installed and configured
- The MS loopback adapter must be configured to deal with the ARP problem
- Each 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.

*NB. Services on Windows respond to all ports by default*



Failure to correctly configure the real servers to handle the ARP problem is the most common problem in DR configurations. Please refer to Section E for more details.



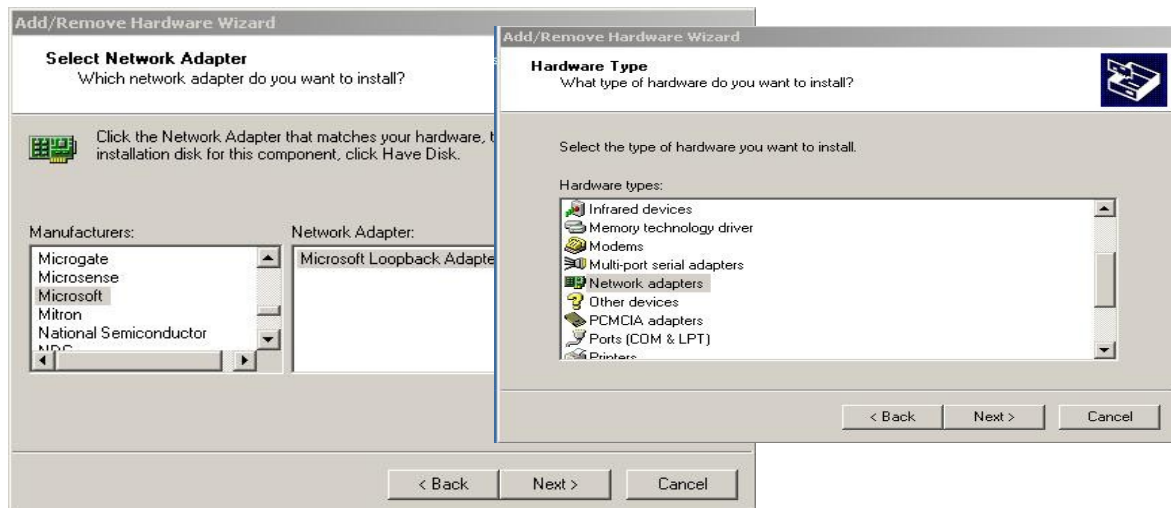
Remember that for all real servers in Direct Routing mode the load balanced application must respond to both the virtual IP as well as the servers real IP. With Windows the IP address must either be set to (All Unassigned) or use the Advanced tab to add a second IP address.

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

Windows server supports the direct routing method through the use of the MS loopback adapter to handle the traffic.

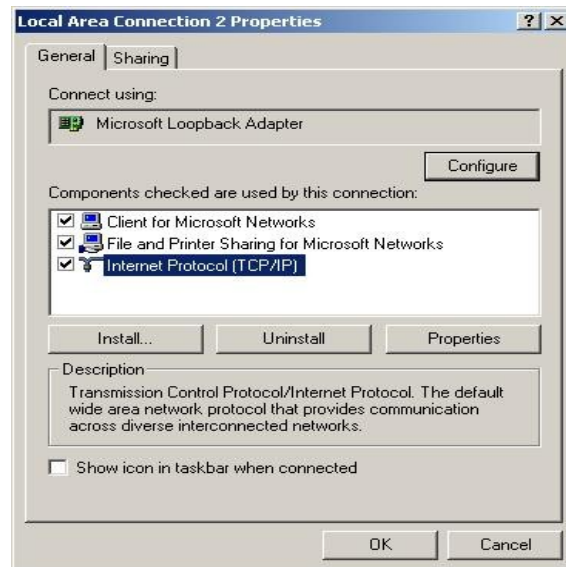
### **Installing the Microsoft loopback adapter**

1. Click Start, point to Settings, click Control Panel, and then double-click Add/Remove Hardware.
2. Click Add/Troubleshoot a device, and then click Next
3. Click Add a new device, and then click Next
4. Click No, I want to select the hardware from a list, and then click Next
5. Click Network adapters, and then click Next
6. In the Manufacturers box, click Microsoft
7. In the Network Adapter box, click Microsoft Loopback Adapter, and then click Next
8. Click Finish

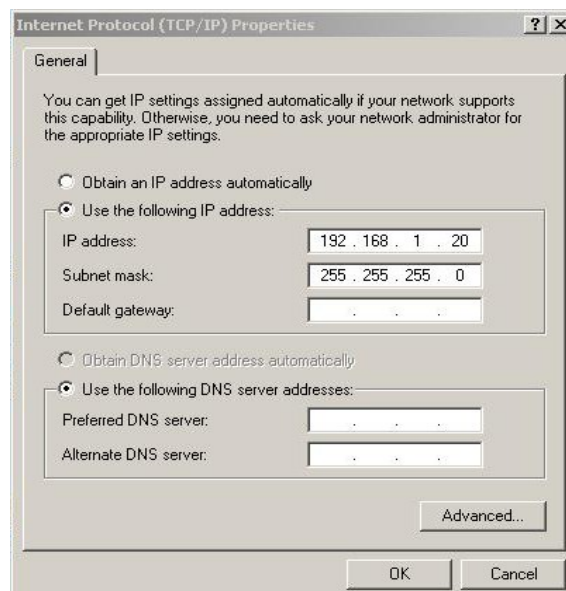


## Configuring the loopback adapter

1. Click Start, point to Settings, click Control Panel, and then double-click Network and Dial up Connections
2. Right click the new local adapter and select properties
3. Remove the tick from Client for Microsoft Networks
4. Remove the tick from File and Printer Sharing for Microsoft Networks



5. Select TCP/IP Properties



4. Fill in the virtual server IP address e.g. 192.168.1.20 and the subnet mask
5. Click on the *Advanced* button

6. Change the Interface Metric to 254 (This stops the adapter responding to ARP requests)
7. Click OK and save all changes



9. Repeat the above process for all of the web 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)

In Windows 2008 we have a whole new way of controlling networking. Microsoft finally have a sensible way of controlling network interfaces.

*NB. Please read the previous section relating to MS loopback adapter installation.*



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.

### **Weak and strong host behavior in Windows**

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.

- You still need to configure the loopback adapter with the VIP (but you don't need to set the metric)
- You still need to disable the firewall (or enable traffic to and from the loopback)
- Then you need to use the following command line magic:

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

First you will need to rename the specific adapters from the default of “Local Area Network Connection 1” to either “net” or “loopback” respectively i.e.



Or if you want look up the index number instead using the following command:

```
netsh interface ipv4 show interface
```



## 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.



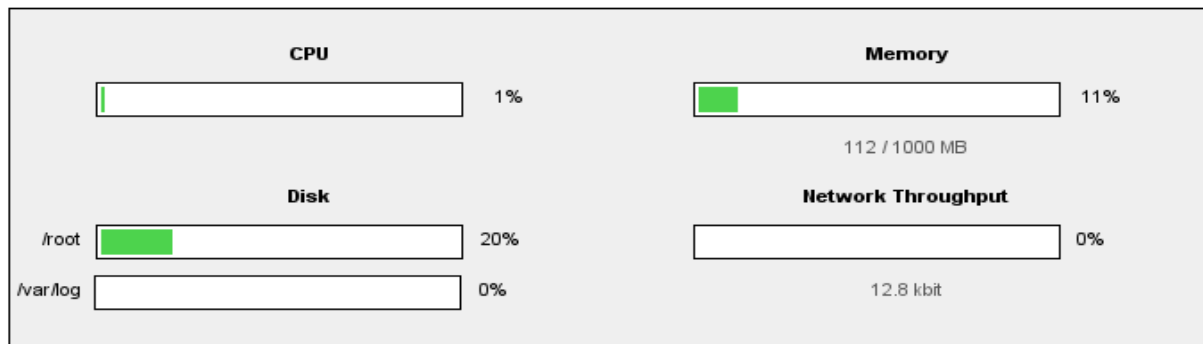
When using a two-arm NAT load balancing method the test client must be in the external subnet.

## 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.

### VIEW CONFIGURATION > SYSTEM OVERVIEW



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

+	HTTP_Cluster - 192.168.2.214:80 total connections:0
+	FTP_Cluster - 192.168.2.11:80 total connections:0
+	SMTP_Cluster - 192.168.2.1:80 total connections:0

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

## Health check diagnosis

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

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

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

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

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 make sure that you 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 it is not enabled by default.

## 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.

If this is the case you can use persistence by 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 technology 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).

*NB. Cookies can only be used in HTTP/HTTPS based applications. Refer to Section D for more details.*

## **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 with:

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

Access the web interface either via links at the console or from a web browser on a client connected to the same network. The default IP address is *192.168.2.21/24*.

Log in to the web interface at *http://192.168.2.21:9080* with:

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

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

- Specify the IP address, Netmask & Default Gateway
- Use *Edit Configuration > Network Interface Configuration*
- Now refer to later sections of this section 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.



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 the default gateway.



If you have not used the wizard then the web interface will default 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 must be configured first, followed by the master.

### *Network interface configuration*

- Connect the serial interface cable
- 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. The default IP address is *192.168.2.21/24*. If you haven't changed the administration (eth0) address, you can setup a client with an IP address in the same subnet, e.g. *192.168.2.11/24*. To change the IP address of eth0, at the console use:

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

- Log in to the web interface at *http://192.168.2.21:9080/lbadmin/* (or the new address if 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
- Make sure that the serial (Null modem cable) is attached between the master & slave load balancer for the heartbeat signal and also make sure they are both plugged into the same network switch before turning on the master load balancer
- Power up the master load balancer
- After the master has booted, log into *http://192.168.2.21:9080/lbadmin/* (you'll need to ensure that you have reconfigured the IP address of the slave unit first) as the user **loadbalancer** with the password of **loadbalancer**
- Use *Edit Configuration > Network Interface Configuration*
- Specify the IP address, Netmask & Default Gateway
- Use *Edit Configuration > DNS & Hostname*
- Specify the IP address of the master load balancer

- Any changes to the configuration of the master load balancer will now be automatically replicated to the slave
- Now refer to later sections of this section to configure the virtual & real servers

**i** When you are only configuring a cluster pair of load balancers you can use the IP address you configure for administration. But you must configure a Floating IP as well as for the VIP so that it can be shared between the master and slave load balancer.

**i** 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

Once you have configured the IP addresses for your chosen configuration you can then set up your load balanced services.

### *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.

The screenshot shows the LoadBalancer.org web interface. At the top, there is a language dropdown set to 'English' and a version indicator 'ENTERPRISE R16 v6.7 Master'. Below the header is a navigation bar with tabs: 'View Configuration', 'Edit Configuration', 'Maintenance', 'Reports', and 'Logs'. The main content area displays the breadcrumb 'EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER'. The form contains the following fields:
 

- 'Label' with the value 'VIP Name'.
- 'Virtual Server (ipaddress:port)' with the value '10.0.0.20:80'.
- 'Persistent' with a dropdown menu set to 'no'.
- An 'Update' button at the bottom.

