Appliance Administration v6.6

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

Table of Contents

Loadbalancer.org terminology

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

What is a virtual IP address?

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

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

What are your objectives?

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

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

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

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

ℹ 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 terminology definition:

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

What are the different load balancing methods supported?

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

Layer 4	DR (Direct Routing)	Ultra-fast local server based load balancing Requires handling the ARP issue on the real servers	1 ARM
Layer 4	NAT (Network Address Translation)	Fast Layer 4 load balancing, the appliance becomes the default gateway for the real servers	2 ARM
Layer 4	TUN	Similar to DR but works across IP encapsulated tunnels	1 ARM
Layer 7	SSL Termination *(Pound)*	Usually required in order to process cookie persistence in HTTPS streams on the load balancer - Processor intensive	1 ARM
Layer 7	SNAT (HAProxy)	Layer 7 allows great flexibility including full SNAT and WAN load balancing, cookie insertion and URL switching Much slower than Layer 4	1 ARM

Key:

🟨 Recommended

🟪 Recommended only if cookie insertion is mandatory

⬜ Only required for Direct Routing implementation across routed networks

The one-arm direct routing (DR) mode is the recommended mode for Loadbalancer.org installations 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 RIP & VIP at the same time; or if the host operating system cannot be modified to handle the ARP issue (*see real server configuration section*).

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).

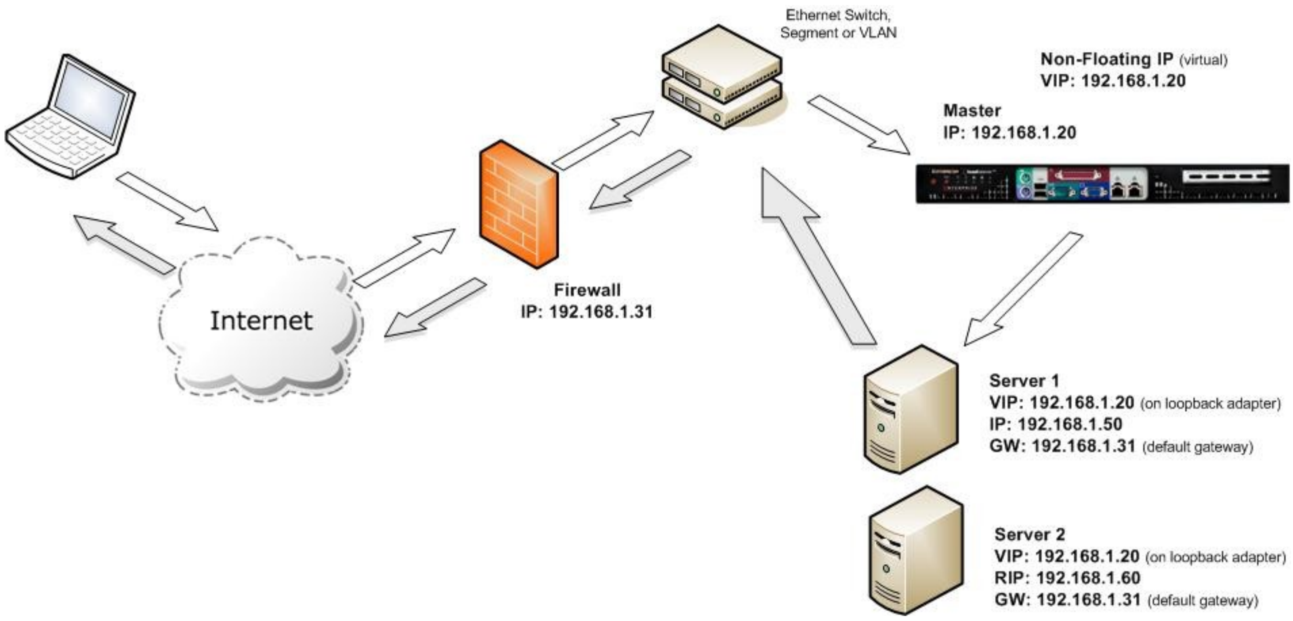
Network engineers with experience of hardware load balancers will have often used this method.

If your application requires that the load balancer handles cookie insertion then you need to use the SNAT configuration. 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 the administration manual 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)

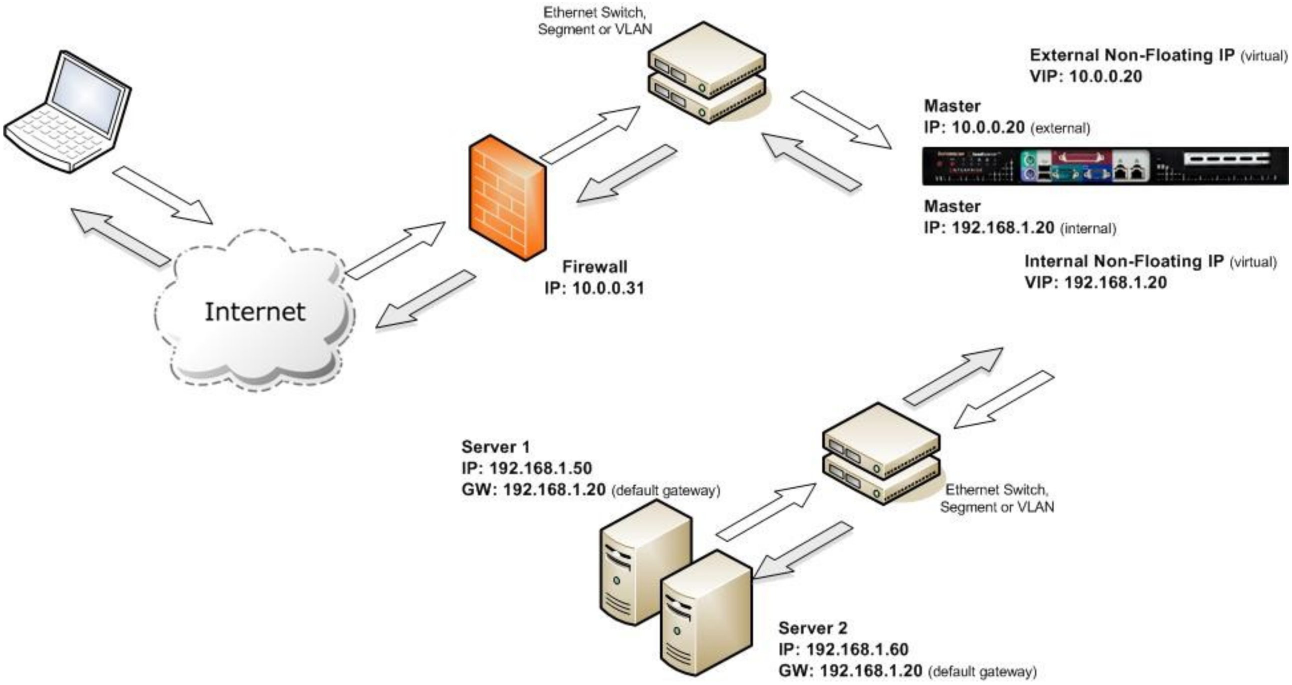

Ethernet Switch, Segment or VLAN
Non-Floating IP (virtual)
VIP: 192.168.1.20
Master
IP: 192.168.1.20
Internet
Firewall
IP: 192.168.1.31
Server 1
VIP: 192.168.1.20 (on loopback adapter)
IP: 192.168.1.50
GW: 192.168.1.31 (default gateway)
Server 2
VIP: 192.168.1.20 (on loopback adapter)
RIP: 192.168.1.60
GW: 192.168.1.31 (default gateway)

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.
- The real servers must have their default gateway configured to point at the load balancer.
- If you want the real servers to be able to access the internet on their own, i.e. browse the web, you will need to set up a MASQUERADE rule in the firewall script *(some vendors incorrectly call this S-NAT).*
- If you want real servers to be accessible on their own IP address for non-load balanced services, i.e. SMTP, you will need to set up individual SNAT and DNAT firewall script rules for each real server.
- Move 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. The real servers must also have their default gateways 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. The administration manual has configuration details for one-arm NAT mode.

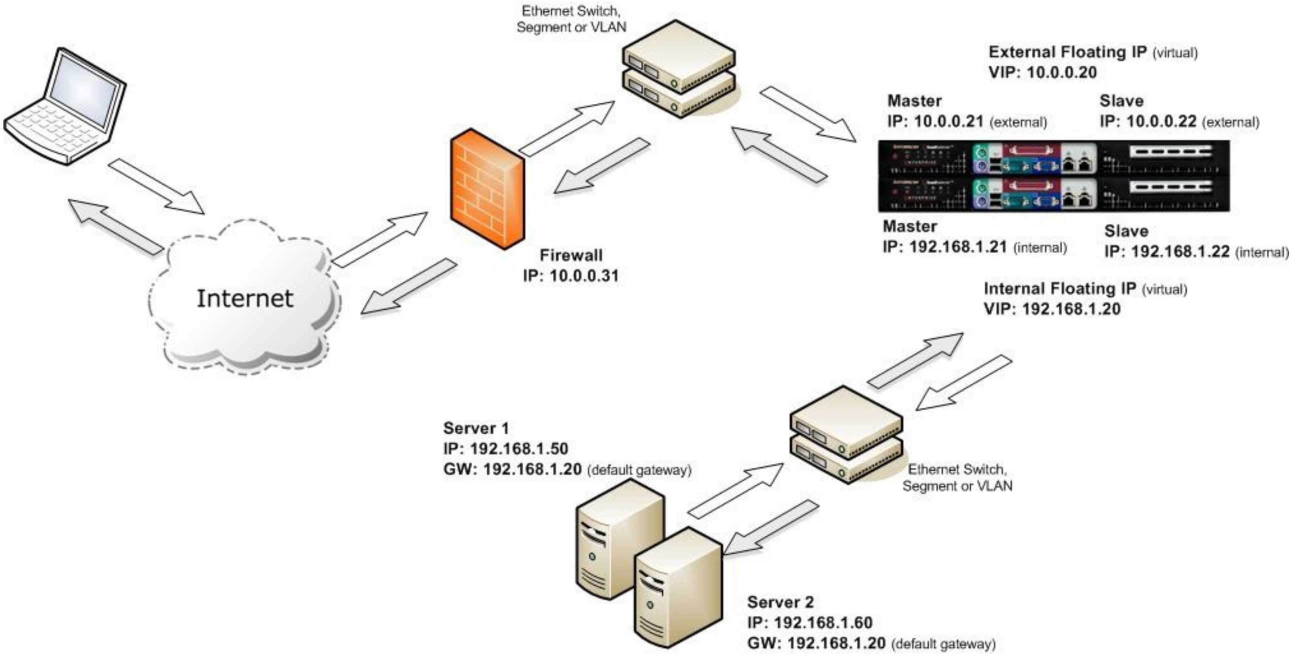
High-availability configuration of two Loadbalancer.org appliances

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)

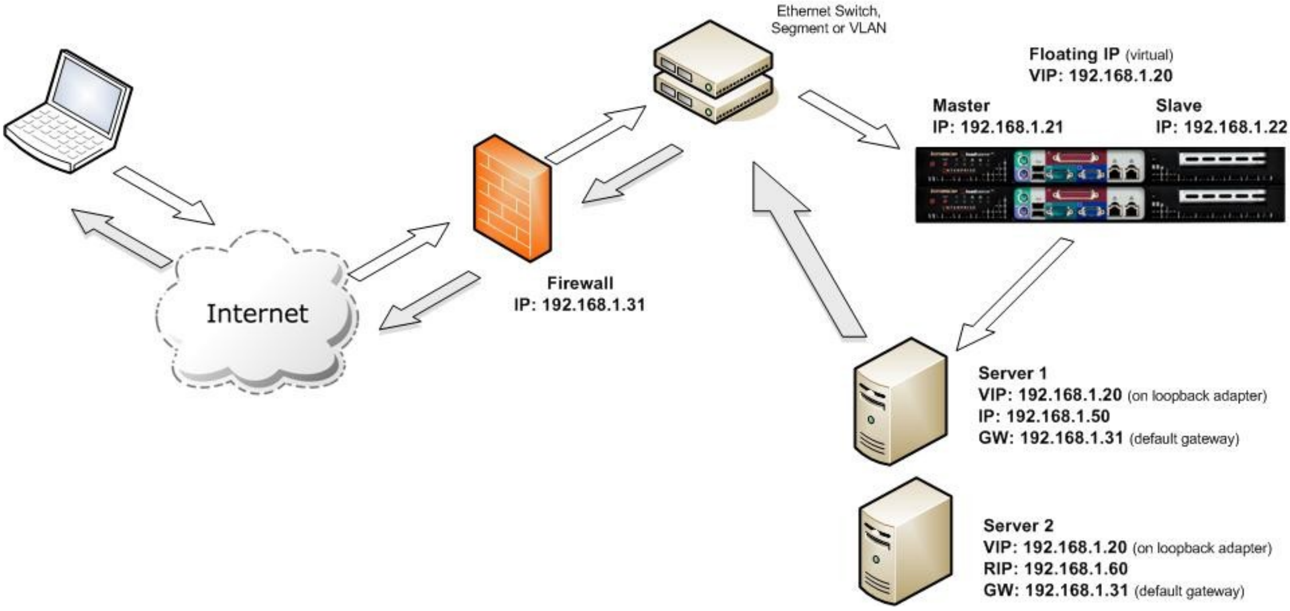

Ethernet Switch, Segment or VLAN
External Floating IP (virtual)
VIP: 10.0.0.20
Master
IP: 10.0.0.21 (external)
Slave
IP: 10.0.0.22 (external)
Internet
Firewall
IP: 10.0.0.31
Master
IP: 192.168.1.21 (internal)
Slave
IP: 192.168.1.22 (internal)
Internal Floating IP (virtual)
VIP: 192.168.1.20
Server 1
IP: 192.168.1.50
GW: 192.168.1.20 (default gateway)
Ethernet Switch, Segment or VLAN
Server 2
IP: 192.168.1.60
GW: 192.168.1.20 (default gateway)

Notes:

- Administration of the load balancers is via any active IP address.
- One floating IP must be configured for the real servers to use as a default gateway.
- One floating IP must be configured for hosting the virtual server.

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 gateways 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.

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

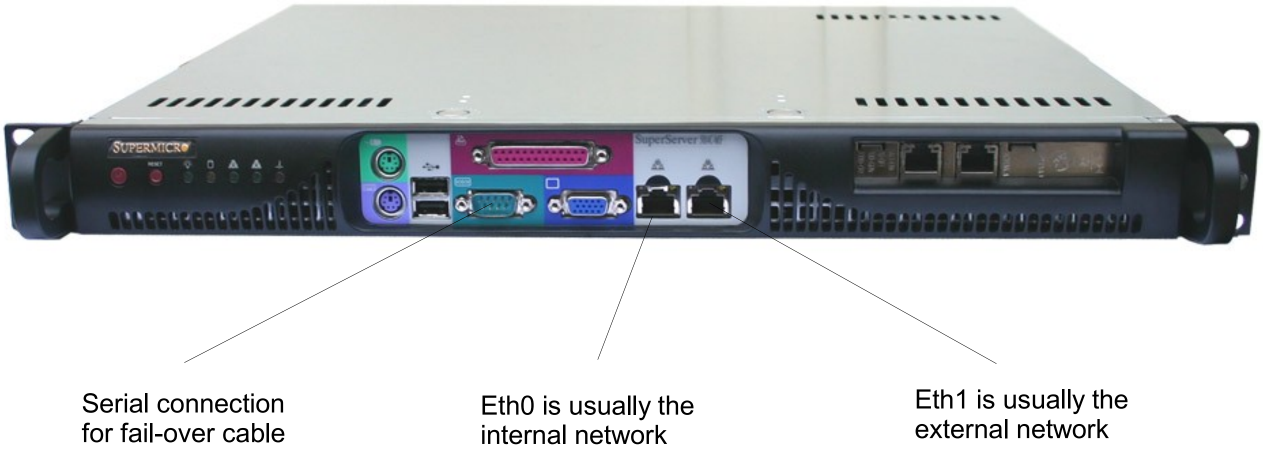
1. Remove all packaging
2. Rack mount the appliance as required using the supplied rails
3. The power supply is an auto sensing unit (115v or 230v)
4. Connect the power lead from the power socket to the mains or UPS
5. Connect your network cable from your switch or hub to the network port Gb1
6. If using a two-armed configuration connect a second network cable to port Gb2

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


SUPERMICRO
SuperServer
Serial connection for fail-over cable
Eth0 is usually the internal network
Eth1 is usually the external network

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/***

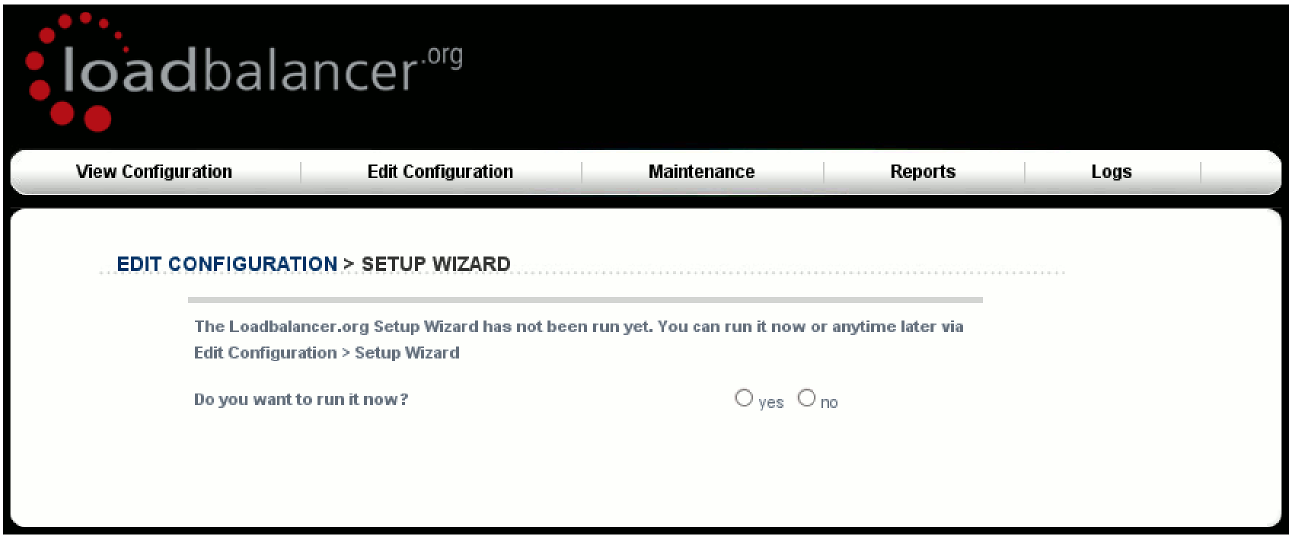
(replace 192.168.2.21 with the correct address)

Log in to the web interface:

User: **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.



EDIT CONFIGURATION > SETUP WIZARD

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

Do you want to run it now? ○ yes ○ no

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.

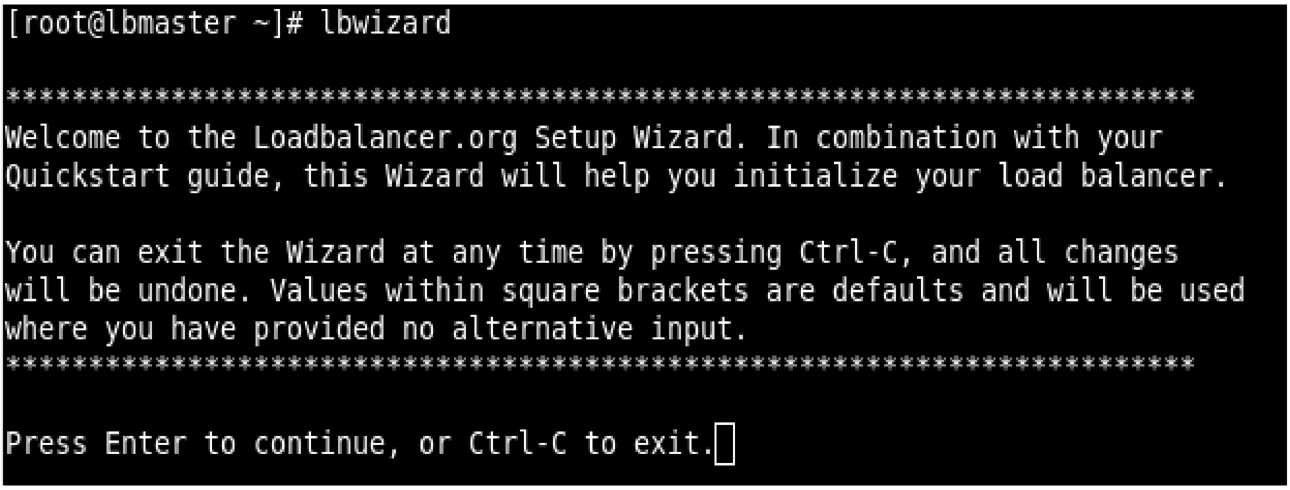
NB. For full configuration using only the web interface please see the administration manual

Network interface configuration

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

- Type the following command:
  - lbwizard

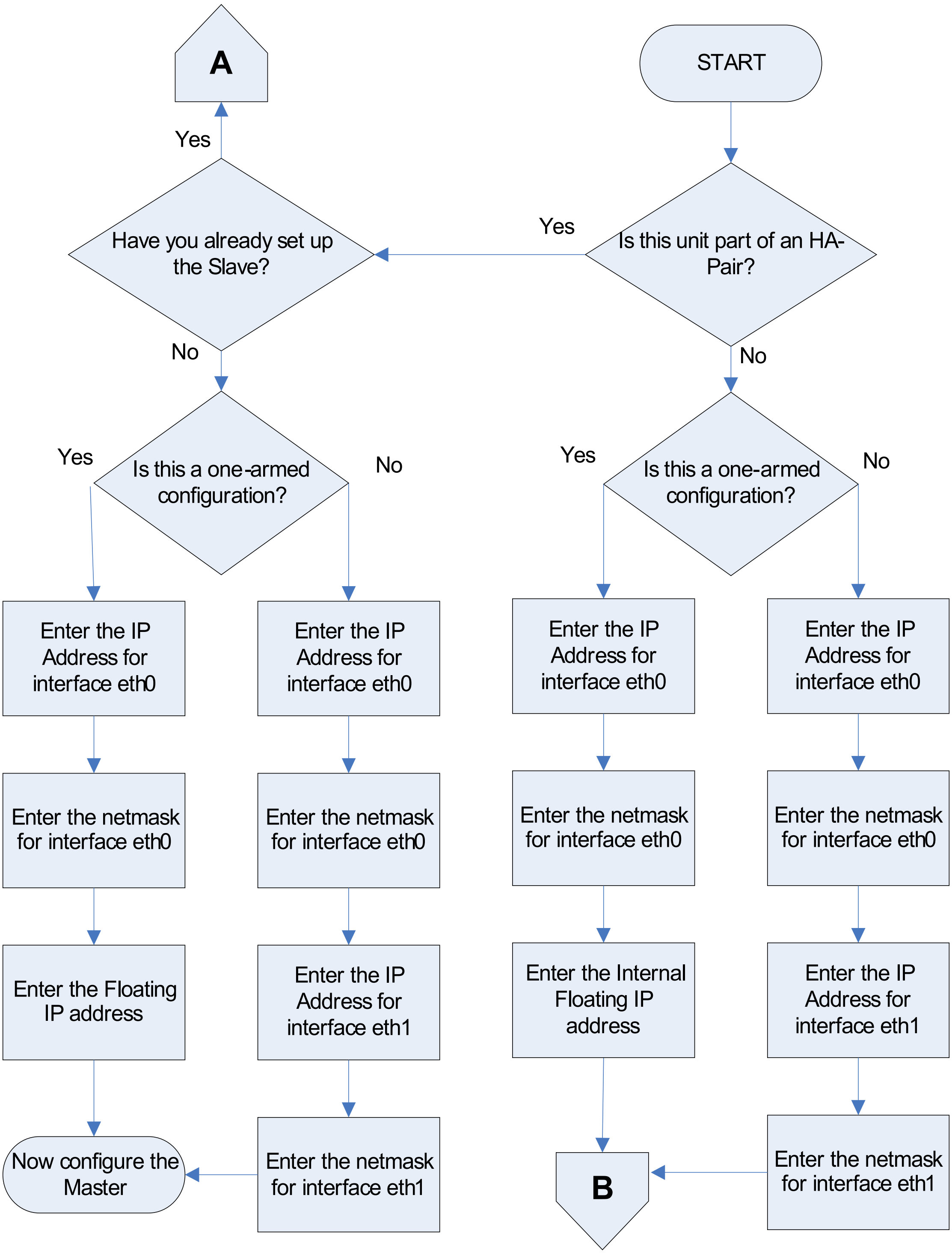
This activates the console based configuration wizard which will ask a series of questions in order to get you started quickly.



```
[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.
```

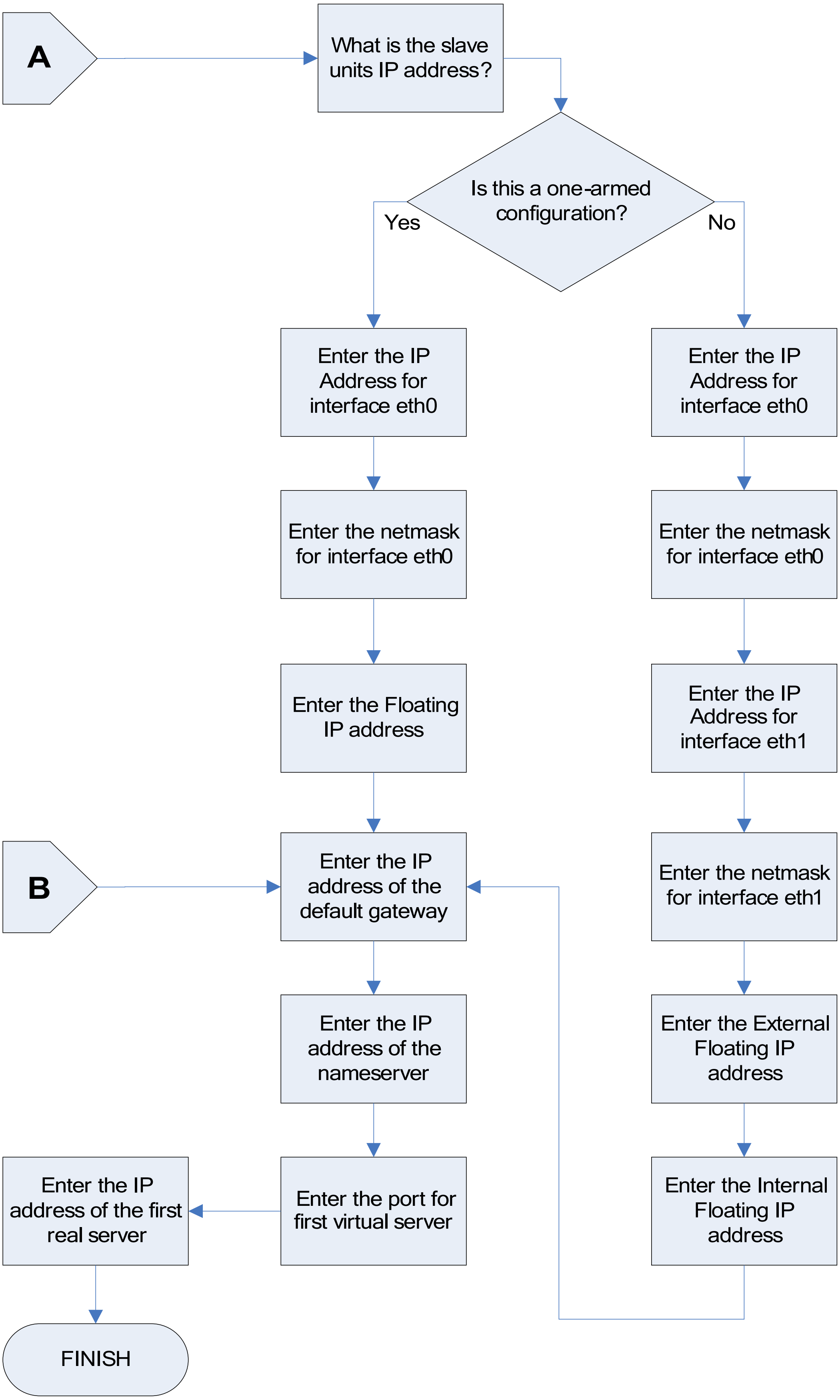
Flow diagram explanation of the wizards

This flow diagram (in two parts) explains the possible paths to take when configuring your appliance using eithet the web based wizard or the console wizard:


A
START
Yes
Have you already set up the Slave?
Yes
Is this unit part of an HA-Pair?
No
No
Yes
Is this a one-armed configuration?
No
Yes
Is this a one-armed configuration?
No
Enter the IP Address for interface eth0
Enter the IP Address for interface eth0
Enter the IP Address for interface eth0
Enter the IP Address for interface eth0
Enter the netmask for interface eth0
Enter the netmask for interface eth0
Enter the netmask for interface eth0
Enter the netmask for interface eth0
Enter the Floating IP address
Enter the IP Address for interface eth1
Enter the Internal Floating IP address
Enter the IP Address for interface eth1
Now configure the Master
Enter the netmask for interface eth1
B
Enter the netmask for interface eth1

Flow diagram explanation of the wizards, diagram #1

Flow diagram explanation of the wizards, diagram #2


A
What is the slave units IP address?
Is this a one-armed configuration?
Yes
No
Enter the IP Address for interface eth0
Enter the netmask for interface eth0
Enter the Floating IP address
Enter the IP Address for interface eth0
Enter the netmask for interface eth0
Enter the IP Address for interface eth1
Enter the netmask for interface eth1
Enter the External Floating IP address
Enter the Internal Floating IP address
B
Enter the IP address of the default gateway
Enter the IP address of the nameserver
Enter the port for first virtual server
Enter the IP address of the first real server
FINISH

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

Once you have decided on your load balancing configuration the questions in the console 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 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 interface eth0 [192.168.2.21]:
Answer: 10.0.0.21 (This is for the external subnet)

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

Question: Enter the IP address for interface eth1 [10.0.0.21]:
Answer: 192.168.2.21 (This is for the internal 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 (backend) [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.

Once the wizard is completed 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).

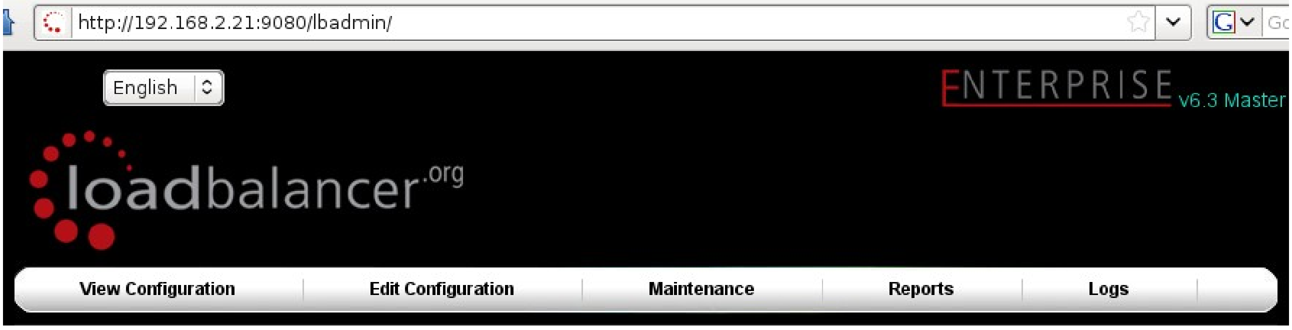
With a web browser access the web interface i.e. ***http://192.168.2.21:9080/lbadmin/***

(replace 192.168.2.21 with the correct address)

Log in to the web interface:

User: **loadbalancer**
Password: **loadbalancer**

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

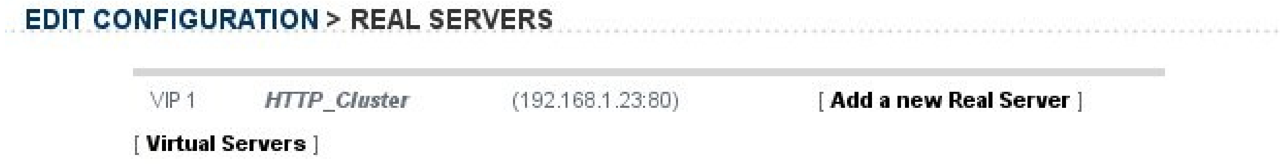


All of your administration tasks can be carried out through the web interface.