- 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 some Real Servers (typically web servers) to the cluster

### *Real Server configuration (RIP) in one-arm DR mode*

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

The screenshot shows the LoadBalancer.org web interface. At the top, there is a language dropdown set to 'English' and a version indicator 'ENTERPRISE R16 v6.7 Master'. Below the header is a navigation bar with links: 'View Configuration', 'Edit Configuration', 'Maintenance', 'Reports', and 'Logs'. The main content area is titled 'EDIT CONFIGURATION > ADD A NEW REAL SERVER'. It contains a form with the following fields:

Label	<input type="text" value="RIP Name"/>	?
Real Server (ipaddress:port)	<input type="text" value="IPAddress:80"/>	?
Weight	<input type="text" value="1"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?
Forwarding Method	<input type="text" value="DR"/>	?

Below the form is an 'Update' button.

- 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

You have now finished the configuration of both load balancers for the cluster. Now you must configure the web servers to respond to the load balancers requests. As we are using a one-arm DR load balancing method each web server requires the ARP problem to be handled:

- 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
- Each 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



Failure to correctly configure the real servers to handle the ARP problem is the most common problem in DR configurations. Please refer to Section E for more details.

## Configuring the Virtual Servers (VIP) in two-arm NAT mode

Once you have configured the IP addresses for your chosen configuration you can then set up your load balanced services.

### *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

The screenshot shows the LoadBalancer.org web interface. At the top, there is a language dropdown set to 'English' and a version indicator 'ENTERPRISE R16 v6.7 Master'. Below the header is a navigation bar with tabs: 'View Configuration', 'Edit Configuration', 'Maintenance', 'Reports', and 'Logs'. The main content area displays the breadcrumb 'EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER'. Below this is a form with the following fields:
 

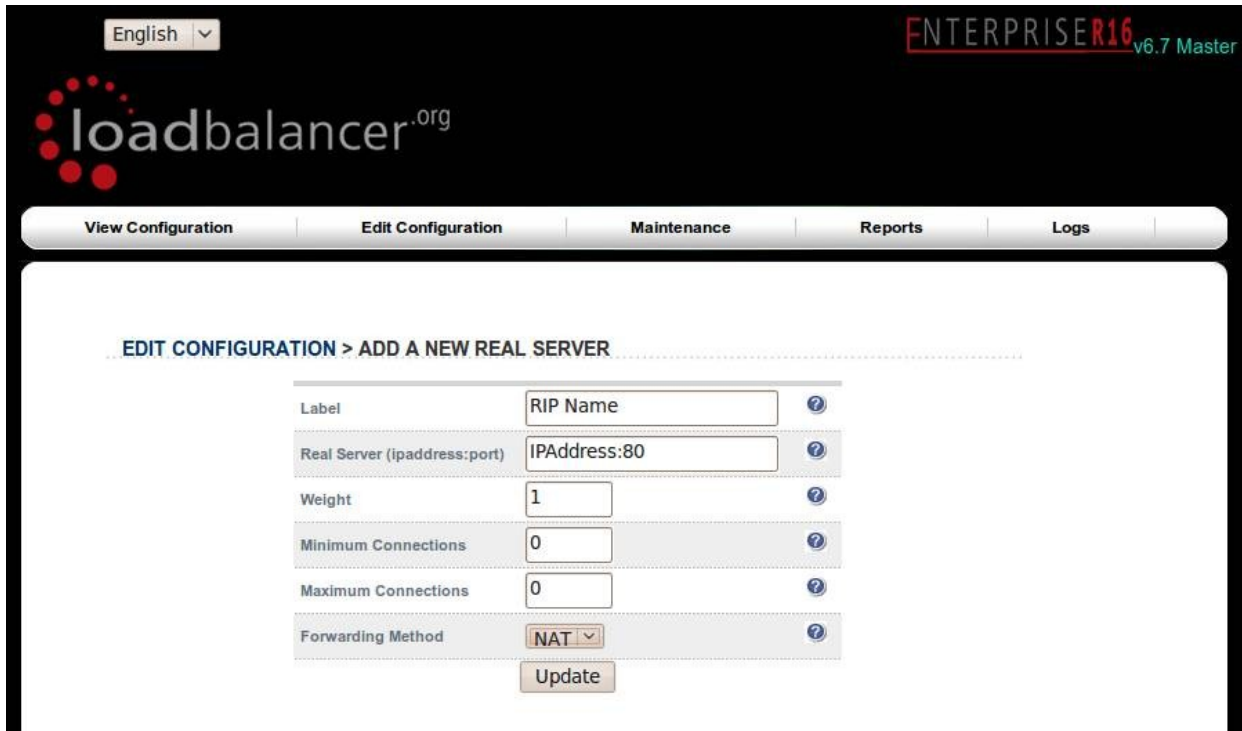
- 'Label' with the value 'VIP Name'.
- 'Virtual Server (ipaddress:port)' with the value '10.0.0.20:80'.
- 'Persistent' with a dropdown menu set to 'no'.
- An 'Update' button at the bottom of the form.

- 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 some Real Servers (web servers) to the cluster

## Real server configuration (RIP) in two-arm NAT mode

Each Virtual Server needs a cluster of real servers (backend 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



The screenshot shows the LoadBalancer.org web interface. At the top, there is a language dropdown set to 'English' and a version indicator 'ENTERPRISE R16 v6.7 Master'. Below the logo, a navigation bar contains links: 'View Configuration', 'Edit Configuration', 'Maintenance', 'Reports', and 'Logs'. The main content area is titled 'EDIT CONFIGURATION > ADD A NEW REAL SERVER'. It contains a form with the following fields: 'Label' (containing 'RIP Name'), 'Real Server (ipaddress:port)' (containing 'IPAddress:80'), 'Weight' (containing '1'), 'Minimum Connections' (containing '0'), 'Maximum Connections' (containing '0'), and 'Forwarding Method' (a dropdown menu set to 'NAT'). Each field has a help icon to its right. At the bottom of the form is an 'Update' button.

- 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.

You have now finished the configuration of both load balancers for the cluster. Now you must configure the web servers to respond to the load balancers requests.



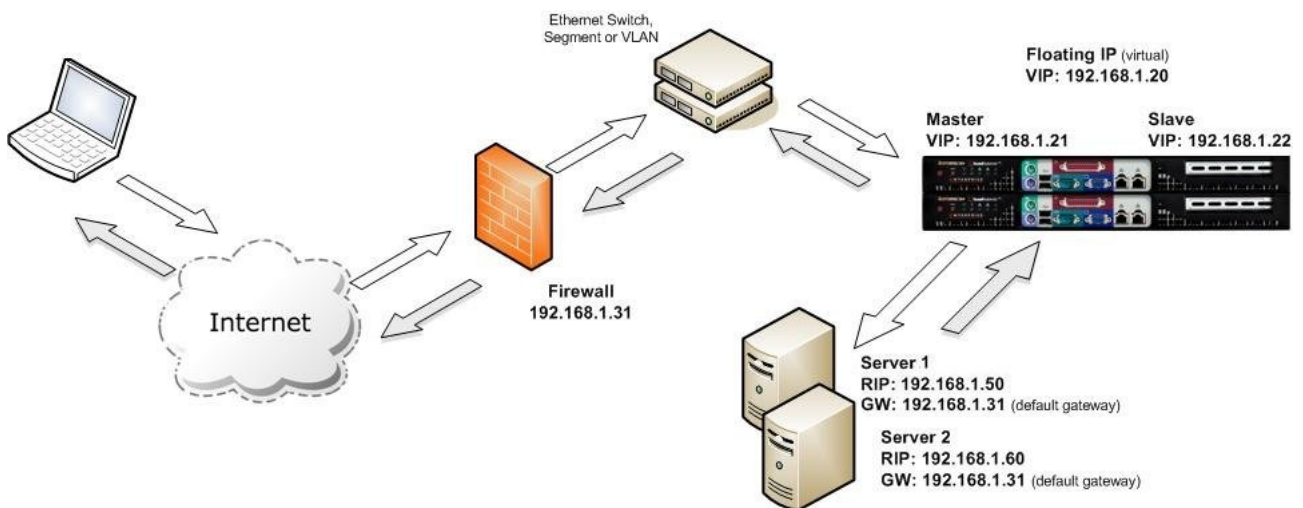
When using a two-arm NAT load balancing method each web server has to be in the same subnet as the internal load balancer and the default gateway must point at the load balancer.

### Example 3: layer 7 configuration one-arm SNAT mode

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 to 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.

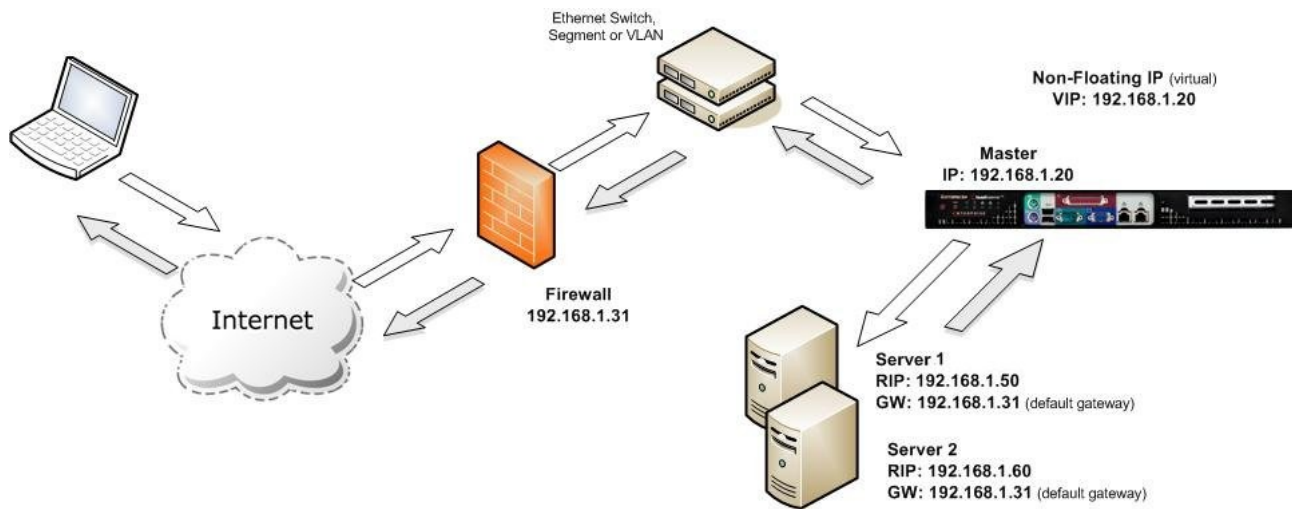
#### *Network diagram for layer 7 SNAT mode (clustered pair)*



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 E of this manual.*

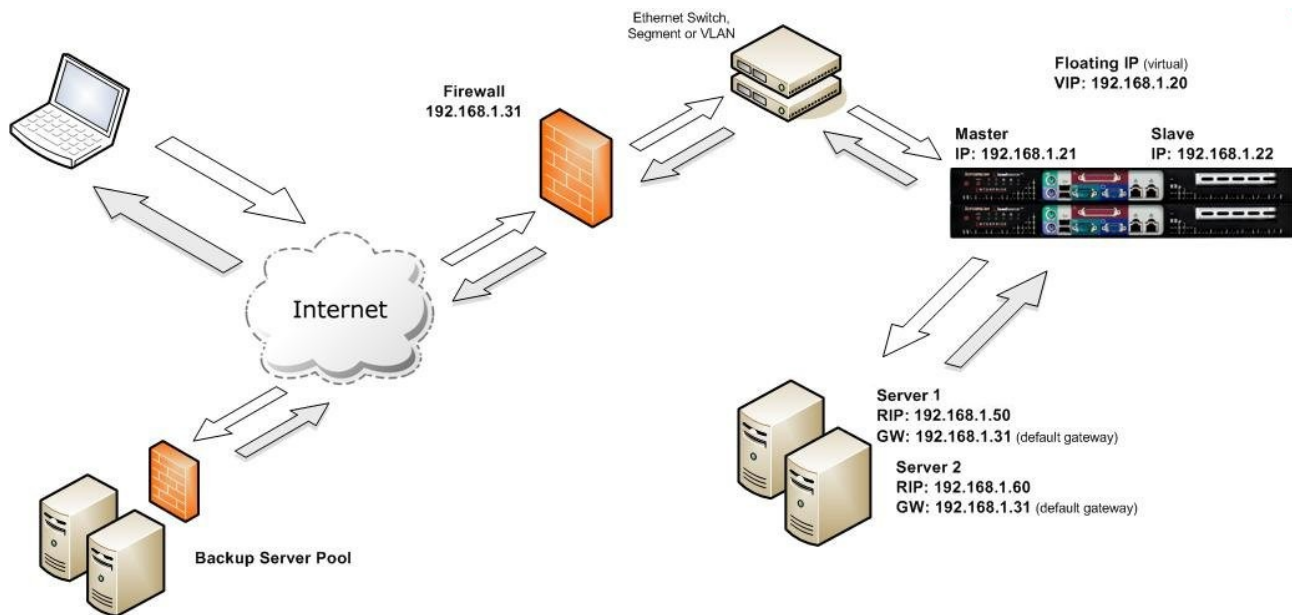
### Network diagram for layer 7 SNAT mode (single unit)



#### Notes:

- As with other modes a single unit does not require a Floating IP.
- 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 (layer 7 HAProxy)

- 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:portno*. 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.

Label	<input type="text" value="VIP Name"/>	?
Virtual Server (ipaddress:port)	<input type="text" value="10.0.0.20:80"/>	?
Persistence mode	<input type="text" value="cookies"/>	?
Fallback	<input type="text" value="127.0.0.1:80"/>	?

- 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 added as a Floating IP and activated.

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

## Real Server configuration (HAProxy)

- Use *Edit Configuration > Real Servers (HAProxy)* and you should see your Virtual Servers listed, select the one you want and click on **add real server**.
- You just need to give the *IPAddress:Port* of your web server and specify a relative weight.

**EDIT CONFIGURATION > ADD A NEW REAL SERVER (HAPROXY)**

Label	<input type="text" value="RIP Name"/>	?
Real Server (ipaddress:port)	<input type="text" value="IPAddress:"/>	?
Weight	<input type="text" value="1"/>	?

- 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 (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)

### EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER SSL TERMINATION (POUND)

Virtual Server (ipaddress:port)	<input type="text" value="10.0.0.20:443"/>	
Backend Cluster	<input type="text" value="10.0.0.20:80"/>	
Ciphers to use	<input type="text"/>	
<input type="button" value="Update"/>		

- 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.



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


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.


## 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 (CSR).

Goto Edit Configuration > SSL Termination (Pound), then click [Modify] next to the relevant Virtual Server, then click [Manage this SSL Certificate].

**EDIT CONFIGURATION > VIRTUAL SERVERS SSL TERMINATION (POUND)**

Virtual Server (ipaddress:port)	<input type="text" value="192.1681.20:443"/>	
Backend Cluster	<input type="text" value="192.1681.20:80"/>	
Ciphers to use	<input type="text"/>	



[Manage this SSL Certificate](#) 

The following default information is displayed. This can be changed as needed.

When you have entered your correct details the CSR is generated for you:

**NB. Make sure you backup, i.e. save to a text file both the CSR & the Private Key. Copy the Certificate Signing Request and provide it to your Certificate Authority.**

They in turn will then sign the Certificate which you should paste into the Signed Key field of the form and upload.

-----BEGIN CERTIFICATE REQUEST-----

```
MIIB7ZCAVYCAQAwgawwCzAJBgNVBAYTA1VTRV9EwYDVQQKEWhhZDZfY2ZET
MEGAAGUUEBjMKv2lSBWUz3r3vbjEBMBGAG1UECHMSQXZlc3RlcnlnYXNlCBjBMu
MRAwDgYDVQLLEWdTRXBvbgwYDVQQDESR3d3ZlxlcnlnYXNlLnNv
bWtENMCUGCSqGSIb3DQEJARYYczV3cG9ydeBjvhvZDGVyc2NhbgGJUY29tMfGMAOG
CSqGSIb3DQEBAQUAA4GNADCBiQKBGQCO0aorBf1Ua+BPsoJjV/cb7HihJgyuuMS
UPcQTBKly1zomj5eI20uY6eqhzh9QIeupTzHwT1JmIa4//AAI5v8yGPMhZTP1
3hskG+FlbXlSvBgHxd50c15UvoEL/oulKAHP4e1a1SxofN+HCvREh4vg003M
XjEY2cvw+IDAQABAAwDAQYJKoZIhvcNAQEEAQDQYEAKOZL1C0CgULyU+U8iND+
```

Paste your signed certificate here.

Upload Signed Certificate

-----BEGIN PRIVATE KEY-----

```
MIICXAIBAAKGBGQCO0aorBf1Ua+BPsoJjV/cb7HihJgyuuMSUPcQTBKly1zomj5e
I20uY6eqhzh9QIeupTzHwT1JmIa4//AAI5v8yGPMhZTP13hskG+FlbXlSvBgH
hxd50c15UvoEL/oulKAHP4e1a1SxofN+HCvREh4vg003MjEY2cvw+IDAQAB
AoGAgvze3v0qgZyZ720Rv+shpQnolZzNMJskshUBXJzJlYgK9KscP1Jb
OGJFRgnzL1q4JgmqfRrxhhYzV1TRFnrhP53RTMCSQCO9PMhKdz5eI20uZ3t
yUdhkroblYqyDilryw+ph06ZSg549tHh+dbUokc9bKcEQ9PMhKdz5eI20uZ3t
7h5eNaaSx5Z2mWu7nsRLKLVLxqraGfDeT7BXhQtaLxe/FaNNe63LULhK+nJ
w1YbqMKABAwVn2Q0FdbWq4nJ6f4da0LT40uzkXZxsmZGU+xxftieF/UUcVv
```

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.



If you need to convert, upload, modify or chain a certificate look at the [Advanced SSL Considerations](#) section.

## **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.

- At the login prompt login as *root*
- The default password is *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 easy configuration 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 temporary command:

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

*NB. This is just temporary, remember to make the change permanent by using the web interface from a client.*

## **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 the 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.

## 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 A - the Quick Start Guide, at the start of this manual.

### *Solving the ARP problem*

Each web server needs a loopback IP address to be configured as the VIP 192.168.1.20. 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.

#### **Solving for Linux (with arp\_ignore sysctl values)**

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:

```
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
```

If you don't wish to re-boot 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 startup 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**

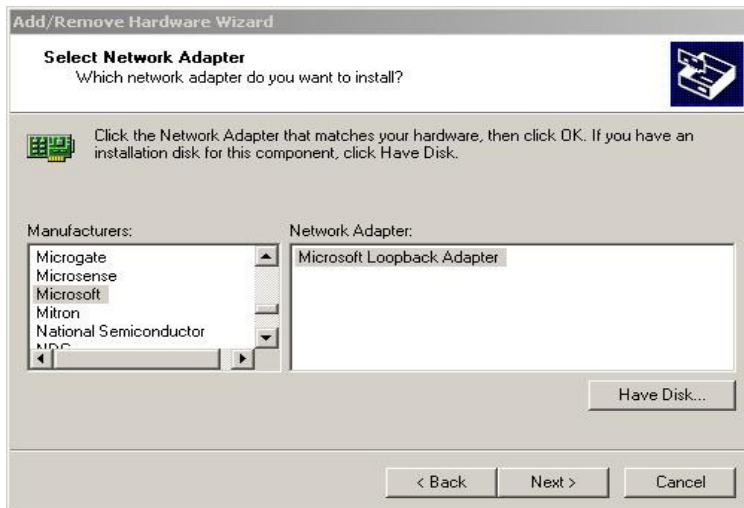
Windows server supports the direct routing method through the use of the MS loopback adapter to handle the traffic.

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

1. Click Start, point to Settings, click Control Panel, and then double-click Add/Remove Hardware.
2. Click Add/Troubleshoot a device, and then click Next
3. Click Add a new device, and then click Next
4. Click No, I want to select the hardware from a list, and then click Next
5. Click Network adapters, and then click Next



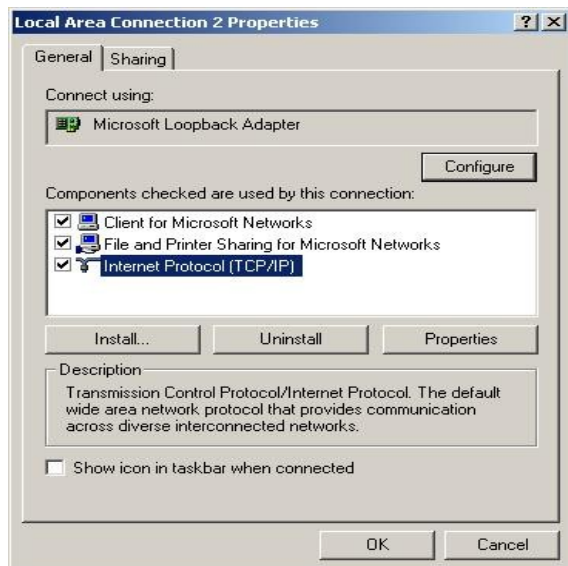
6. In the Manufacturers box, click Microsoft
7. In the Network Adapter box, click Microsoft Loopback Adapter, and then click Next



8. Click Finish

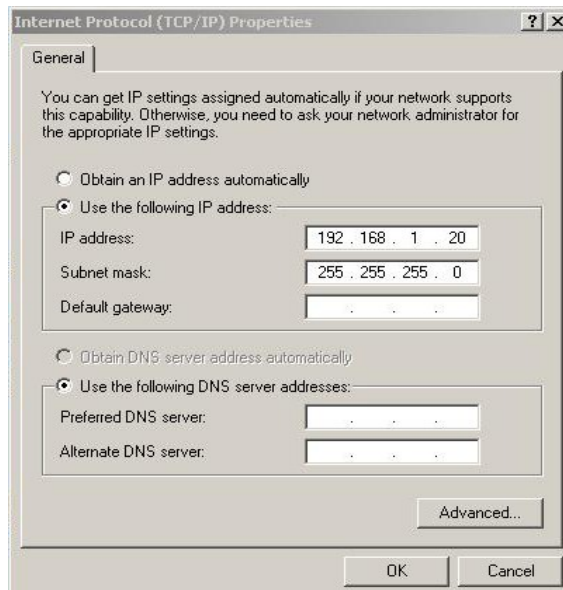
### Configuring the loopback adapter

1. Click Start, point to Settings, click Control Panel, and then double-click Network and Dial up Connections
2. Right click the new local adapter and select properties
3. Remove the tick from Client for Microsoft Networks
4. Remove the tick from File and Printer Sharing for Microsoft Networks