Additional real servers (web interface)

The console wizard sets up one virtual server with one real server (backend 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**.

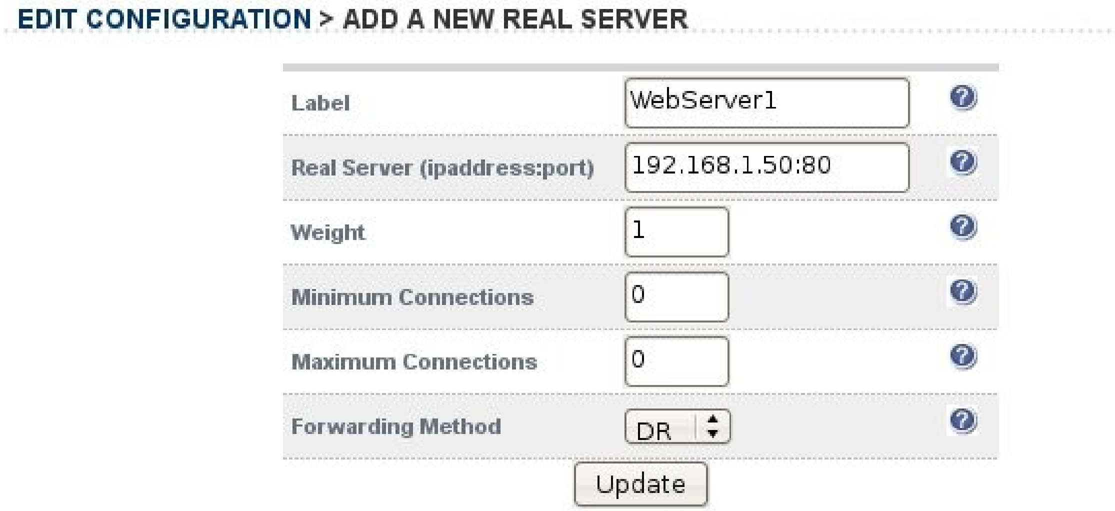


EDIT CONFIGURATION > REAL SERVERS

VIP 1 *HTTP_Cluster* (192.168.1.23:80) [**Add a new Real Server**]

[**Virtual Servers**]

- You just need to give the IP address and port number of your web server.



EDIT CONFIGURATION > ADD A NEW REAL SERVER

Label	WebServer1
Real Server (ipaddress:port)	192.168.1.50:80
Weight	1
Minimum Connections	0
Maximum Connections	0
Forwarding Method	DR

- Correctly specify your real servers IP address and service port.
- Weight defaults to 1 making real servers active immediately.
- Leave the minimum & maximum connections as 0 for unrestricted.
- The forwarding method will default to NAT if you have a two-arm configuration or DR if you have a one-arm configuration.

You have now finished the configuration of both load balancers for the cluster. Now you must configure the 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: *Advanced NAT considerations* in the administration manual.

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)

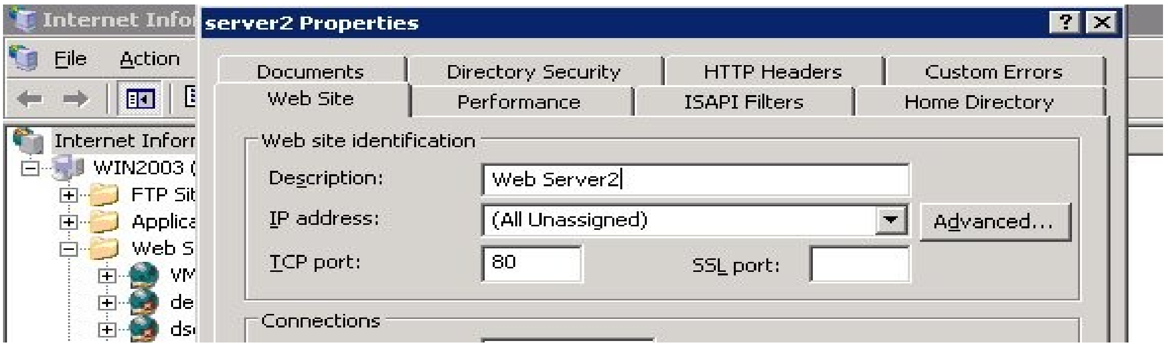
Failure to correctly configure the real servers to handle the ARP problem is the most common problem in DR configurations. Please refer to: *Advanced DR considerations* in the administration manual.

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: *Advanced DR considerations* in the administration manual.



Description: Web Server2

IP address: (All Unassigned)

TCP port: 80

SSL port:

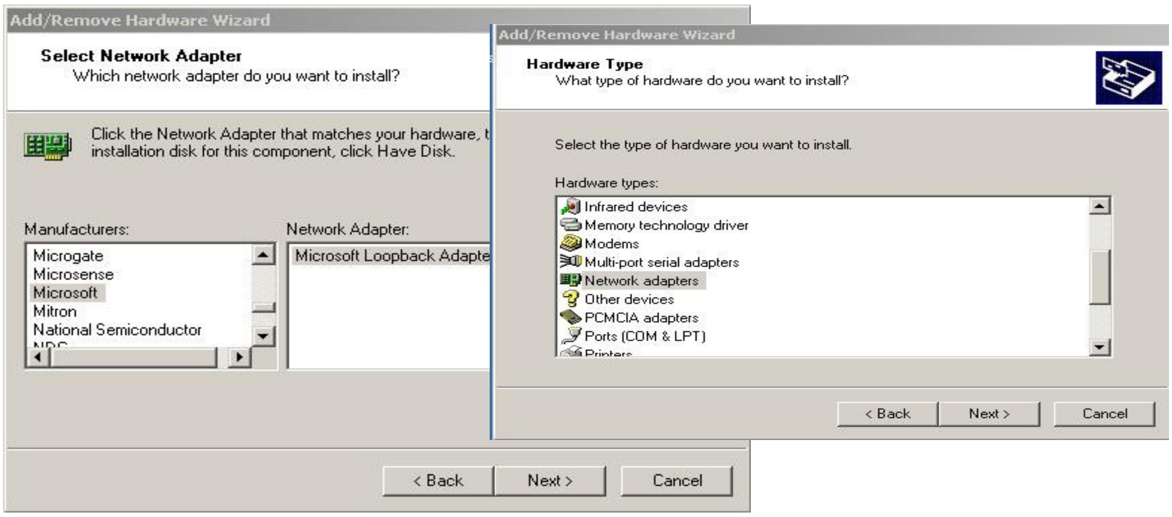
Remember that for all real servers in Direct Routing mode the load balanced application must respond to both the virtual IP as well as the servers real IP. With Windows the IP address must either be set to (All Unassigned) or use the Advanced tab to add a second IP address.

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.



Add/Remove Hardware Wizard

Select Network Adapter
Which network adapter do you want to install?

Click the Network Adapter that matches your hardware, t
installation disk for this component, click Have Disk.

Manufacturers: Microsoft

Network Adapter: Microsoft Loopback Adapte

Add/Remove Hardware Wizard

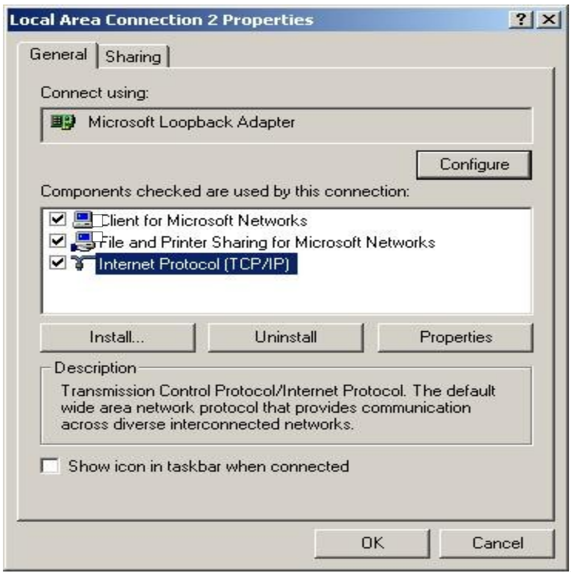
Hardware Type
What type of hardware do you want to install?

Select the type of hardware you want to install.

Hardware types: Network adapters

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



Local Area Connection 2 Properties

General | Sharing

Connect using:

Microsoft Loopback Adapter

Components checked are used by this connection:

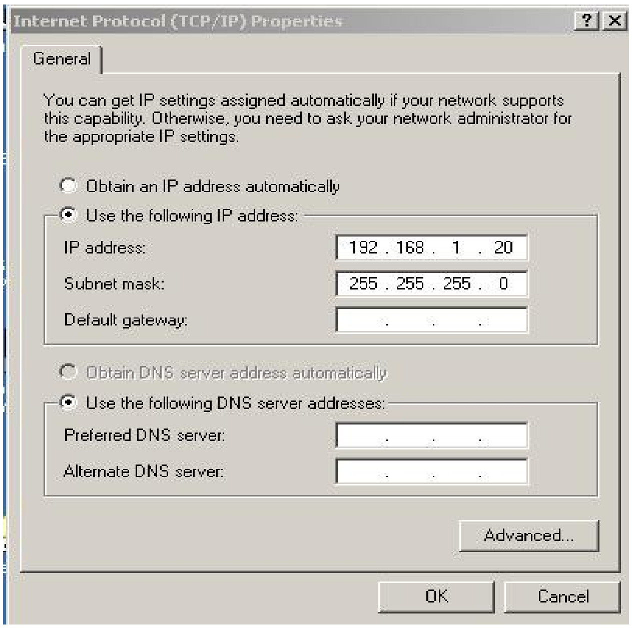
☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☑ Internet Protocol (TCP/IP)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

☐ Show icon in taskbar when connected

5. Select TCP/IP Properties



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

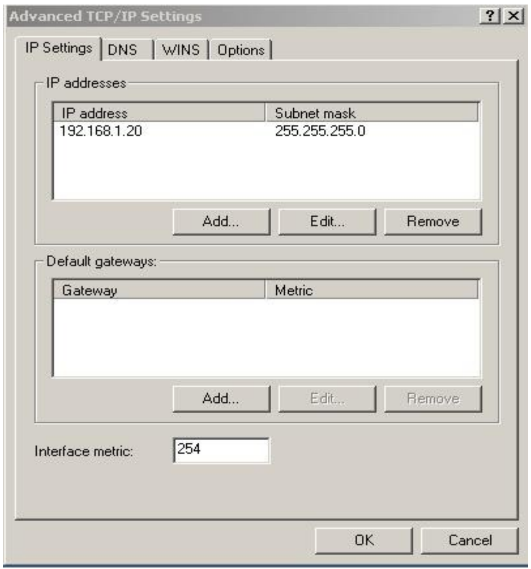
IP address: 192 . 168 . 1 . 20
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: . . .
Alternate DNS server: . . .

4. Fill in the virtual server IP address i.e. 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



IP addresses

IP address	Subnet mask
192.168.1.20	255.255.255.0

Default gateways:

Gateway	Metric

Interface metric: 254

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

For Windows server 2003 you will need to disable the built in firewall (or manually changes the rules). By default the Windows firewall will block all connections to the Loopback adapter.

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 you will need to disable the built in firewall (or manually change the rules). By default the Windows firewall will block all connections to the Loopback adapter.

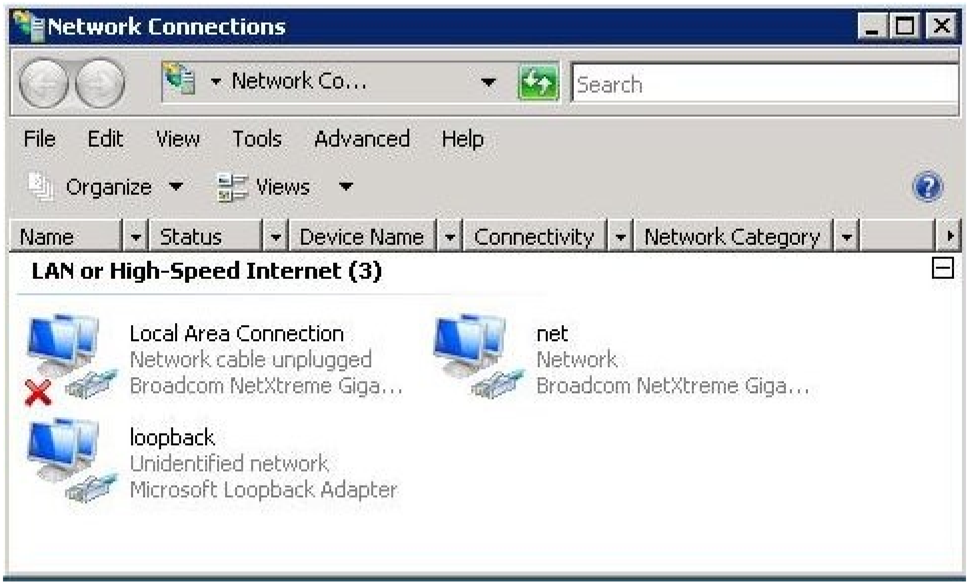
Weak and strong host behavior in Windows

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.



LAN or High-Speed Internet (3)

Local Area Connection
Network cable unplugged
Broadcom NetXtreme Giga...

net
Network
Broadcom NetXtreme Giga...

loopback
Unidentified network
Microsoft Loopback Adapter

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 i.e. 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.