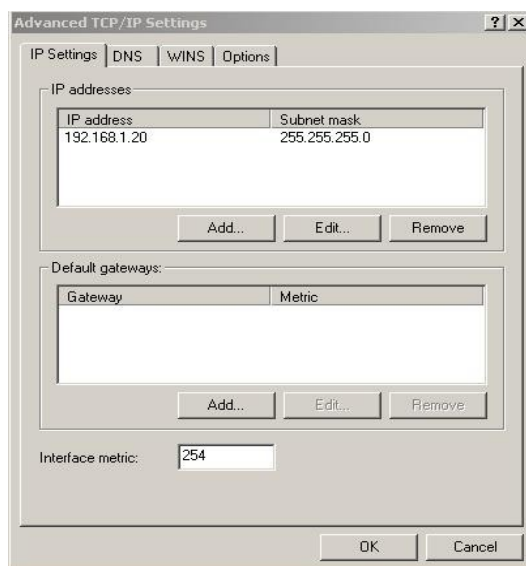


5. Select TCP/IP Properties





6. Fill in the virtual server IP address e.g. 192.168.1.20 and the subnet mask
7. Click on the *Advanced* button
8. Change the Interface Metric to 254 (This stops the adapter responding to ARP requests)

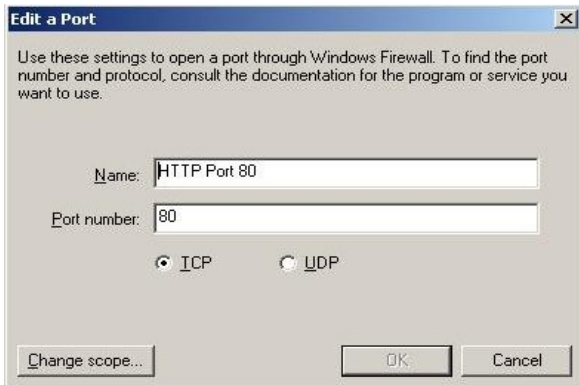


9. Click OK and save all changes
10. Repeat the above process for all of the web servers

## 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**

### Weak and strong host behavior in Windows

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.

- You still need to configure the loopback adapter with the VIP (but you don't need to set the metric)
- You still need to configure relevant firewall rules or disable the firewall
- Then you need to use the following command line magic:

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

First you will need to rename the specific adapters from the default of “Local Area Network Connection” and “Local Area Network Connection 1” to either “net” or “loopback” respectively i.e.



Or if you want look up the index number instead using the following command:

```
netsh interface ipv4 show interface
```

### Windows 2008 R2 Firewall Settings

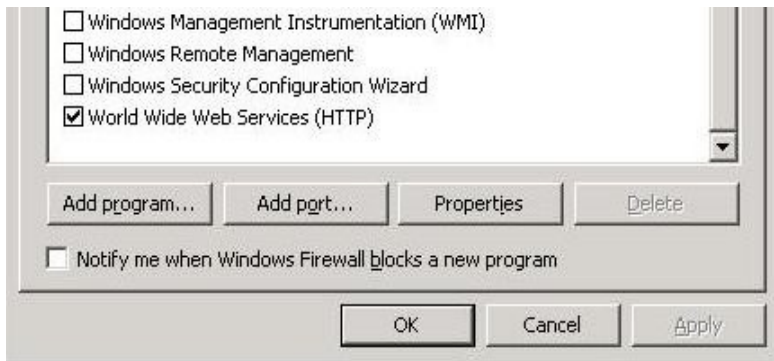
Windows 2008 automatically creates several default firewall rules for both inbound and outbound traffic. By default, all outbound traffic is allowed and all inbound traffic is blocked except where a rule allows it. Outbound rules can also be enabled if necessary. There are 3 firewall policies and interfaces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

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



## Windows 2008 R1 Firewall Settings

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



## 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

The setup wizard will automatically enable the `autonat` feature, 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 the outgoing interface 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 autonnat script rc.nat. You should probably disable autonnat 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 autonnat. 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 autonnat:

```
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](https://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:

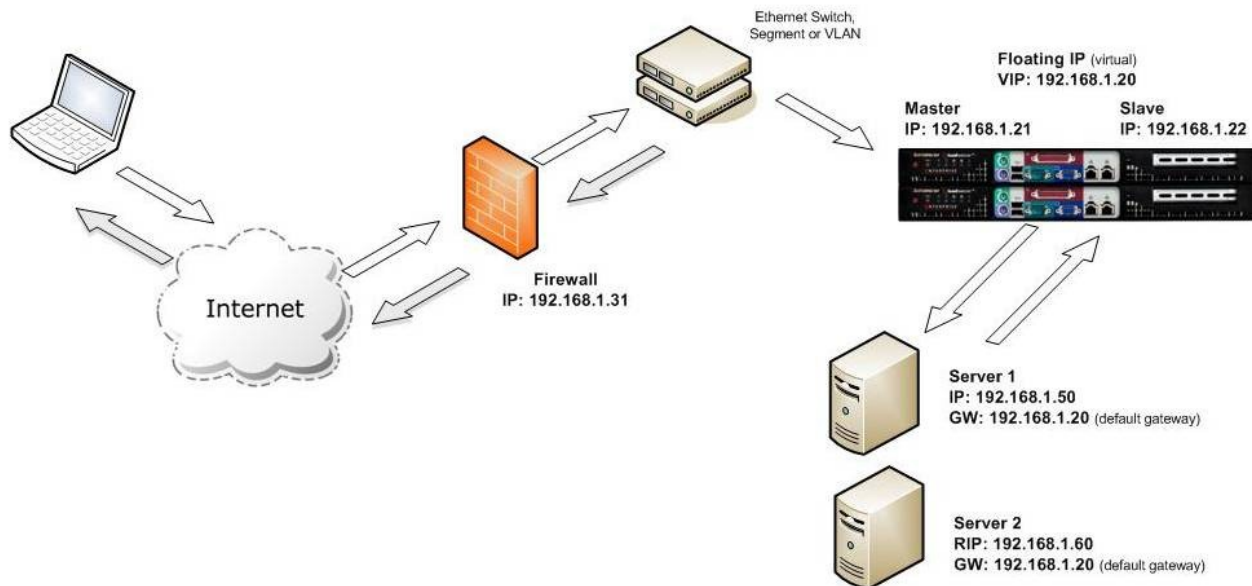
SOURCE                      192.168.1.50:80                      DEST                      x.x.x.x:3456

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.

In order for one arm NAT to work correctly you must modify the firewall script on the load balancers to disable ICMP redirects:

```
# For single NIC NAT you will also need to disable re-directs
# director is gw for realservers so turn OFF icmp redirects (1 on, 0 off)
echo "0" >/proc/sys/net/ipv4/conf/all/send_redirects
echo "0" >/proc/sys/net/ipv4/conf/default/send_redirects
echo "0" >/proc/sys/net/ipv4/conf/eth0/send_redirects
echo "0" >/proc/sys/net/ipv4/conf/eth1/send_redirects
echo "0" >/proc/sys/net/ipv4/conf/eth2/send_redirects
```

Make sure that these lines are active by removing the # at the start of each echo command.



## *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).


## 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*.

### EDIT CONFIGURATION > NETWORK INTERFACE CONFIGURATION

IP Address (eth0) :	<input type="text" value="192.168.3.80"/>	IP Address (eth1) :	<input type="text" value="192.168.2.80"/>
Netmask :	<input type="text" value="255.255.255.0"/>	Netmask :	<input type="text" value="255.255.255.0"/>
Default Gateway :	<input type="text" value="192.168.2.1"/>		
Bond eth0+eth1 (bond0) :	<input type="checkbox"/> 		

Update

- Aliases 

Interface	IP	Netmask	
<input type="text" value="eth0"/>	<input type="text"/>	<input type="text"/>	[ save ] [ cancel ]

- VLANs 

Interface	VLAN Number	IP	Netmask	
<input type="text" value="eth0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	[ save ] [ cancel ]

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.



*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.*

## Heath Monitoring









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

### *Loadbalancer Health*

When a clustered pair is deployed rather than a single appliance (strongly recommended by Loadbalancer.org), the loadbalancers 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 active) scenarios can be avoided.

### Heartbeat Configuration

**EDIT CONFIGURATION > MODIFY HEARTBEAT CONFIGURATION**

Serial	<input type="text" value="ttyS0"/>	
Bcast	<input type="text" value="none"/>	
Ucast	<input type="text" value="none"/>	
Keepalive	<input type="text" value="3"/>	
Deadtime	<input type="text" value="10"/>	
Warntime	<input type="text" value="5"/>	
Ping node	<input type="text" value=""/>	
Auto_fallback	<input type="text" value="on"/>	

### Serial Cable

This method requires a null modem cable (supplied) to be connected between the two loadbalancers in the cluster. This enables heartbeat checks to utilise 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 node can access. If one node loses access to the ping node then a failover will occur. However if both nodes lose access nothing will change.

## *Real Server Health*

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

### **Configuration**

Check Type	<input type="text" value="connect"/>	
Service to check	<input type="text" value="http"/>	
Check Port	<input type="text"/>	
Check Command	<input type="text"/>	
Virtual Host	<input type="text"/>	
Login	<input type="text"/>	
Password	<input type="text"/>	
Protocol	<input type="text" value="tcp"/>	
Granularity	<input type="text" value="255.255.255.255"/>	
File to check	<input type="text" value="check.txt"/>	
Response expected	<input type="text" value="OK"/>	
Email Alerts	<input type="text"/>	

## 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)



---

Don't hesitate to contact Loadbalancer.org support to discuss any specific requirements you may have. Further firewall examples and a description of the firewall wizard functionality are described in the advanced section of the administration manual.

---

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

# This example marks HTTP & HTTPS connections only

VIP1="192.168.0.66"

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

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

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.

## 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 its 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.

---

### *Limiting Passive Ports*

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

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

This would only allow ports 50000-50100.

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

### **For Linux**

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

pasv\_max\_port - max is 65535  
pasv\_min\_port - min is 1024

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

PassivePorts 50000 - 50100

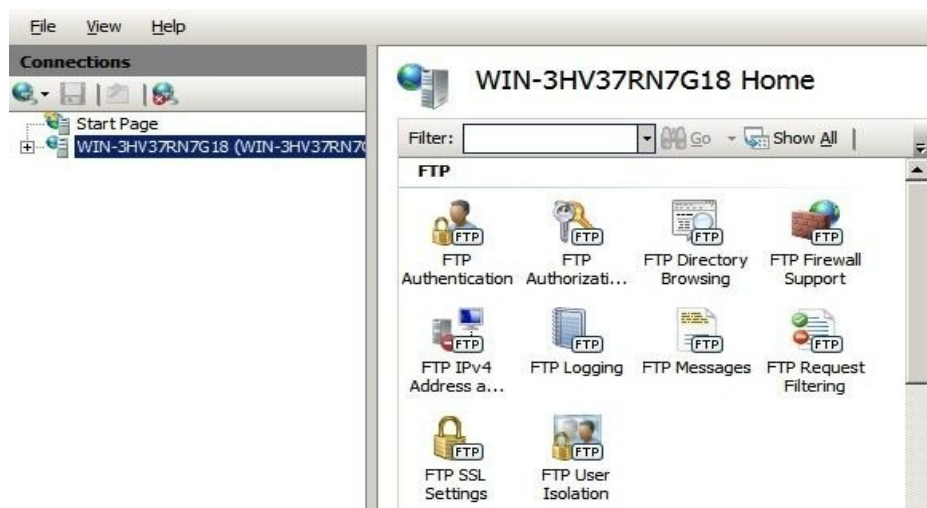
in pureftpd, the following startup switch can be used:

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

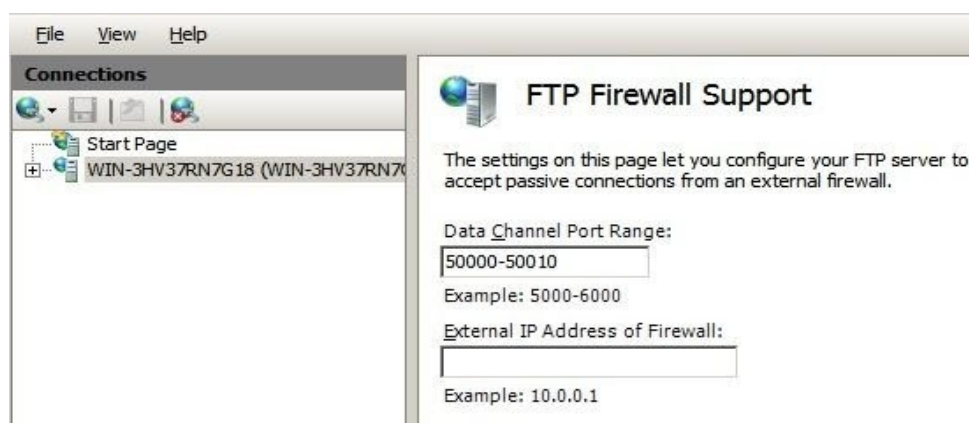
N.B. The firewall should also be configured to limit the ports to the same ranges.

## 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



## FTP Negotiate Health Check

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

Check Type	negotiate	?
Service to check	fto	?
Check Port		?
Check Command		?
Virtual Host		?
Login	username	?
Password	password	?
Protocol	tcp	?
Granularity	255.255.255.255	?
File to check	check.txt	?
Response expected	OK	?

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

## FTP Recommended persistence settings

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

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

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

## 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.

*NB. From v6.7 this timeout out is created automatically in /etc/rc.d/rc.lvstimeout*

### *Server maintenance when using persistence*

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

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

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

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

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

## *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.*

## 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.

Label	<input type="text" value="RDP Cluster"/>	?
Virtual Server (ipaddress:port)	<input type="text" value="192.168.2.32:3389"/>	?
Extra Ports	<input type="text"/>	?
Persistence mode	<input type="text" value="RDP cookie"/>	?
Fallback	<input type="text" value="192.168.2.99:3389"/>	?

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.

## Load balancing based on URL match with Pound

If you need to intercept the requested URL at Layer 7 and load balance based on a URL hash then you will need to use reverse proxy Pound as the load balancing agent. The pound configuration file is stored here:

```
/usr/local/etc/pound.cfg
```

You can use WINSCP to remotely access and edit this file as required.

*NB. Once you have edited this file by hand do NOT USE the web interface to set up SSL certificates or SSL termination.*

The following example shows how you can split up a cluster based on URL matching:

```
ListenHTTP 123.123.123.123,80

    # Images server(s)
    UrlGroup ".*.(jpg|gif)"
    BackEnd 192.168.0.8,80,1
    EndGroup

    # Send all requests for /myurlmatch to one back end server
    UrlGroup "/myurlmatch.*)"
    BackEnd 192.168.0.9,80,1
    EndGroup

    # Catch-all server(s)
    UrlGroup ".*"
    BackEnd 192.168.0.10,80,1
    BackEnd 192.168.0.11,80,1
    EndGroup
```

*NB. You will need to make sure this file is copied to both the Master and the Slave load balancer if you have a clustered pair.*

## 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, which seems to be similar to your scenario:

```
# 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're now have separate frontend and backend sections.

The 'acl' option provides the functionality that you're after; 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 detail see: <http://haproxy.1wt.eu/download/1.3/doc/configuration.txt> (do a search for 'path\_beg' on that page). Unfortunately all the HAProxy documentation is bundled together in massive text files like this, so it's not the easiest to decipher.

## 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.

## 8021q VLAN support

Native 8021qVLAN support can be enabled to load balance clusters on multiple VLANs. To configure "eth0.2", write a "/etc/sysconfig/network-scripts/ifcfg-eth0.2" file with:

```
VLAN=yes
DEVICE=eth0.2
BOOTPROTO=static
ONBOOT=yes
TYPE=Ethernet
IPADDR=192.168.1.21
NETMASK=255.255.255.0
```

Then create a Floating IP of say 192.168.1.31.

For this example *ifconfig* now shows:

```
eth0.2 Link encap:Ethernet HWaddr 00:40:63:D9:7D:28
inet addr:192.168.1.21 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b) TX bytes:460 (460.0 b)
eth0.2:0 Link encap:Ethernet HWaddr 00:40:63:D9:7D:28
inet addr:192.168.1.31 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500
```

*NB. From v6.7 VLAN tags can be added to aliases via the web interface.*



## 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
```

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)  
...
```

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"  
...
```

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"  
...
```

## Feedback agents

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

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

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

## *Installing the Windows agent*

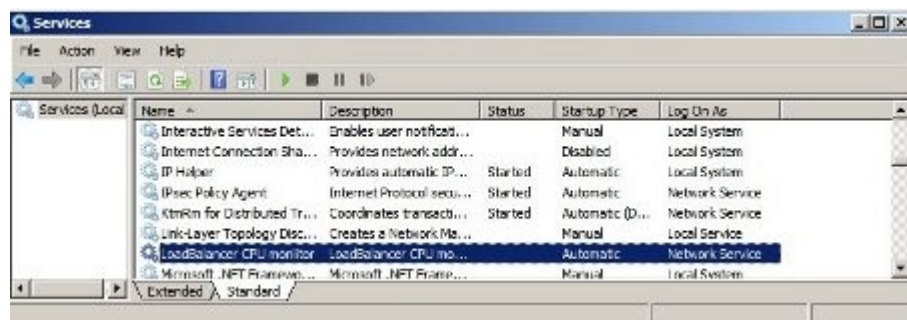
Download the agent from:

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

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



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



## *Installing the Linux/Unix agent*

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

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

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

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

/etc/init.d/xinetd restart

Testing:
telnet 127.0.0.1 3333

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

## *Custom HTTP agent*

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

## Changing the local date, time & time zone

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

*Logs > Change the date/time settings*

To manually set the date and time 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.*

## **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 xml configuration file and keep it stored in a safe location off the loadbalancer. Ideally 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

Goto *Maintenance > Disaster Recovery*

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

Also for the firewall configuration:

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

### *Backing up to the loadbalancer*

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

Log in to the web interface:

**Username:** loadbalancer

**Password:** loadbalancer

Goto *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 any Loadbalancer.org appliance to v6.7 via any USB stick (1GB or more in capacity).

**i** 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 develop and support v5.*

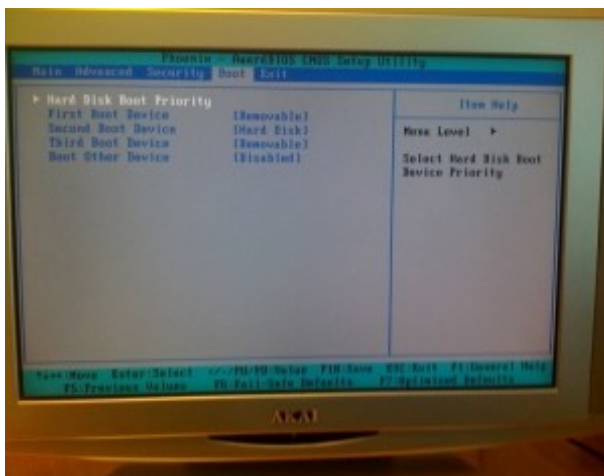
The v6.7 image requires a high speed 4Gb IDE DOM / Flash. If you are already running v6 then you will already have this and should be able to simply re-image your current IDE DOM / Flash. If you are upgrading from v5 you will need to purchase a 4Gb IDE DOM / flash card and then use the following procedure to build it from the USB stick.

**i** 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 v6.5 appliance once the following steps have been completed.

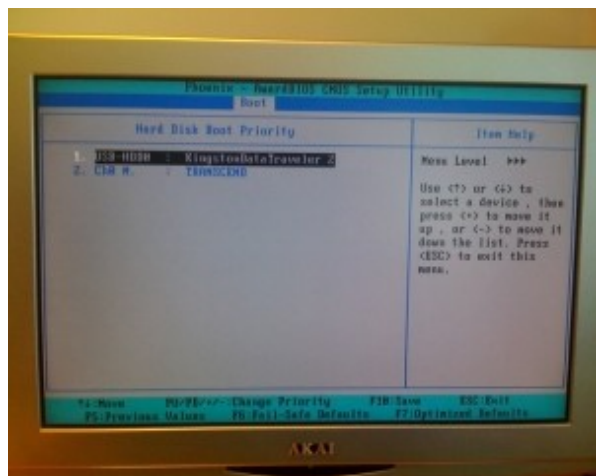
The USB stick bootable version of the ISO file is available at <http://www.loadbalancer.org/download/v6.7a.iso>

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 (stick must be plugged in at that stage).



First boot device (Removable)



Hard disk boot priority (USB)

When it boots choose:

Default image

Then at the command line type:

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

at the first prompt, select option 1  
at the second prompt, select option 1  
then Yes to all other prompts

The image transfers onto any IDE HD or IDE DOM / Flash module.