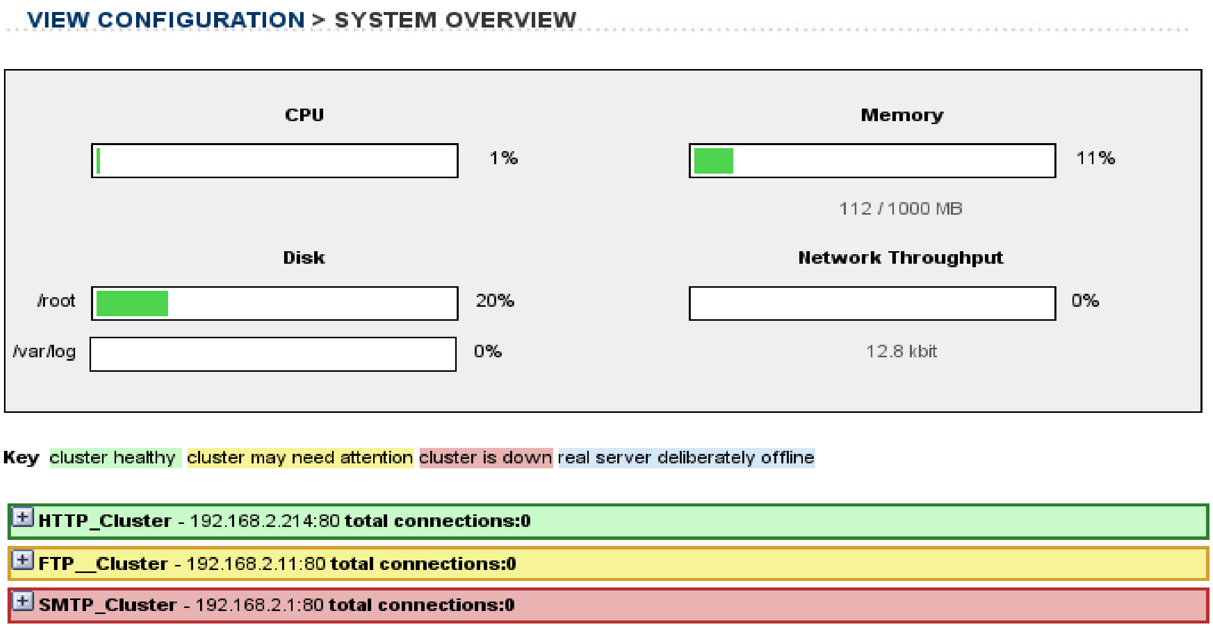
Why test two clients? *If you use a single client it will most likely keep on hitting the same server for multiple requests. This is to do with the way that the load balancing algorithms are optimized.*

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**

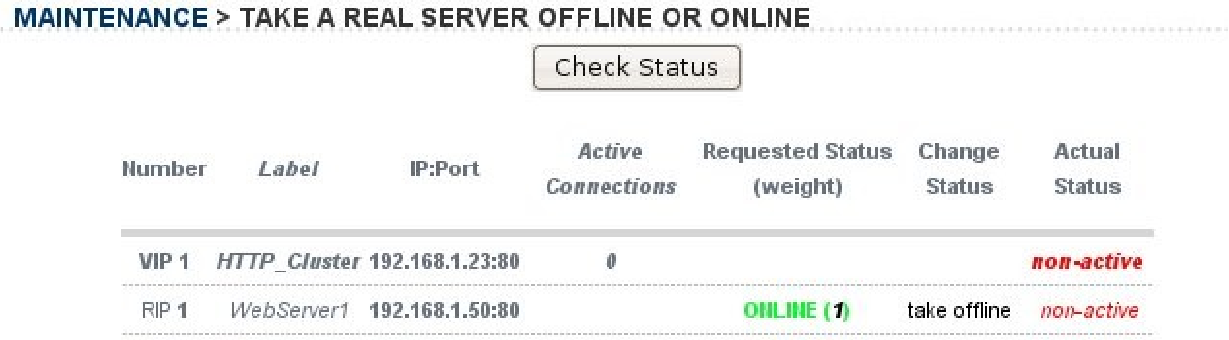
⊞ **SMTP_Cluster** - 192.168.2.1:80 **total connections:0**

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 > Take a real server offline or online* section of the web interface and check that when you take servers offline the connections are redirected to the rest of the cluster as expected.

The example below shows that the requested status of WebServer1 is online but the actual status is non-active. 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*.



MAINTENANCE > TAKE A REAL SERVER OFFLINE OR ONLINE

Number	*Label*	IP:Port	*Active Connections*	Requested Status (weight)	Change Status	Actual Status
VIP 1	*HTTP_Cluster*	192.168.1.23:80	*0*			*non-active*
RIP 1	*WebServer1*	192.168.1.50:80		ONLINE (*1*)	take offline	*non-active*

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.

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).

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.

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

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. *This is the only persistence method available for non HTTP/HTTPS based applications.*

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 (see example 3).

Loadbalancer.org technical support

If you have any questions regarding the Loadbalancer.org appliance don't hesitate to contact the support team support@loadbalancer.org.

Loadbalancer.org hardware support

Hardware support can be arranged either direct with Armari, or through our dedicated support team.

Configuring the Loadbalancer.org appliance as a single unit (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:
  - **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 *192.168.2.21/255.255.255.0.*
Just log into *http://192.168.2.21:9080/lbadmin/*
Username**:** *loadbalancer*
P**assword:** *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

When you are only configuring a single load balancer 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 eth1. 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 if required in global settings.

Configuring the Loadbalancer.org appliance as a 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.

Network interface configuration

- Power up the slave load balancer first
- Log in:
  - 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. If you have not already changed the administration (eth0) address then set up a client with an IP address in the correct subnet i.e. 192.168.2.11/24.
Then log into the default administration address: *http://192.168.2.21:9080/lbadmin/*

User: **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.
- After the master has booted, just log into http://192.168.2.21:9080/lbadmin/ 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 slave load balancer
- Now any changes to the configuration of the master load balancer will be automatically replicated to the slave.

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.

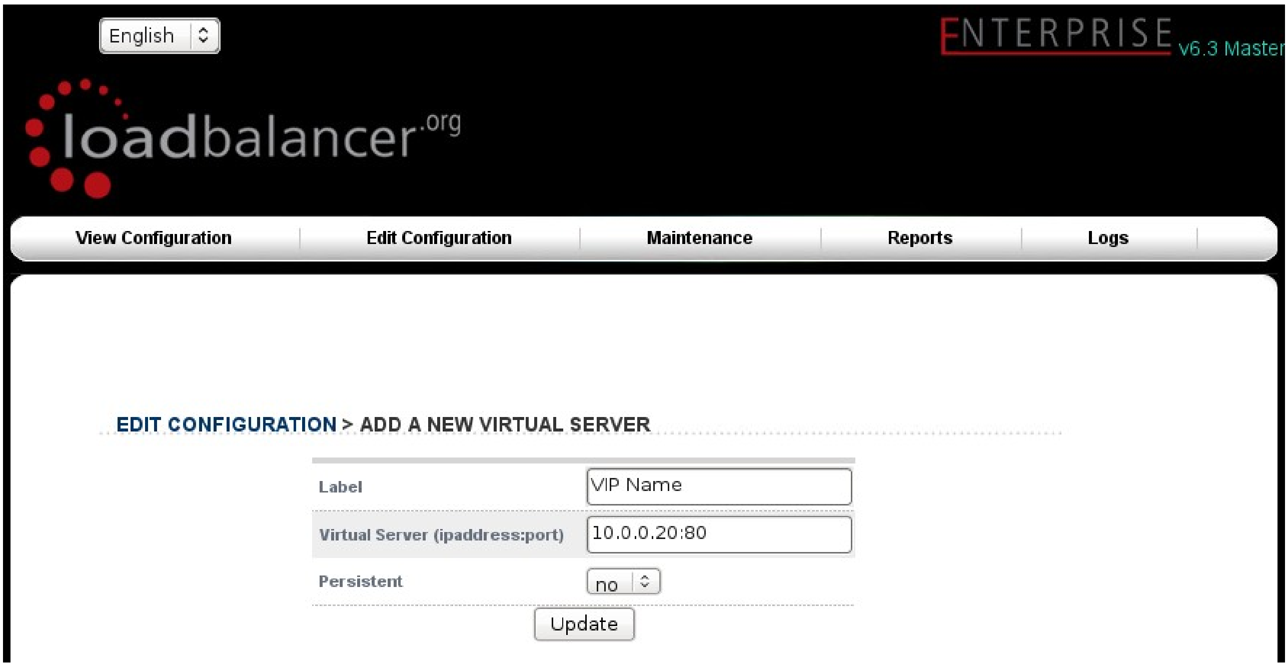
If you are using two-arm NAT mode you must also configure an IP address on eth1. Then configure an internal Floating IP for the Real Servers to use this as the default gateway.

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* > *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.



EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER

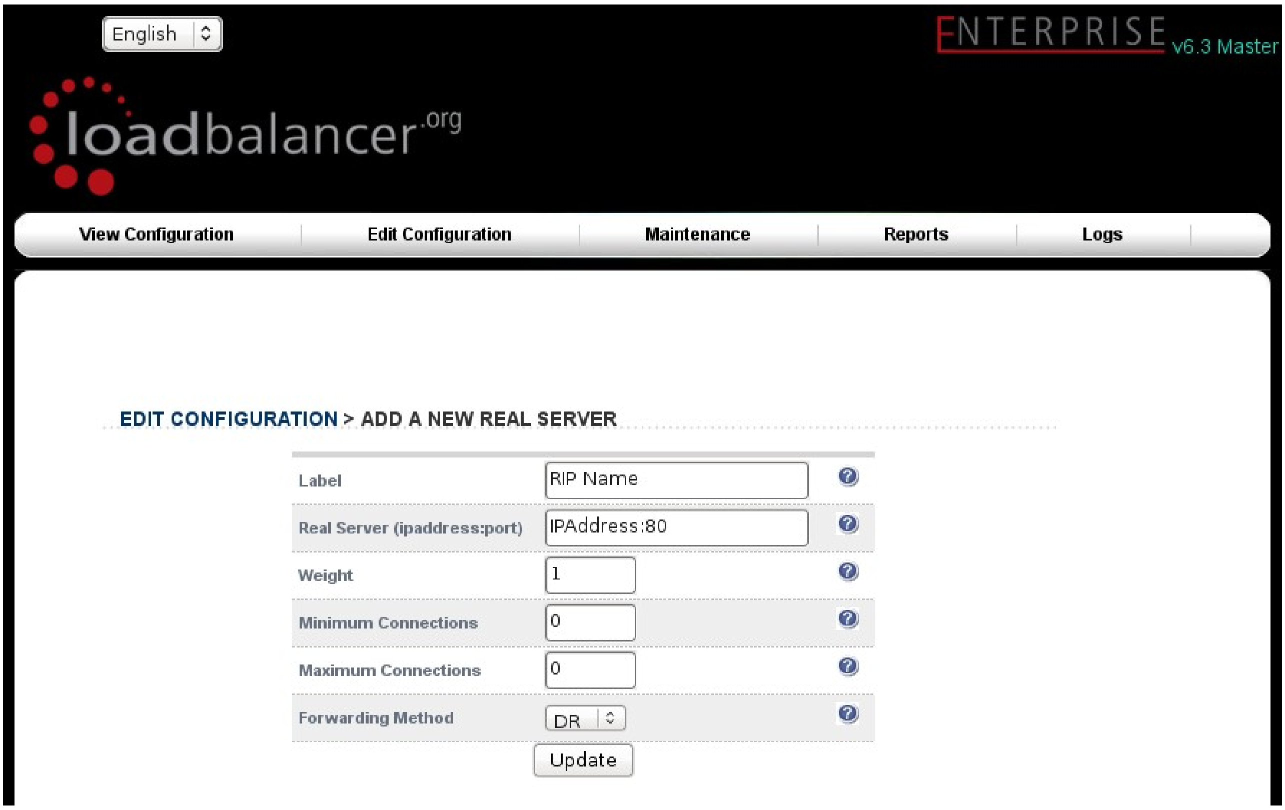
Label	VIP Name
Virtual Server (ipaddress:port)	10.0.0.20:80
Persistent	no

- For this example we are load balancing both HTTP and HTTPS so you need to set up 2 Virtual Servers, for example 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 (web servers) to the cluster.

Real Server configuration (RIP) – one arm DR method

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

- 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 real server**.
- You just need to give the IP Address and Port number of your web server.



EDIT CONFIGURATION > ADD A NEW REAL SERVER

Label	RIP Name
Real Server (ipaddress:port)	IPAddress:80
Weight	1
Minimum Connections	0
Maximum Connections	0
Forwarding Method	DR

Update

- 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 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.

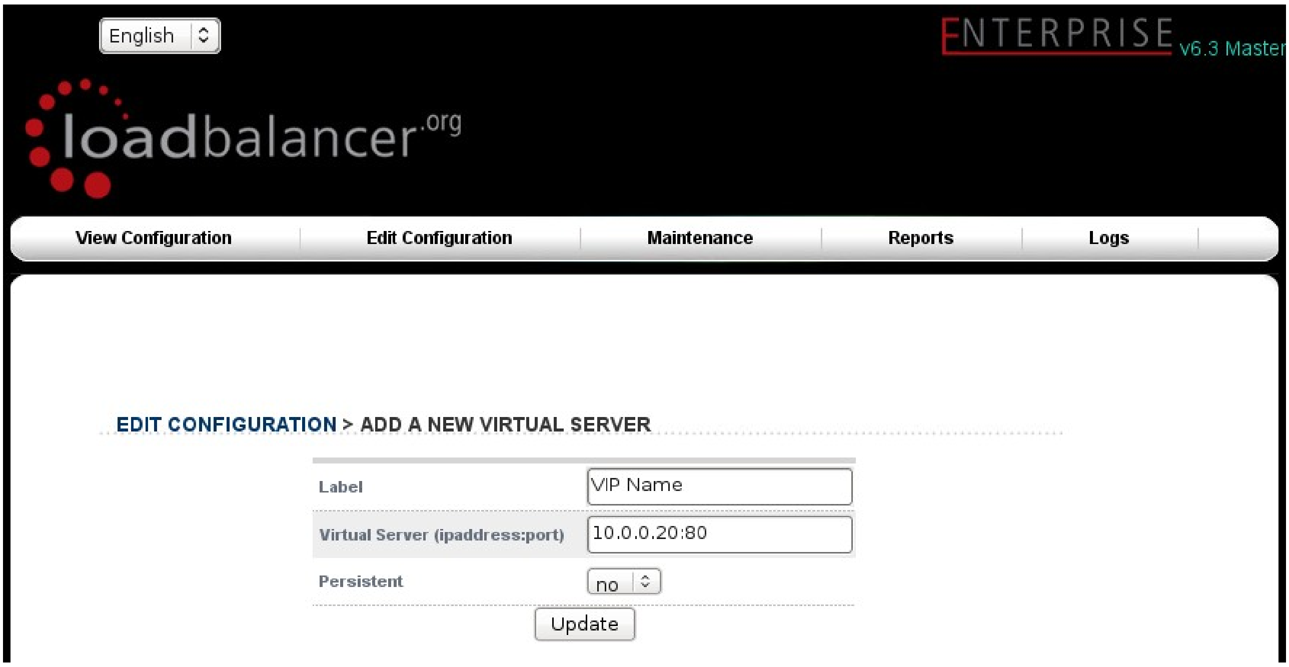
Failure to correctly configure the real servers to handle the ARP problem is the most common problem in DR configurations. Please refer to: *Advanced DR Considerations*

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* > *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.



EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER

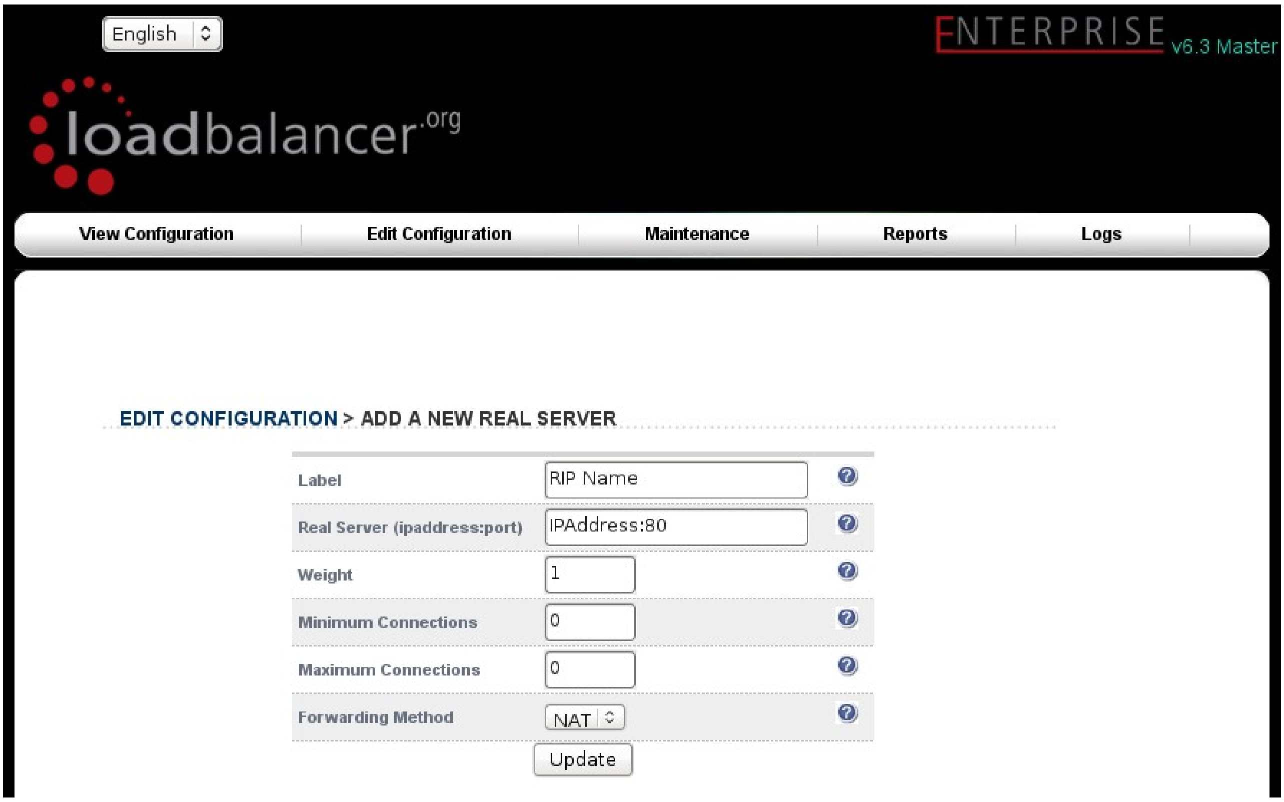
Label	VIP Name
Virtual Server (ipaddress:port)	10.0.0.20:80
Persistent	no

- 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 *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 real server**.
- You just need to give the IP Address and Port number of your web server.