**\*\*\*\* You now have a fully functioning v6.6 appliance \*\*\*\***

*NB: If you need to upload a previous configuration go to **Maintenance > Disaster Recovery** in the web interface to upload your backup XML file.*

## Disaster Recovery after Master Failure

- Locate your backup copy of the XML configuration file (*lb\_config.xml*)

*NB: It is possible to recover the master if you do not have the *lb\_config.xml* from the master. The simplest way is take a copy of the *lb\_config.xml* from */etc/loadbalancer.org* on the slave machine and edit the following lines:*

**Change:**

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

**To:**

```
<network>  
  <hostname>lbmaster</hostname>  
  <slave>192.168.2.165</slave> (specify IP of your slave unit)  
  <fullsync></fullsync>
```

**Change:**

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

**To:**

```
<eth0>  
  <ip>192.168.2.166</ip> (specify IP of your master unit)  
  <netmask>255.255.255.0</netmask>
```

- If the 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 Loadbalancer image using a USB stick by following the steps on the previous page



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

- Log onto the master appliance as:

**Username:** loadbalancer

**Password:** loadbalancer

- At the terminal stop the heartbeat service with the following command  
"service heartbeat stop"
- Copy your backup lb\_config.xml 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"
- To access the xml file (normally the USB device is named sda1) enter the following commands  
"mount /dev/sda1 /mnt"  
"cd /mnt"  
"ls" to list the files on the USB stick
- Copy the xml config file to the loadbalancer  
"cp lb\_config.xml /etc/loadbalancer.org"  
"umount /mnt"
- Shutdown the master unit using the following command  
"shutdown -h now"
- Reconnect the Heartbeat (Serial) cable and the network cable
- Power-on the master unit

Wait a few minutes and your cluster should be restored with the master unit as the active appliance and the slave as the passive appliance.

To confirm this on the web interface click "Reports" then "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.

## 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.

Again, the easiest way is if you have a backup of the XML configuration file (lb\_config.xml) from the Slave. If you haven't backed that up as well, you might like to do it now.

If the Slave has failed and you do not have a backup of its config, below is a procedure for recovering from the Master.

1. Download the Master's config, using *Maintenance > Disaster Recovery > Download XML configuration file*.
2. In a text editor, make the following changes to the lb\_config.xml :
  1. In the <physical> <network> section,
    1. Change <hostname> to lbslave.
    2. Remove any IP address in the <slave> tag.
    3. Change <fullsync> to off.
  2. In the <physical> <rip> section, change the IP addresses for eth0 and eth1 as necessary.
3. Save your modifications to lb\_config.xml
4. If the Slave loadbalancer is still connected to the network and the serial link, disconnect it.
5. Fix the slave server.
6. With the slave still disconnected from the network and serial link, restore the Loadbalancer image, using the instructions at <http://blog.loadbalancer.org/how-to-recover-your-load-balancers-to-v65-via-usb-stick/>
7. When the install is complete, log in to the slave with username root and password loadbalancer.
8. At the root prompt, run service heartbeat stop to stop the heartbeat service.
9. Transfer the master's config and recovery script, from <http://www.loadbalancer.org/download/recoveryscripts/>, to the new slave on a USB key.
10. Plug the USB key into the slave. Run the command fdisk -l to discover the device that the system has allocated to the USB key. It will usually be /dev/sdb1
11. Mount the USB key using mount <device> /mnt, and change to the directory with the config and recovery script.
12. Start the recovery by running php lbrecoverv66. When the prompt returns, wait a couple of minutes then shut down the slave server with shutdown -h 0
13. Reconnect the slave to the network and the serial link, and restart the server.
14. When the slave has finished booting, log on the master's web interface and restart the heartbeat service using *Maintenance > Restart Heartbeat*.

**\*\*\*\* Your cluster has now been restored \*\*\*\***

## **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 loadbalancer.

### *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

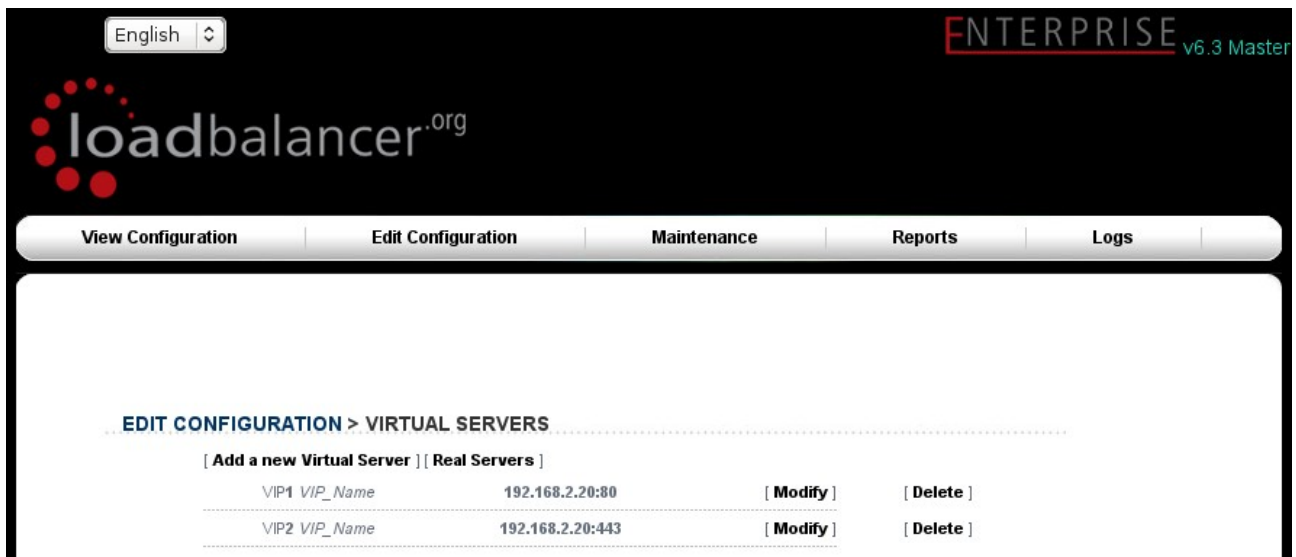
This menu option allows you to add, remove or modify virtual servers from your cluster.

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.*



The screenshot shows the web interface of the loadbalancer.org Enterprise v6.3 Master. The top navigation bar includes a language dropdown set to 'English', the 'loadbalancer.org' logo, and a menu with 'View Configuration', 'Edit Configuration', 'Maintenance', 'Reports', and 'Logs'. The 'Edit Configuration' menu item is active. Below the navigation bar, the page title is 'EDIT CONFIGURATION > VIRTUAL SERVERS'. There are two tabs: '[ Add a new Virtual Server ]' and '[ Real Servers ]'. The '[ Real Servers ]' tab is selected, displaying a table with two rows of virtual server configurations. Each row includes a 'VIP' (VIP1 and VIP2), a 'VIP\_Name', an IP address with port (192.168.2.20:80 and 192.168.2.20:443), and buttons for '[ Modify ]' and '[ Delete ]'.

VIP	VIP_Name	IP Address	Actions
VIP1	VIP_Name	192.168.2.20:80	[ Modify ] [ Delete ]
VIP2	VIP_Name	192.168.2.20:443	[ Modify ] [ Delete ]

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.

English

ENTERPRISE v6.3 Master

loadbalancer.org

View Configuration | Edit Configuration | Maintenance | Reports | Logs

EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER

Label

VIP Name

Virtual Server (ipaddress:port)

10.0.0.20:80

Persistent

yes

Update

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.

## EDIT CONFIGURATION > VIRTUAL SERVERS

Label	<input type="text" value="VIP_Name"/>	?
Virtual Server (ipaddress:port)	<input type="text" value="192.168.2.20:80"/>	?
Persistent	<input type="text" value="yes"/>	?
Persistence Timeout	<input type="text" value="300"/>	?
Scheduler	<input type="text" value="wrr"/>	?
Fallback Server	<input type="text" value="127.0.0.1:80"/>	?
Check Type	<input type="text" value="connect"/>	?
Service to check	<input type="text" value="http"/>	?
Check Port	<input type="text"/>	?
Check Command	<input type="text"/>	?
Virtual Host	<input type="text"/>	?
Login	<input type="text"/>	?
Password	<input type="text"/>	?
Protocol	<input type="text" value="tcp"/>	?
Granularity	<input type="text" value="255.255.255.255"/>	?
File to check	<input type="text" value="check.txt"/>	?
Response expected	<input type="text" value="OK"/>	?
Email Alerts	<input type="text"/>	?
Forwarding Method	<input type="text" value="DR"/>	?
Feedback Method	<input type="text" value="none"/>	?
<input type="button" value="Update"/>		

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 DR only 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
  - 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).

English

ENTERPRISE v6.3 Master

loadbalancer.org

View Configuration | Edit Configuration | Maintenance | Reports | Logs

**EDIT CONFIGURATION > REAL SERVERS**

VIP 1	<b>VIP_Name</b>	(192.168.2.20:80)	[ Add a new Real Server ]	
RIP 2	<b>RIP_Name</b>	192.168.2.60:80	1	[ Modify ] [ Delete ]
RIP 1	<b>RIP_Name</b>	192.168.2.50:80	1	[ Modify ] [ Delete ]
VIP 2	<b>VIP_Name</b>	(192.168.2.20:443)	[ Add a new Real Server ]	

[ Virtual Servers ]

English
ENTERPRISE v6.3 Master
loadbalancer.org

View Configuration
Edit Configuration
Maintenance
Reports
Logs

EDIT CONFIGURATION > ADD A NEW REAL SERVER

Label	<input type="text" value="RIP Name"/>	?
Real Server (ipaddress:port)	<input type="text" value="IPAddress:80"/>	?
Weight	<input type="text" value="1"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?
Forwarding Method	<input type="text" value="DR"/>	?
<input type="button" value="Update"/>		

*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.

*NB. If you take a server offline from the maintenance page and then bring it back online, the weight will be set back to one, just click on the 1 in order to link through to the 'modify real server screen' and change the weight back to the desired amount.*

## 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.



The screenshot shows the LoadBalancer.org Enterprise v6.3 Master web interface. The top navigation bar includes a language dropdown set to 'English', the 'loadbalancer.org' logo, and a menu with 'View Configuration', 'Edit Configuration', 'Maintenance', 'Reports', and 'Logs'. The main content area is titled 'EDIT CONFIGURATION > VIRTUAL SERVERS (HAProxy)'. Below this title, there are two tabs: '[ Add a new Virtual Server ]' and '[ Real Servers ]'. The '[ Real Servers ]' tab is active, displaying a table with two rows of virtual server configurations. Each row includes a 'VIP' label, a 'VIP\_Name', an IP address with port, and 'Modify' and 'Delete' buttons.

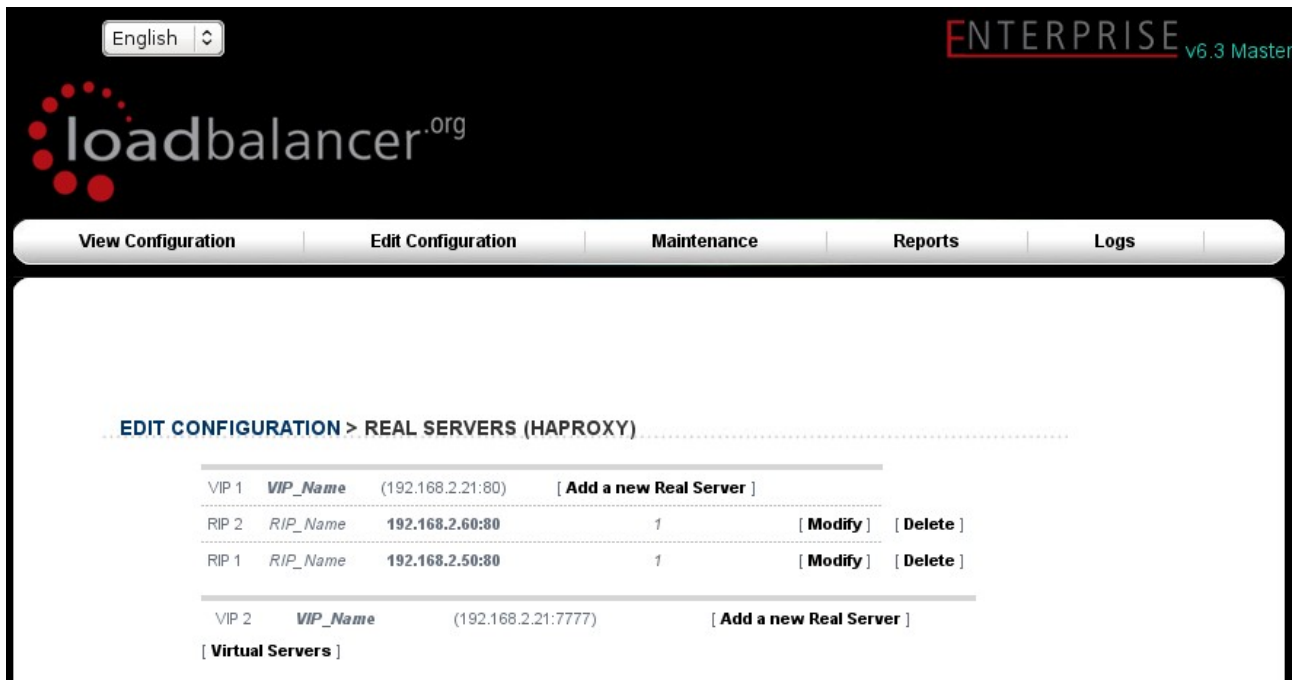
VIP	VIP_Name	IP:Port	Modify	Delete
VIP1	VIP_Name	192.168.2.21:80	[ Modify ]	[ Delete ]
VIP2	VIP_Name	192.168.2.21:7777	[ Modify ]	[ Delete ]

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.



English

ENTERPRISE v6.3 Master

loadbalancer.org

View Configuration | Edit Configuration | Maintenance | Reports | Logs

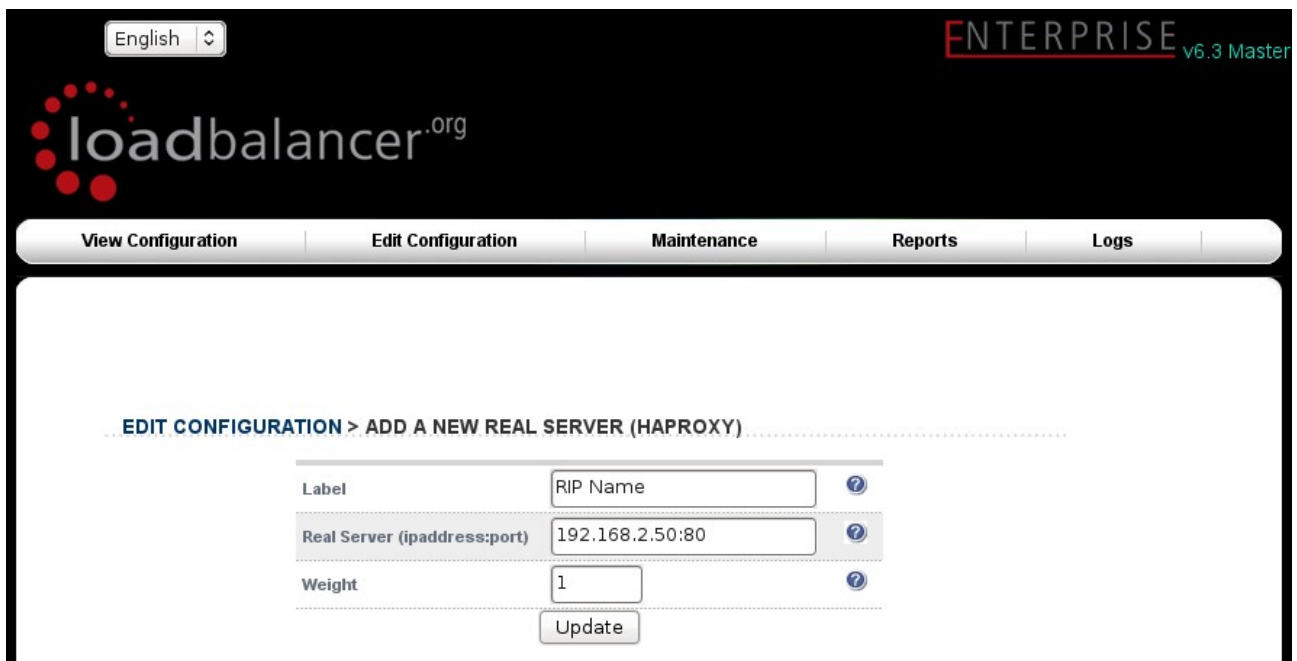
EDIT CONFIGURATION > REAL SERVERS (HAProxy)

VIP 1	VIP_Name	(192.168.2.21:80)	[ Add a new Real Server ]	
RIP 2	RIP_Name	192.168.2.50:80	1	[ Modify ] [ Delete ]
RIP 1	RIP_Name	192.168.2.50:80	1	[ Modify ] [ Delete ]

VIP 2 VIP\_Name (192.168.2.21:7777) [ Add a new Real Server ]

[ Virtual Servers ]

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.



English

ENTERPRISE v6.3 Master

loadbalancer.org

View Configuration | Edit Configuration | Maintenance | Reports | Logs

EDIT CONFIGURATION > ADD A NEW REAL SERVER (HAProxy)

Label	<input type="text" value="RIP Name"/>	?
Real Server (ipaddress:port)	<input type="text" value="192.168.2.50:80"/>	?
Weight	<input type="text" value="1"/>	?

Update

*NB. Any changes to the Layer 7 configuration require 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)*. 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

Virtual Server (ipaddress:port)	192.168.1.31:443
Backend	192.168.1.31:80
<button>Add a new Virtual Server</button>	

- Configure the Virtual Server as 192.168.1.31:443
- Configure the Back-end as 192.168.1.31:80
- Click the button to add the new Virtual Server to the Pound configuration file.
- **IMPORTANT:** You must restart the Pound service in order to activate the changes i.e. [Maintenance > Services > Restart Pound-SSL](#)

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

## Manage this SSL certificate

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

The screenshot shows the 'loadbalancer.org' interface with a navigation bar containing 'View Configuration', 'Edit Configuration', 'Maintenance', 'Reports', and 'Logs'. The 'Edit Configuration' tab is active, and the sub-section 'GENERATE SSL CERTIFICATE REQUEST' is selected. The form contains the following fields:

Country code (C)	US	?
State or Province (ST)	Delaware	?
City (L)	Wilmington	?
Organisation (O)	Loadbalancer.org, Inc.	?
Organisation unit (OU)	Support	?
Domain (CN)	www.loadbalancer.org	?
Email address	support@loadbalancer.org	?

Below the form is a button labeled 'Generate SSL Certificate Request (Server1)'.

When you have entered your correct details the CSR is generated for you:

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

Copy the Certificate Signing Request and provide it to your Certificate Authority. They in turn will then sign the Certificate which you should paste into the Signed Key field of the form and upload.

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

*Maintenance > Services > Restart Pound-SSL*



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

```
-----BEGIN CERTIFICATE-----
MIICsDCCAhmgAwIBAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxGzAJBgNV
BAYTAkFVMRMwEQYDVQQLIExpTb21ILVN0YXRIMSEwHwYDVQQKEzhJbnRlcm5ldCBX
kU6DJupvN6U6PRI7+zcKqd8wUiY8+3CyYKHtJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBKnyHYbQTvU8MeR3ilhe2fw+qVE2pgxWYWaGm8QwTsxQCkgbxiG
SXUWIWb0+k2j2L1z2PszFwClwQ=
-----END CERTIFICATE-----

-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwlm3HyoVxjrymusOeIFgZlWyuablrreCplo+iy
pSxEruhpmqmdj2tYpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPALafmh88bS7fh
rPCmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ/9Ec
SlqR7x/WAQUUnFKVxQAMDatpeXSp3FGgXF+mpffusjEw=
-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIIEwDCCBCmgAwIBAgIJY7GlcWfelAdoGNs+XVGezANBgkqhkiG9w0BAQUFADCB
Ai9TXvRlcD5q0mB+nyK9fB2aBzOiaiHSilWzAJeQjuqA+Q93jNew+peuj4AhdvGN
n/KK/+1Yv61w3+7g6ukFMARVBNG=
-----END CERTIFICATE-----
```

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

### Import certificates exported from Windows Server

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:  
<http://www.slproweb.com/download/Win32OpenSSL-v0.9.8b.exe>

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

Now you can type the below command to perform the conversion:

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

To convert your .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.



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

To convert your .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.

## Physical Load Balancer Configuration

The physical load balancer configuration is unique to each individual load balancer.

### 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 MASQ (NAT) routing then you will need to configure the external network *eth1* as well (or an alias on *eth0* via the *rc.firewall* script).

It is recommended to configure your *Default Gateway* here.

### DNS & Hostname

It is important that the master and slave load balancer have the correct *hostname* set in order for replication of data via SCP to work. 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, i.e. a router.

Serial	<input type="text" value="ttyS0"/>	
Bcast	<input type="text" value="none"/>	
Keepalive	<input type="text" value="3"/>	
Deadtime	<input type="text" value="10"/>	
Warntime	<input type="text" value="5"/>	
Ping node	<input type="text"/>	
Auto_failback	<input type="text" value="on"/>	

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

## Global Settings

This form allows you to change the global time outs for the health checking agent '*Ldirectord*'. It is recommended that you leave the *check interval* at 10 seconds and *check timeout* at 5 seconds. You may want the *negotiate timeout* set higher as negotiate checks take longer. When tuning these figures pay careful attention to the *Ldirectord* log.

The quiescent setting controls whether a real server is completely removed from the load balancer routing table when it has failed a health check or if the weight is just set to zero. When quiescent is set to 'no' then a real server failure will result in all connections moving to another server. When quiescent is set to 'yes' non-persistent connections will time out in 2 minutes (**or on a client re-connect**) but persistent connection will continue being directed to the downed real server until the persistence time out value expires.