EDIT CONFIGURATION > ADD A NEW REAL SERVER

Label	RIP Name
Real Server (ipaddress:port)	IPAddress:80
Weight	1
Minimum Connections	0
Maximum Connections	0
Forwarding Method	NAT

Update

- 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.

Advanced DR considerations

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

Solving the ARP problem

Each web server needs a loop back 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.

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 the quick start section at the start of the administration manual.

Loopback method 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.

Resolving ARP issues for Solaris

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

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

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

Resolving ARP issues 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.

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 IPs.

To solve this:

1) You need to add a line to the *rc.firewall* script on the load balancer to allow all outgoing traffic from the internal network to be MASQUERADED.

i.e.

```
$INT_SUBNET=”192.168.1.0/255.255.255.0”
iptables -t nat -A POSTROUTING -s INT_SUBNET -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.

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
```

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

Explaining the RIP & VIP in NAT mode

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

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

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

Protocol	VIP	Port	RIP	Port
TCP	10.0.0.20	80	192.168.2.50	80

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

Packet rewriting works as follows.

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

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

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

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

Replies get back to the load balancer as:

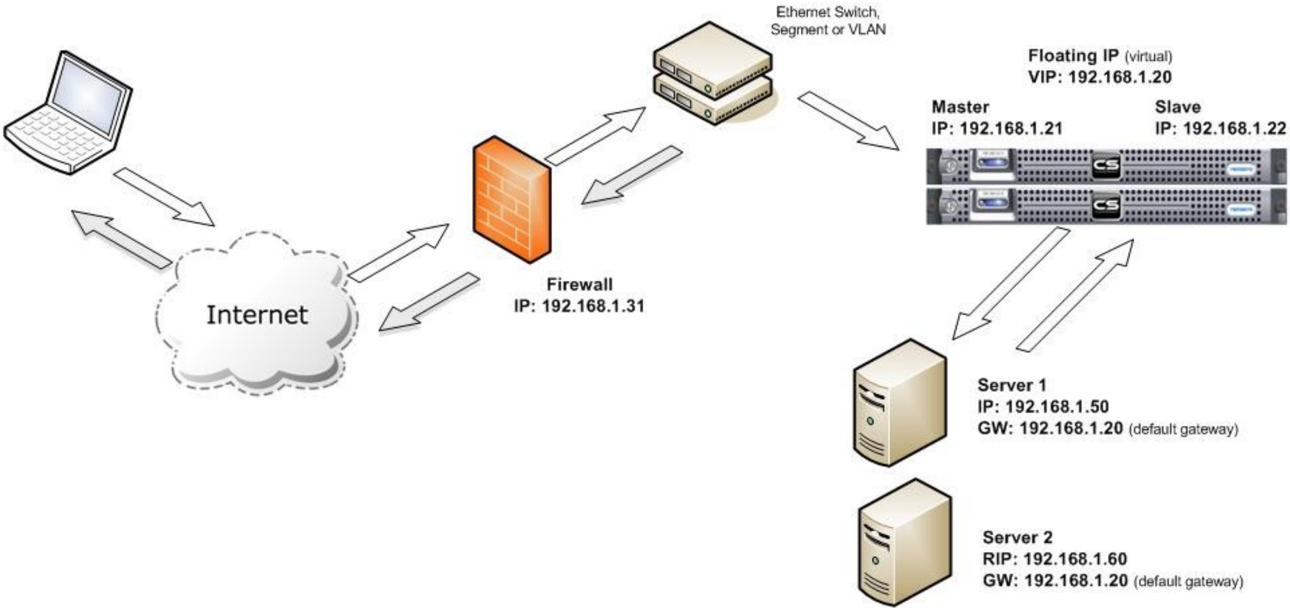
SOURCE 192.168.1.50:80 DEST x.x.x.x:3456

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

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

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

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



Notes:

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

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

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
```

NB. Replace 192.168.1.0 with your local subnet address.

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

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

NB. Replace 192.168.1.21 with your load balancer gateway

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

Advanced 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 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.

Example 3: layer 7 configuration one-arm SNAT mode

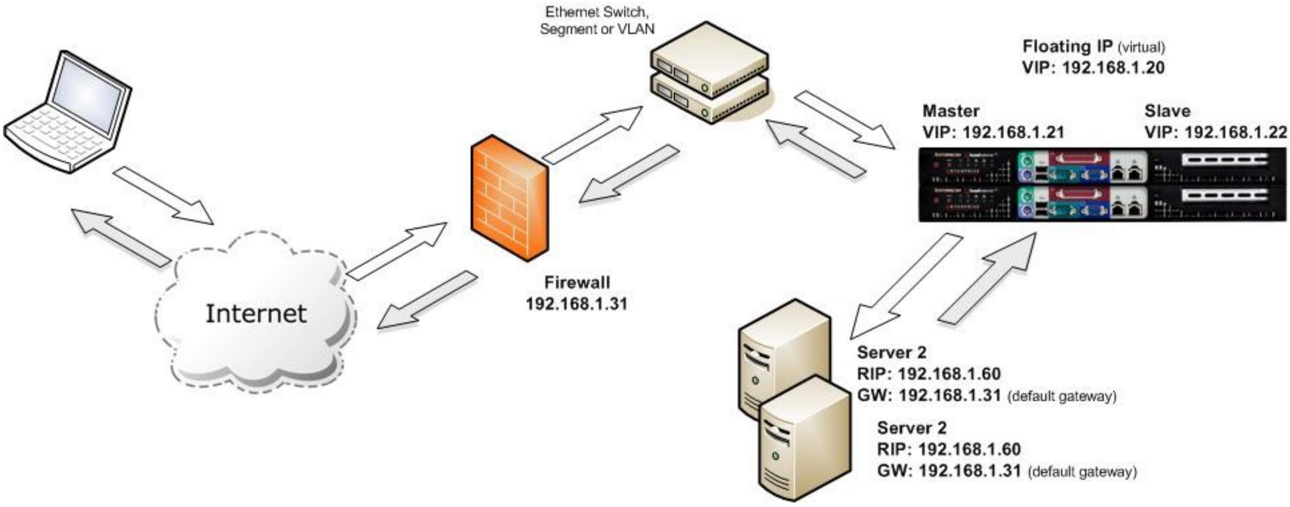
For this example we are going to assume that the e-commerce application does not support persistence. We are going to decrypt the SSL traffic on the load balancer, insert or read the session cookies as required and pass the traffic the the real servers in plain unencrypted HTTP.

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

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

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

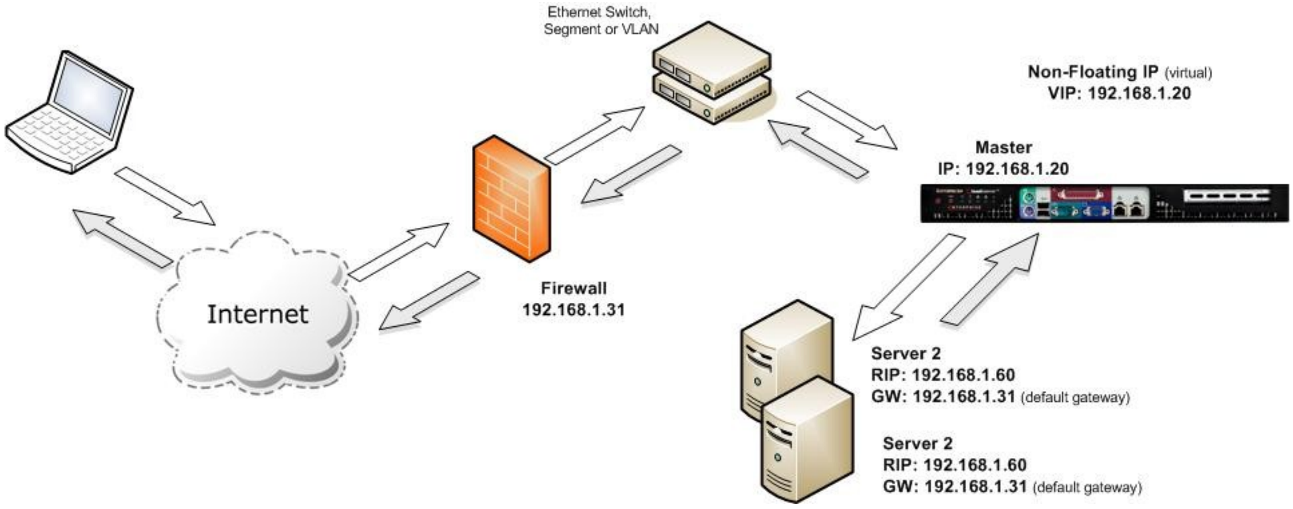
Network diagram for layer 7 SNAT mode (clustered pair)



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

NB. You can configure your web server logs to parse the X-Forwarded-For header to find the client source IP.

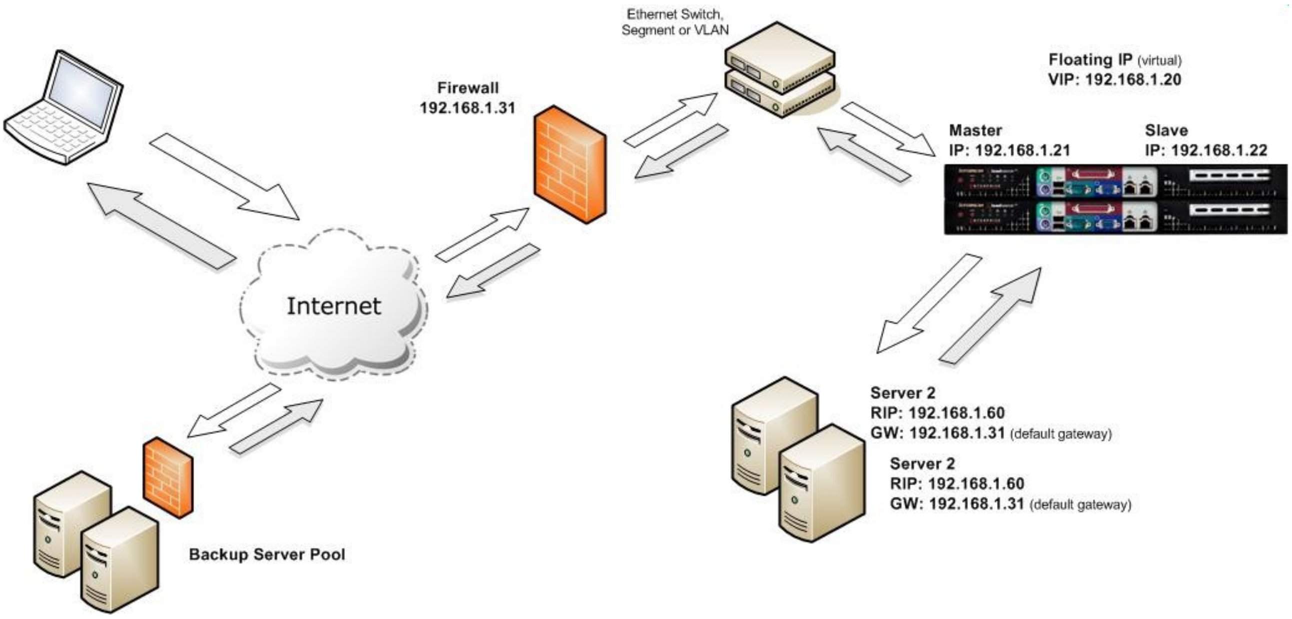
Network diagram for layer 7 SNAT mode (single unit)



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.

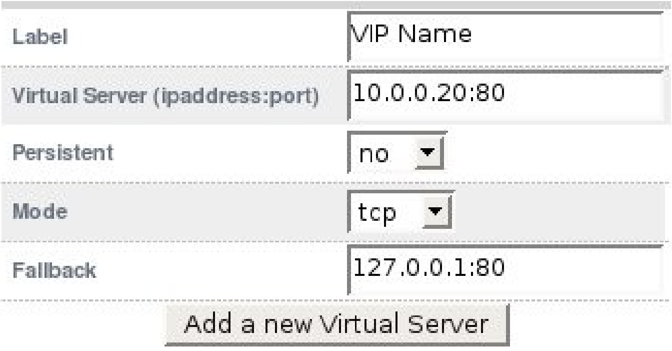
Network Diagram for layer 7 SNAT mode (off site backup)



Because SNAT 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.



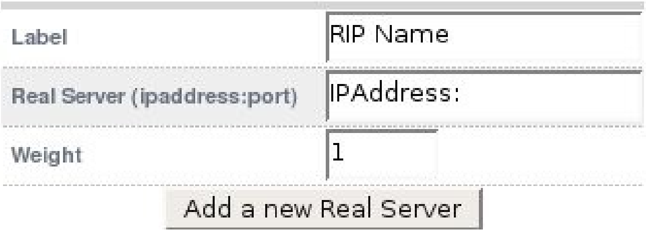
Label	VIP Name
Virtual Server (ipaddress:port)	10.0.0.20:80
Persistent	no
Mode	tcp
Fallback	127.0.0.1:80

- Configure the Virtual Server as 192.168.1.31:80
- Set persistence to 'Yes'.
- Set Mode to 'http' (cookie)
- Set the Fallback 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.
- You will see the following message:
  "192.168.1.31 will be added as an alias. *NB. You may need to restart heartbeat for this change to take effect."*
- This is the Floating Virtual IP being added, you only need to restart heartbeat if this is your very first Floating Virtual IP or if you are not using a clustered pair. *Clustered pairs can bring up the Floating Virtual IP(s) automatically.*

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 IP*Address:Port* of your web server and specify a relative weight.



Label	RIP Name
Real Server (ipaddress:port)	IPAddress:
Weight	1

- Add as many real servers as required.

- You have now finished configuration of the load balancer.

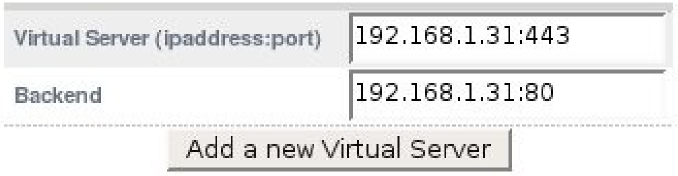
IMPORTANT: The Label is used as the cookie so make sure it is different for each server.

IMPORTANT: You must restart the HA-Proxy service in order to activate the changes i.e. Maintenance > Restart HAProxy.

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

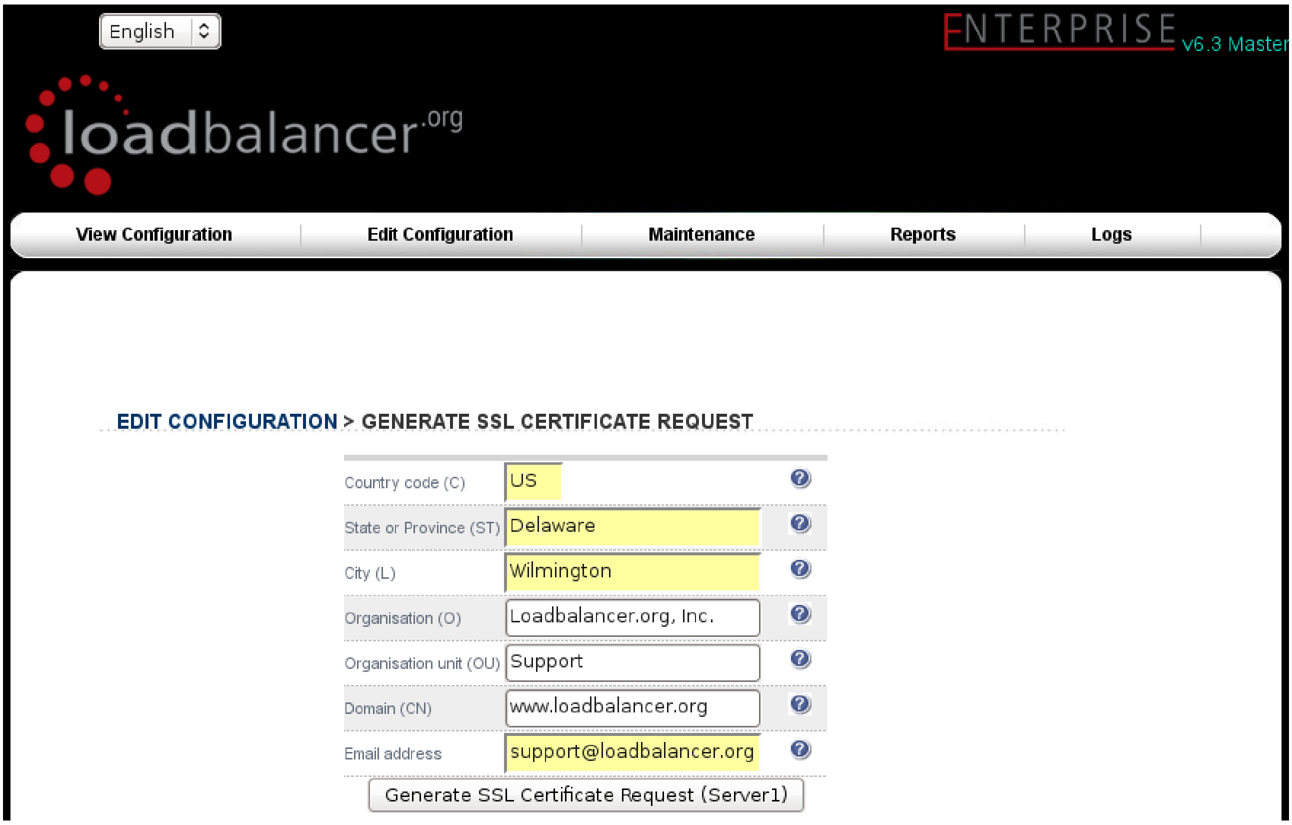
- Configure the Virtual Server as 192.168.1.20:443
- Configure the Backend as 192.168.1.20:80
- Click the button to add the new Virtual Server to the Pound configuration file.

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



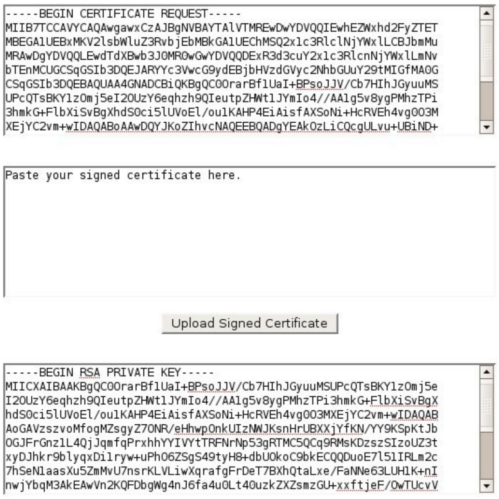
EDIT CONFIGURATION > GENERATE SSL CERTIFICATE REQUEST

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

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.



```
-----BEGIN CERTIFICATE REQUEST-----
MIIB7TCCAVYCAQAwgawxCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhEZWxhd2FyZTET
MBEGA1UEBxMKV2lsbWluZ3RvbjEbMBkGA1UEChMSQ2xlc3RlclNjYWxlLCBJbmMu
MRAwDgYDVQQLEwdTdXBwb3J0MROwGwYDVQQDExR3d3cuY2xlc3RlcnNjYWxlLmNv
bTEnMCUGCSqGSIb3DQEJARYYc3VwcG9ydEBjbHVzdGVyc2NhbGUuY29tMIGfMA0G
CSqGSIb3DQEBAQUAA4GNADCBiQKBgQC0OrarBflUaI+BPsoJJV/Cb7HIhJGyuuMS
UPcQTsBKYlzOmj5eI2OUzY6eqhzh9QIeutpZHWtlJYmIo4//AAlg5v8ygPMhzTPi
3hmkG+FlbXiSvBgXhdS0ci5lUVoEl/oulKAHP4EiAisfAXSoNi+HcRVEh4vg0O3M
XEjYC2vm+wIDAQABoAAwDQYJKoZIhvcNAQEEBQADgYEAkOzLiCQcgULvu+UBiND+
```

```
Paste your signed certificate here.
```

```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC0OrarBflUaI+BPsoJJV/Cb7HIhJGyuuMSUPcQTsBKYlzOmj5e
I2OUzY6eqhzh9QIeutpZHWtlJYmIo4//AAlg5v8ygPMhzTPi3hmkG+FlbXiSvBgX
hdS0ci5lUVoEl/oulKAHP4EiAisfAXSoNi+HcRVEh4vg0O3MXEjYC2vm+wIDAQAB
AoGAVzszvoMfogMZsgyZ7ONR/eHhwpOnkUIzNWJKsnHrUBXXjYfKN/YY9KSpKtJb
OGJFrGnzlL4QjJqmfqPrxhhYYIVYtTRFNrNp53gRTMC5QCq9RMsKDzszSIzoUZ3t
xyDJhkr9blyqxDilryw+uPhO6ZSgS49tyH8+dbUOkoC9bkECQQDuoE7l5lIRLm2c
7hSeNlaasXu5ZmMvU7nsrKLVLiwXqrafgFrDeT7BXhQtaLxe/FaNNe63LUHlK+nI
nwjYbqM3AkEAwVn2KQFDbgWg4nJ6fa4u0Lt40uzkZXZsmzGU+xxftjeF/OwTUcvV
```

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

ℹ If you need to convert, upload, modify or chain a certificate look at the Advanced SSL Considerations section.

Further administration functionality.

The Loadbalancer.org appliance is a standard Armari based server running the GNU/Linux operating system with a custom kernel configured for load balancing. Loadbalancer.org appliances should always be deployed in a fail over configuration for maximum reliability.

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

Each load balancer must, initially, be individually configured. Once this is done all configuration takes place on the master load balancer and 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 load balancer.

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

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 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**

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 to following tools:

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

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

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

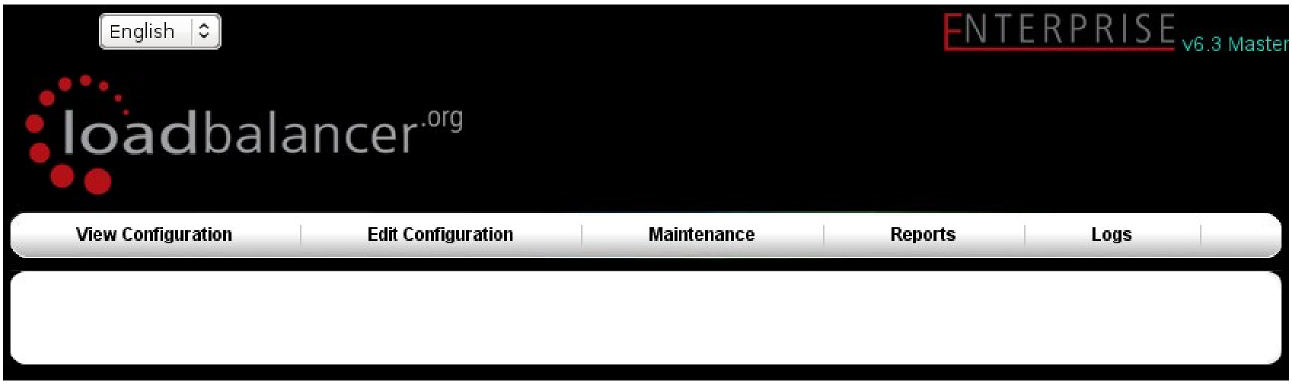
The Web based Administration interface uses a different set of user accounts and passwords based on the simple .htaccess files. This allows you to set up users in three groups, configuration, maintenance and reports.

To access the web based administration interface use:

http://ApplianceIPaddress:9080/lbadmin/

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

Which should bring up the following screen:



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

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

Web interface functionality in detail

This section of the manual describes each function in the web interface and explains the intended functionality.

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

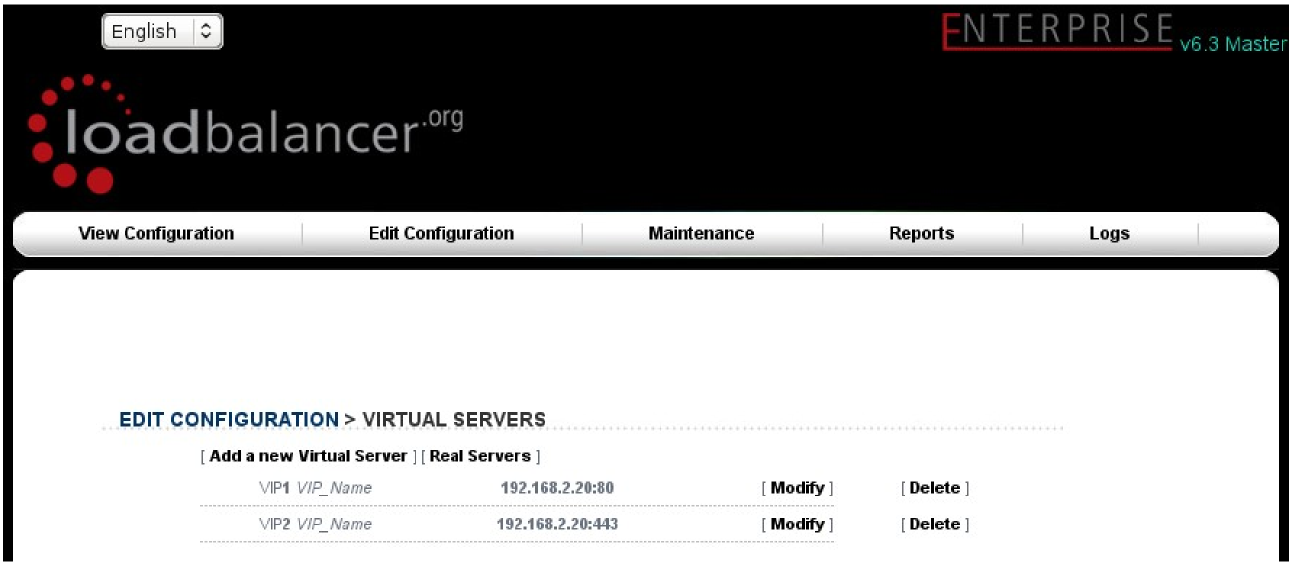
This menu option allows you to add, remove or modify virtual servers from your cluster.

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.*

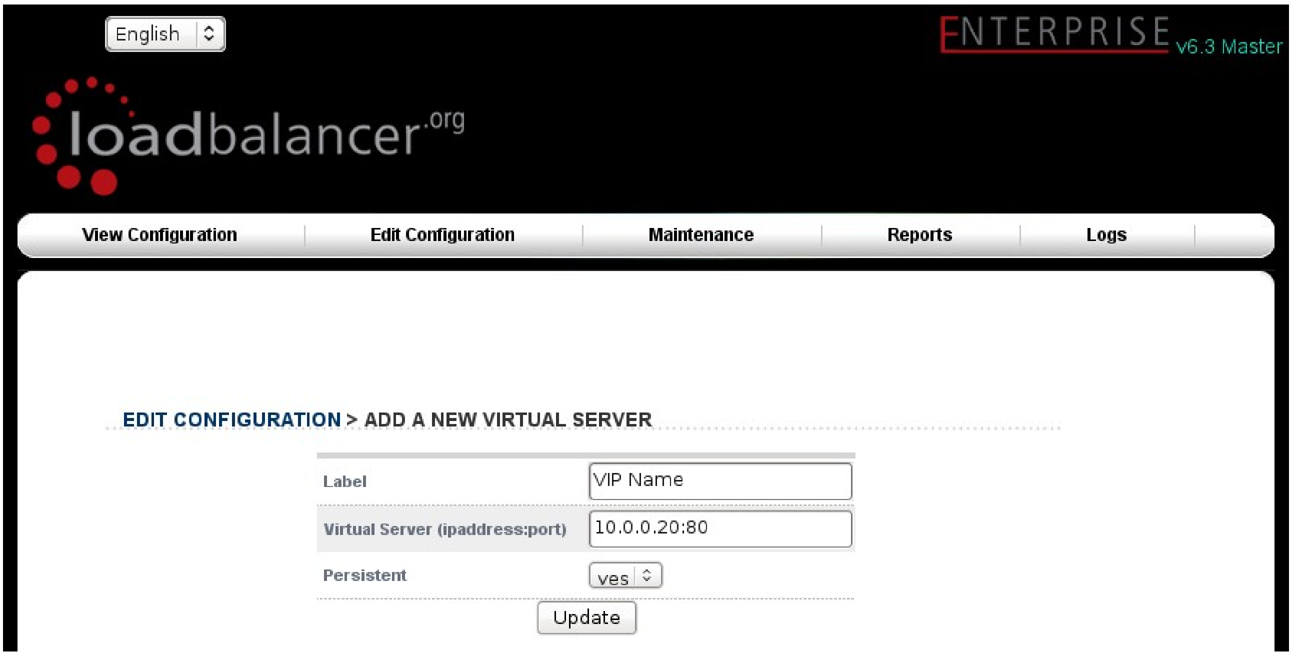


EDIT CONFIGURATION > VIRTUAL SERVERS

[Add a new Virtual Server] [Real Servers]

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.



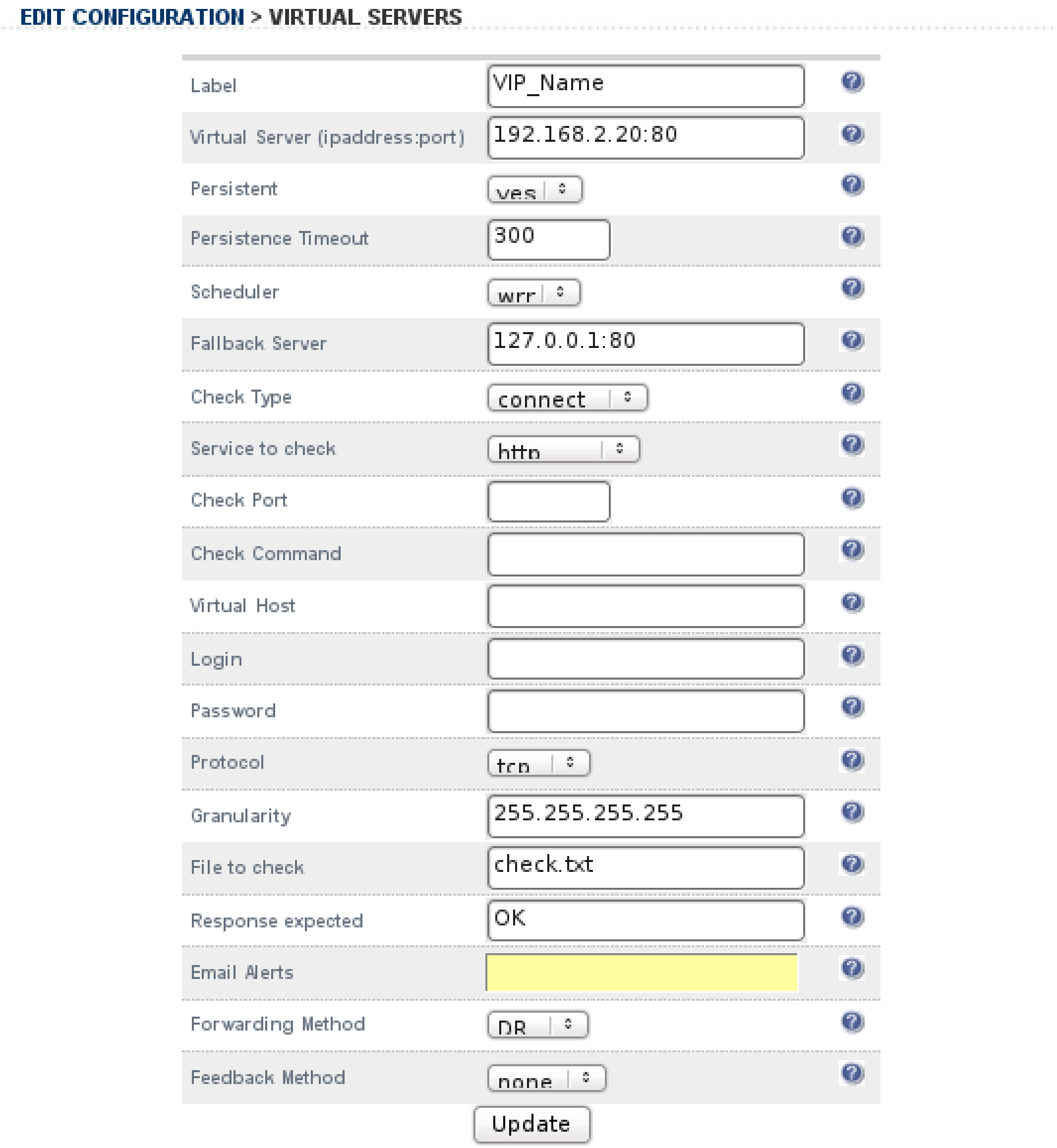
EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER

Label	VIP Name
Virtual Server (ipaddress:port)	10.0.0.20:80
Persistent	yes

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.
- When using a clustered pair of load balancer units the Floating IP will activate instantly.
- When using a single load balancer and a new Floating IP is created you will need to use *Maintenance > Restart Heartbeat* in order to activate the address.
- Just check '*View Configuration > Network Configuration*' to ensure that the Floating IP address has been activated correctly.

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	VIP_Name
Virtual Server (ipaddress:port)	192.168.2.20:80
Persistent	yes
Persistence Timeout	300
Scheduler	wrr
Fallback Server	127.0.0.1:80
Check Type	connect
Service to check	http
Check Port	
Check Command	
Virtual Host	
Login	
Password	
Protocol	tcp
Granularity	255.255.255.255
File to check	check.txt
Response expected	OK
Email Alerts	
Forwarding Method	DR
Feedback Method	none

Here you can modify :

- The virtual IP address and port.
- Whether you want sticky connections
- How long should the connections persist in seconds (300 should be fine)
- What type of scheduler to use :
  - WLC – Weighted Least Connection
  - 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 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