**EDIT CONFIGURATION > GLOBAL SETTINGS**

---

**Layer 4:**

Check Interval	<input type="text" value="6"/>	<a href="#">?</a>
Check Timeout	<input type="text" value="3"/>	<a href="#">?</a>
Negotiate Timeout	<input type="text" value="5"/>	<a href="#">?</a>
Quiescent	<input type="text" value="no"/> <a href="#">v</a>	<a href="#">?</a>
Email Alerts	<input type="text" value="support@mycompany.c"/>	<a href="#">?</a>

**Pound SSL:**

Logging	<input type="text" value="on"/> <a href="#">v</a>	<a href="#">?</a>
Client Timeout	<input type="text" value="30"/>	<a href="#">?</a>
Global Server Timeout	<input type="text" value="60"/>	<a href="#">?</a>

**HAProxy:**

Logging	<input type="text" value="on"/> <a href="#">v</a>	<a href="#">?</a>
contimeout	<input type="text" value="4000"/>	<a href="#">?</a>
clitimeout	<input type="text" value="42000"/>	<a href="#">?</a>
srvtimeout	<input type="text" value="43000"/>	<a href="#">?</a>
maxconn	<input type="text" value="40000"/>	<a href="#">?</a>

When you configure an email address for alerts all server health related events will be automatically delivered.

The Pound SSL & HAProxy logs will be in */var/log/poundssl* and */var/log/haproxy* respectively when enabled.

## Maintenance

### *Maintain Real Servers*

#### System Overview

##### Take a real server offline or online

This form allows you to view all of the Virtual Servers and associated Real Servers, port numbers and weights. Clicking 'take offline' or 'bring online' will change the weight of the server to either 0 or 1 respectively.

This form has been largely replaced by *View Configuration > System Overview*.

#### **MAINTENANCE > TAKE A REAL SERVER OFFLINE OR ONLINE**

Check Status

Number	Label	IP:Port	Active Connections	Requested Status (weight)	Change Status	Actual Status
VIP 1	<i>vip1</i>	192.168.2.246:80	0			<i>non-active</i>
RIP 2	<i>rip1-2</i>	192.168.2.10:80		ONLINE (1)	take offline	<i>non-active</i>
RIP 1	<i>rip1-1</i>	192.168.2.9:80		ONLINE (1)	take offline	<i>non-active</i>
VIP 2	<i>VIP2</i>	192.168.2.183:80	0			<i>non-active</i>
RIP 3	<i>rip2-3</i>	192.168.2.10:80		ONLINE (1)	take offline	<i>non-active</i>
RIP 2	<i>RIP_Name</i>	192.168.2.9:80		ONLINE (1)	take offline	<i>non-active</i>
RIP 1	<i>rip2-2</i>	192.168.2.99:80		ONLINE (1)	take offline	<i>non-active</i>
VIP 3	<i>vip3</i>	192.168.2.155:80	0			<i>non-active</i>
RIP 1	<i>rip3-1</i>	192.168.2.10:80		ONLINE (1)	take offline	<i>non-active</i>

Some points to bear in mind:

- This is for Layer 4 services only
- If you want to take a server down for maintenance
  - Take it offline (i.e. set the weight to zero)
  - Then either wait 2 minutes (even HTTP 1.1 has some persistence)
  - Or look in the status report and wait for active connections to fall to zero
- The online or offline status here is what you WANT, not what you've GOT
  - The active or inactive status is what you've GOT after health checks are taken into account
- Changes may take a few seconds to take effect depending on the current status of ldirectord
- When you take a server offline and then bring it back online the weight is always set to 1. If you need to change the weight just click on it to be taken to the modify real server screen.

## ***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 site (*each license covers up to 6 appliances on a single site*) 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](http://www.loadbalancer.org) enabled through your firewall.

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

*NB. You will need to update both the Master & the Slave one at a time. 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.

*NB. If you manually take all the servers offline this page will NOT be shown, if you want to force it to show then shut down your web 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.



In DR mode the destination port of a connection cannot be changed, so if you want the built in fallback server to work on 127.0.0.1:80 then you need to add the NGINX binding port 80 to the default 9081. This can be done with the console command `lb2ports`. To revert this back to only 9081 use the command `lb1port`.



**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 advanced topics).

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
```

### Scripted Alias configuration (for extra subnets)



Aliases can now be configured from the *Edit Configuration > Network Interface configuration section*.

Any additional Floating IP must have an interface to bind to i.e. eth0 or eth1 in same subnet.  
If you need extra subnets for your Floating IP you will need to define an alias in a local script.



**WARNING: BE CAREFUL!** If you configure anything in the master script it is automatically replicated to the slave (so make sure you make any unique changes to the slave **after** the master).

An example alias in the script would be:

```
#master  
ifconfig eth0:200 192.168.1.21 netmask 255.255.255.0 broadcast 192.168.1.255 up
```

It is important that you do this on the master (assuming you have one) first and then repeat the process on the slave but change the IP address (must be different obviously):

```
#slave  
ifconfig eth0:200 192.168.1.22 netmask 255.255.255.0 broadcast 192.168.1.255 up
```

You could add these lines to the end of the /etc/rc.d/rc.local file which runs at the end of the boot process.

*NB. From v6.7 aliases and VLANs can be generated via the web interface directly.*

### 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.*



**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*.

File Edit View Go Tools Window Help Debug QA

Back Forward Reload Stop

http://192.168.1.100/haproxy?stats

The Mozilla Organiza... Latest Builds

HAProxy

Statistics Report for pid 11389

> General process information

pid = 11389 (nbproc = 1)  
uptime = 0d 0h 21m 16s  
system limits : memmax = unlimited ; ulimit-n = 20039  
maxsock = 20039  
maxconn = 10000 (current conns = 421)

active UP  
active UP, going down  
active DOWN, going up  
active or backup DOWN  
backup UP  
backup UP, going down  
backup DOWN, going up  
not checked

> Proxy instance www.customer2.com : 0 conns (maxconn=10000), 0 queued (0 unassigned), 0 total conns

Server		Queue		Sessions				Errors							
Name	Weight	Status	Act.	Bck.	Curr.	Max.	Curr.	Max.	Limit	Cumul.	Conn.	Resp.	Sec.	Check	Down
dell1	20	UP 2/3 ↓	Y	-	0	0	0	0	100	0	0	0	0	1	0
dell2	20	UP	Y	-	0	0	0	0	100	0	0	0	0	0	0
p3-800	10	UP 2/3 ↓	-	Y	0	0	0	0	50	0	0	0	0	1	0
Dispatcher	-	UP	-	-	0	0	0	0	10000	0	0	0	0	-	-
Total	-	UP	2	1	0	0	0	0	10000	0	0	0	0	2	0

> Proxy instance www.customer1.com : 421 conns (maxconn=10000), 0 queued (0 unassigned), 14427 total conns

Server		Queue		Sessions				Errors							
Name	Weight	Status	Act.	Bck.	Curr.	Max.	Curr.	Max.	Limit	Cumul.	Conn.	Resp.	Sec.	Check	Down
xeon-2.8G	20	UP 2/3 ↓	Y	-	0	0	84	100	100	1477	0	0	0	1	0
opte-2.2G	22	UP	Y	-	0	0	92	110	110	1637	0	0	0	0	0
opte-2.4G	24	UP	Y	-	0	0	104	120	120	1791	0	0	0	0	0
p3-800	10	UP 2/3 ↓	-	Y	0	0	0	0	50	0	0	0	0	1	0
devel	10	DOWN	Y	-	0	0	0	0	5	0	0	0	0	0	1
devel-back	10	DOWN	-	Y	0	0	0	0	5	0	0	0	0	0	1
Dispatcher	-	UP	-	-	0	244	141	710	10000	9522	0	0	0	-	-
Total	-	UP	3	1	0	244	421	710	10000	14427	0	0	0	2	2

## 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)

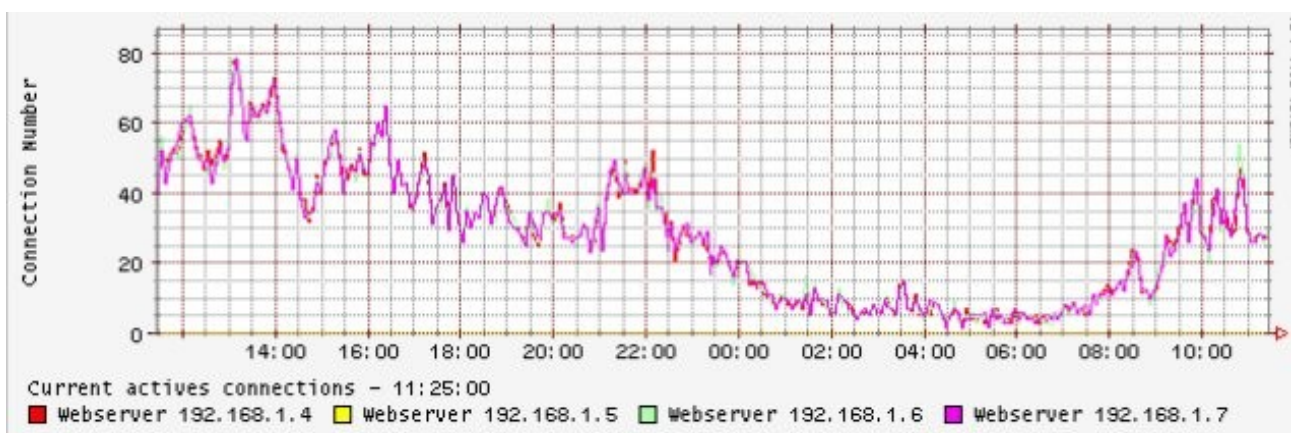
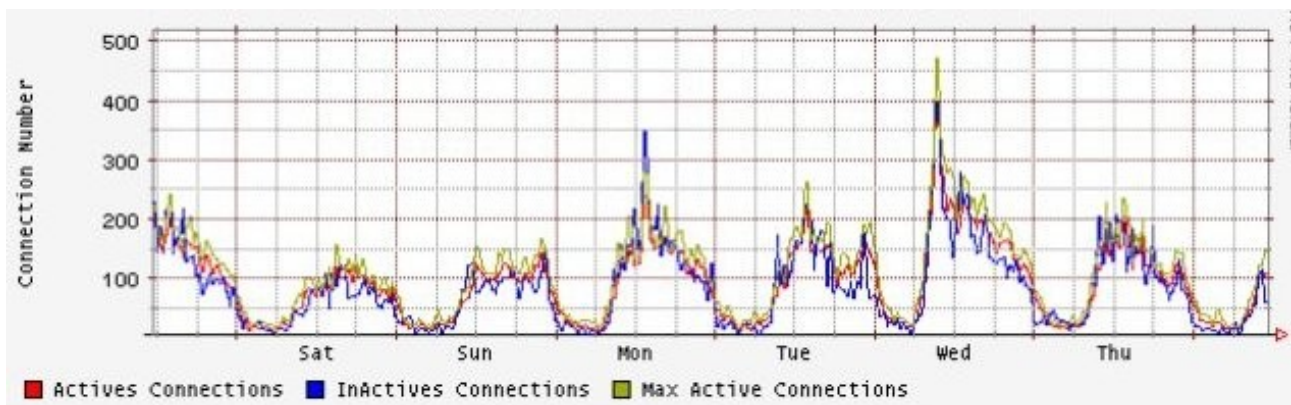
This is the same as the current connections report but is much slower as it looks up the DNS name of each IP address.

## 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 there to prove 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.