  - HTTPS
  - FTP
  - IMAP
  - POP
  - 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
- 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
  - Gate – The default Direct Server Return
  - IPIP – IP encapsulation
  - Masq - 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

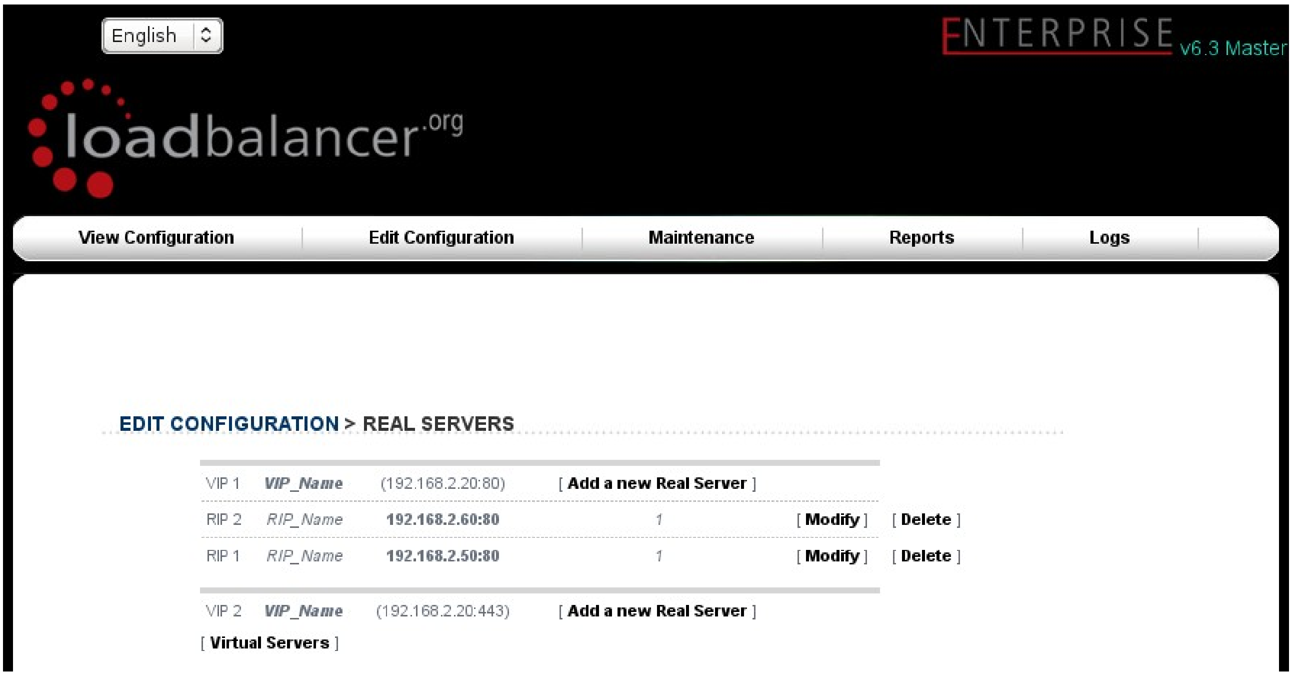
This menu option allows you to add, remove or modify Real Servers from your cluster.

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.

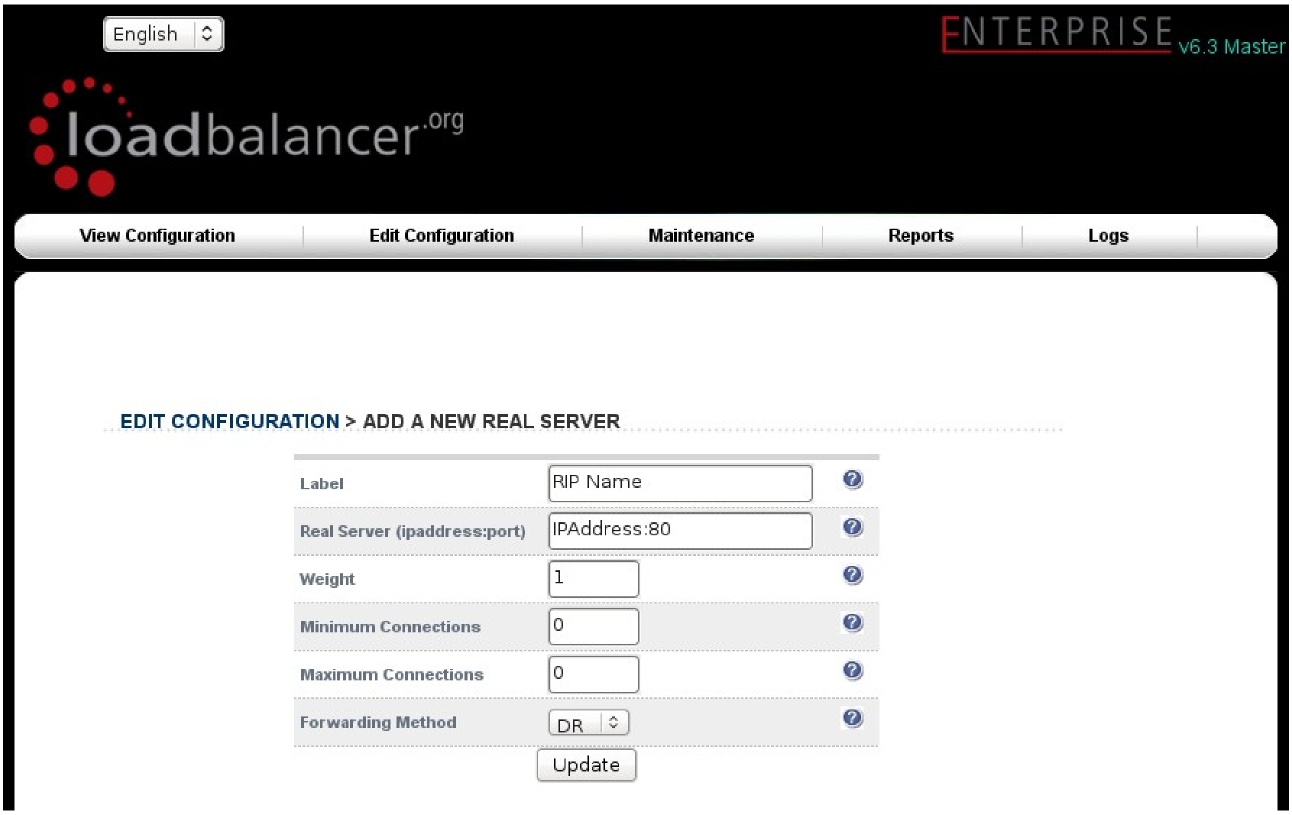
From the overview you can see each web server in the cluster, the IP address, port number and the requested relative weight (0 is off line).



EDIT CONFIGURATION > REAL SERVERS

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

[**Virtual Servers**]



EDIT CONFIGURATION > ADD A NEW REAL SERVER

Label	RIP Name
Real Server (ipaddress:port)	IPAddress:80
Weight	1
Minimum Connections	0
Maximum Connections	0
Forwarding Method	DR

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, it's possible you would increase the weight of the Xeon so that it took more of the load. In general most web servers are so fast these days you tend to find an even distribution of page processing power.

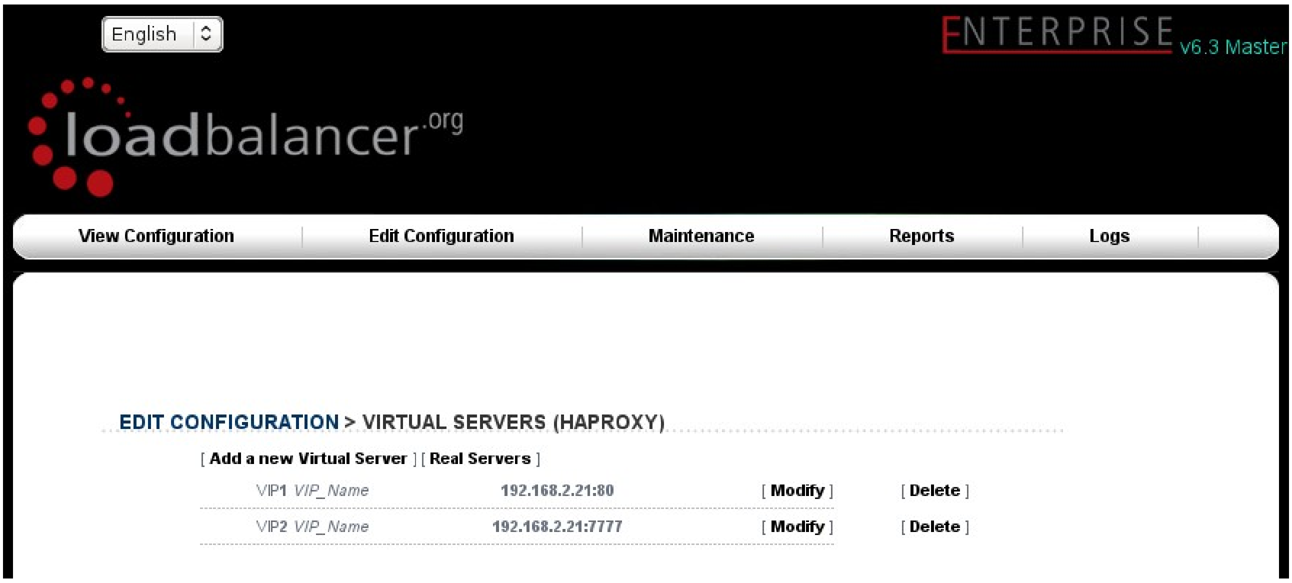
NB. If you take a server offline from the maintenance page and then bring it back online, the weight will be set back to one, just click on the 1 in order to link through to the 'modify real server screen' and change the weight back to the desired amount.

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.



EDIT CONFIGURATION > VIRTUAL SERVERS (HAPROXY)

[Add a new Virtual Server] [Real Servers]

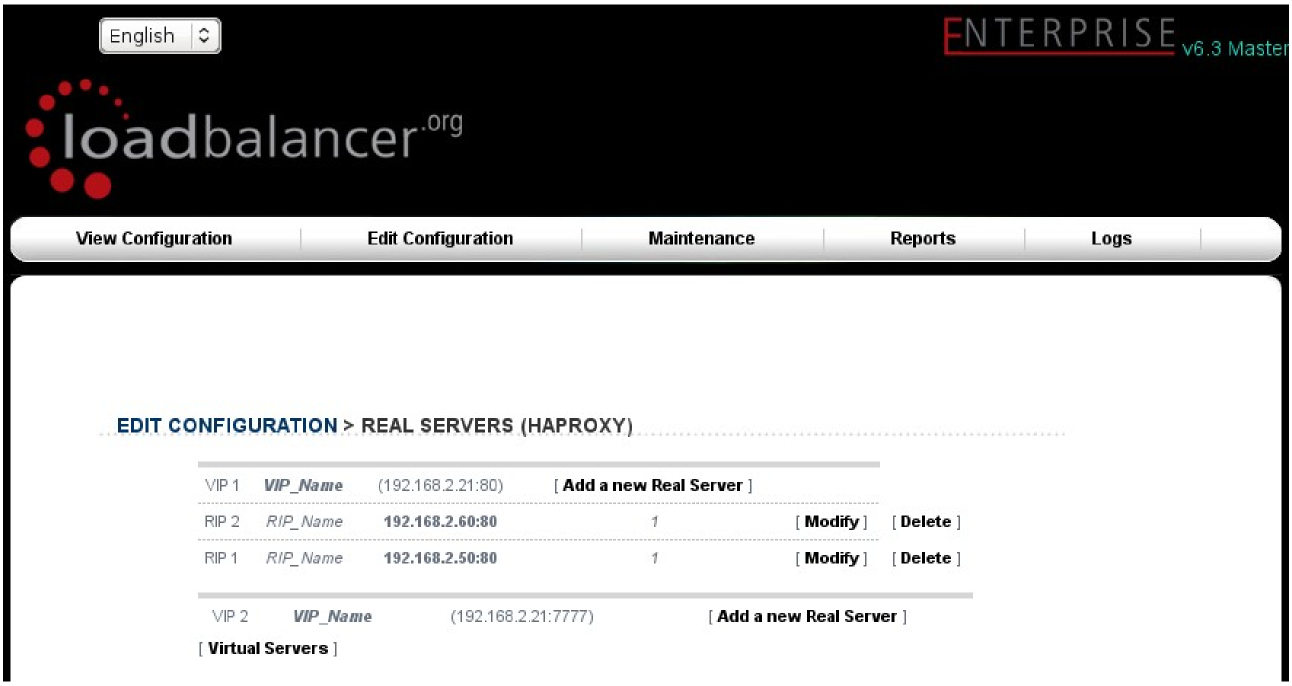
VIP1 *VIP_Name*	192.168.2.21:80	[Modify]	[Delete]
VIP2 *VIP_Name*	192.168.2.21:7777	[Modify]	[Delete]

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.

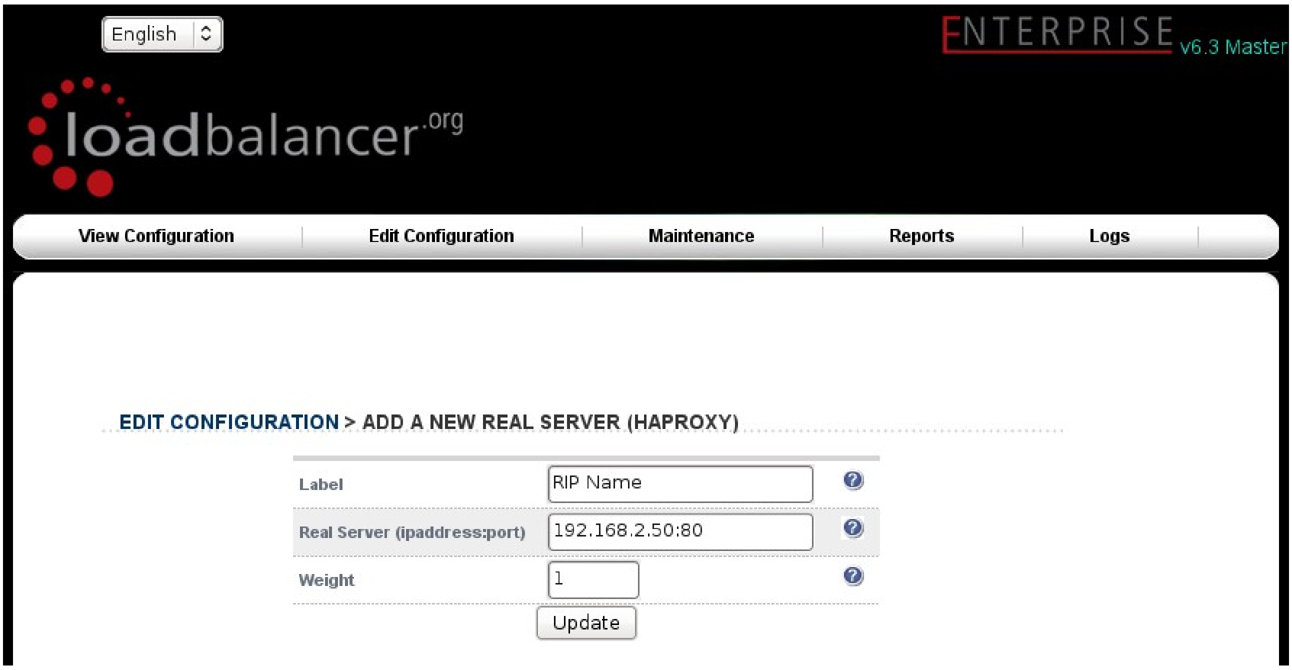


EDIT CONFIGURATION > REAL SERVERS (HAPROXY)

VIP 1	*VIP_Name*	(192.168.2.21:80)	[**Add a new Real Server**]		
RIP 2	*RIP_Name*	**192.168.2.60:80**	*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.



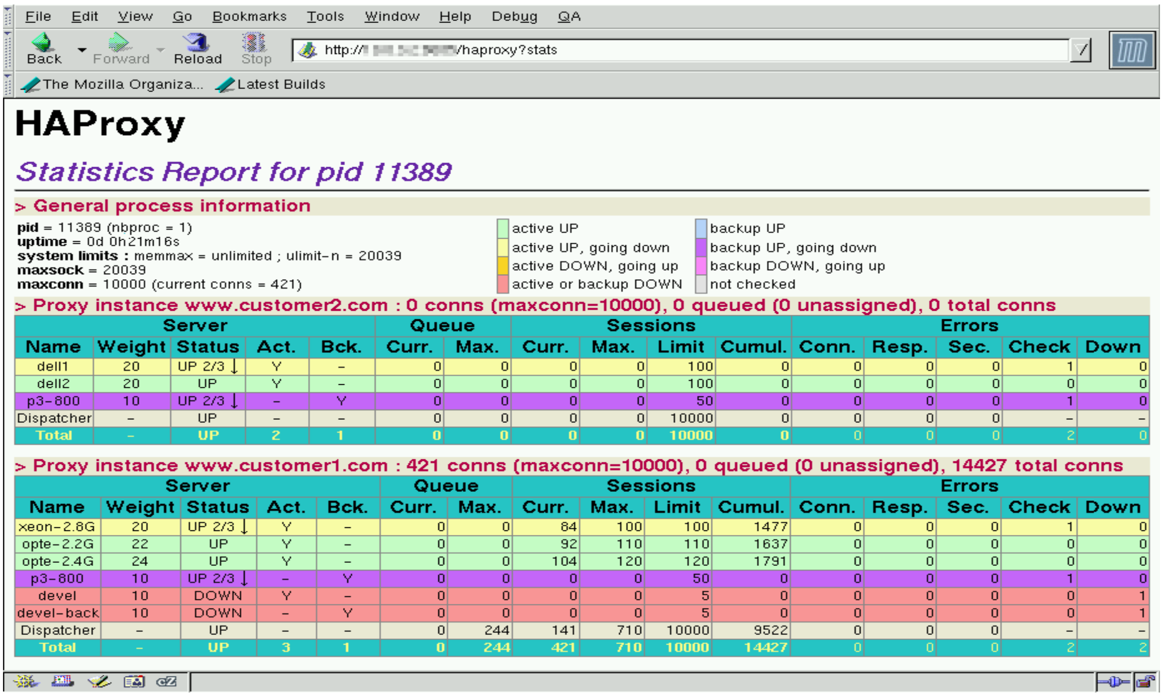
EDIT CONFIGURATION > ADD A NEW REAL SERVER (HAPROXY)

Label	RIP Name
Real Server (ipaddress:port)	192.168.2.50:80
Weight	1

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.

Enable web based statistics (HAProxy)

HAProxy has its own built in method for reporting statistics, server utilization and server health status. This is not enabled by default and you need to manually add it to your list of Virtual Servers.



http://[redacted]/haproxy?stats

HAProxy

Statistics Report for pid 11389

> General process information

pid = 11389 (nbproc = 1)
uptime = 0d 0h21m16s
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

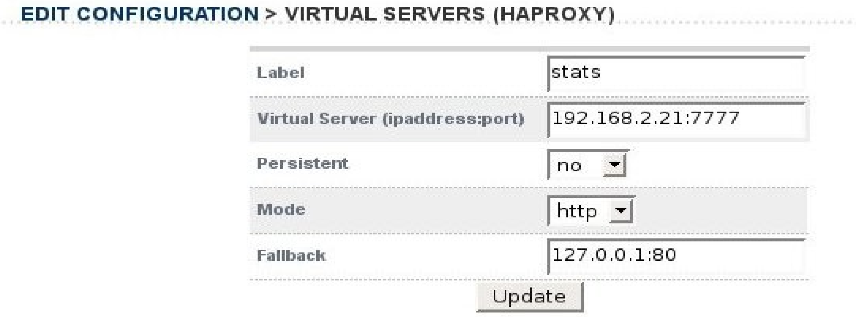
Server					Queue		Sessions				Errors				
Name	Weight	Status	Act.	Bck.	Curr.	Max.	Curr.	Max.	Limit	Cumul.	Conn.	Resp.	Sec.	Check	Down
dell1	20	UP 2/3 ↓	Y	-	0	0	0	0	100	0	0	0	0	1	0
dell2	20	UP	Y	-	0	0	0	0	100	0	0	0	0	0	0
p3-800	10	UP 2/3 ↓	-	Y	0	0	0	0	50	0	0	0	0	1	0
Dispatcher	-	UP	-	-	0	0	0	0	10000	0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

To enable the HAProxy web based statistics just add a new Virtual Server (HAProxy) VIP that is called '**stats**'.
NB. The name is important.

We advise that you choose the administration IP address for your system and port 7777, but you can change as required.



EDIT CONFIGURATION > VIRTUAL SERVERS (HAPROXY)

Label	stats
Virtual Server (ipaddress:port)	192.168.2.21:7777
Persistent	no
Mode	http
Fallback	127.0.0.1:80

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.

Once HAProxy has successfully restarted just use a web browser and point it to the VIP you have chosen i.e. http://192.168.1.21:7777/.

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)

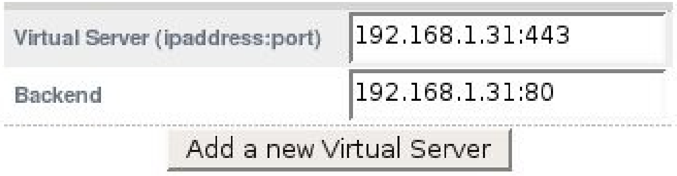
In order for the load balancer to work the box must physically own the Virtual IP address that the clients are accessing before they get re-directed to a real server in the cluster. The Floating Virtual IP(s) are controlled by heartbeat to ensure that only one of the load balancers (normally the master)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

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



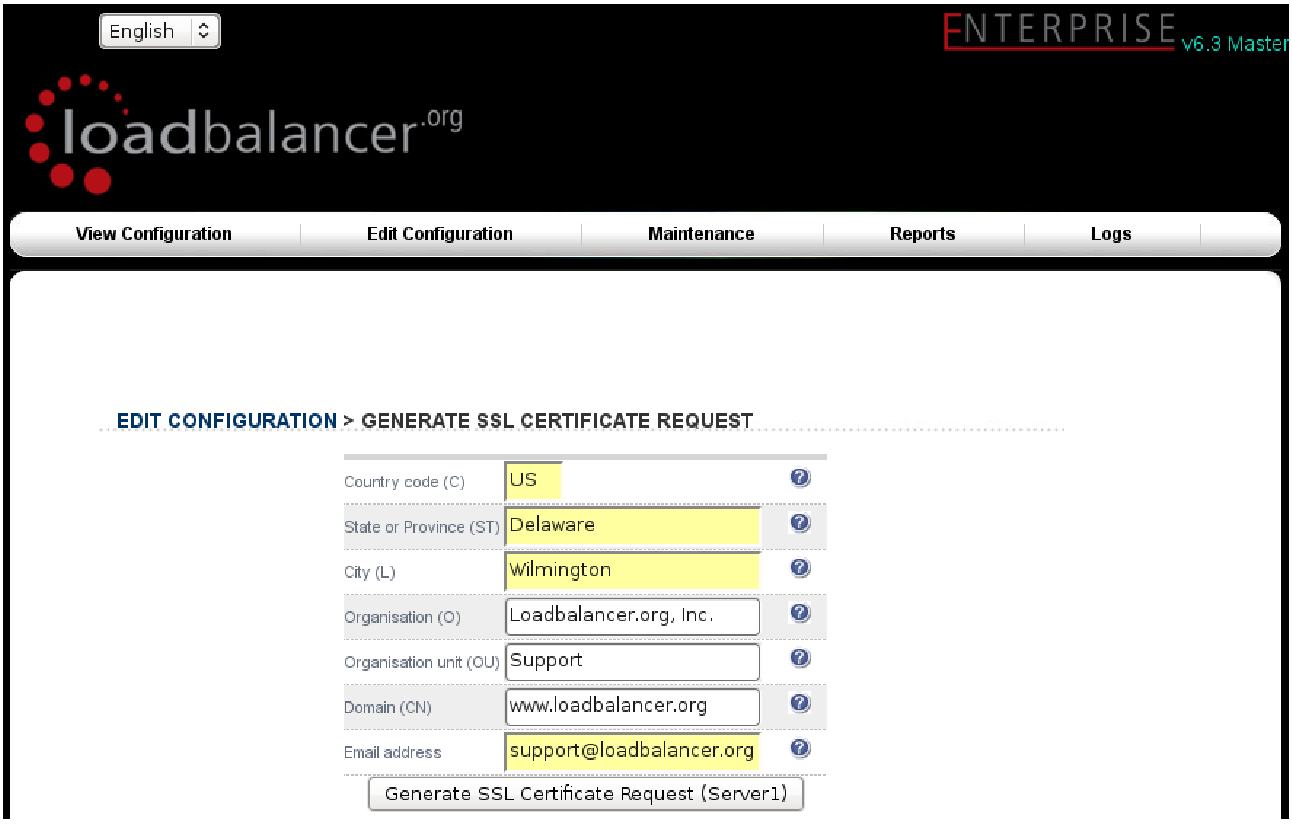
Virtual Server (ipaddress:port)	192.168.1.31:443
Backend	192.168.1.31:80

- Configure the Virtual Server as 192.168.1.31:443
- Configure the Backend 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. Maintained > 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.



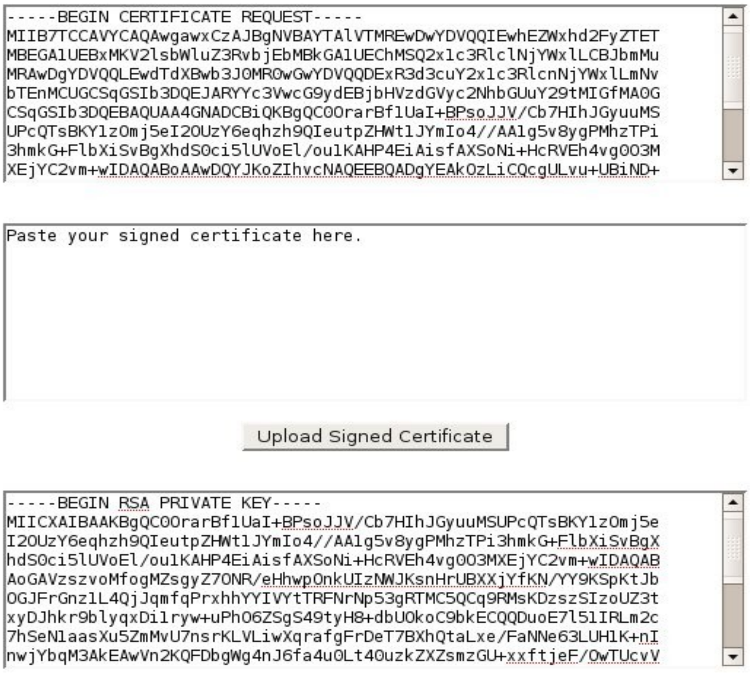
EDIT CONFIGURATION > GENERATE SSL CERTIFICATE REQUEST

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

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.



```
-----BEGIN CERTIFICATE REQUEST-----
MIIB7TCCAVYCAQAwgawxCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhEZWxhd2FyZTET
MBEGA1UEBxMKV2lsbWluZ3RvbjEbMBkGA1UEChMSQ2xlc3RlclNjYWxlLCBJbmMu
MRAwDgYDVQQLEwdTdXBwb3J0MROwGwYDVQQDExR3d3cuY2xlc3RlcnNjYWxlLmNv
bTEnMCUGCSqGSIb3DQEJARYYc3VwcG9ydEBjbHVzdGVyc2NhbGUuY29tMIGfMA0G
CSqGSIb3DQEBAQUAA4GNADCBiQKBgQC0OrarBflUaI+BPsoJJV/Cb7HIhJGyuuMS
UPcQTsBKY1zOmj5eI2OUzY6eqhzh9QIeutpZHWtlJYmIo4//AAlg5v8ygPMhzTPi
3hmkG+FlbXiSvBgXhdS0ci5lUVoEl/oulKAHP4EiAisfAXSoNi+HcRVEh4vg0O3M
XEjYC2vm+wIDAQABoAAwDQYJKoZIhvcNAQEEBQADgYEAkOzLiCQcgULvu+UBiND+
```

```
Paste your signed certificate here.
```

```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC0OrarBflUaI+BPsoJJV/Cb7HIhJGyuuMSUPcQTsBKY1zOmj5e
I2OUzY6eqhzh9QIeutpZHWtlJYmIo4//AAlg5v8ygPMhzTPi3hmkG+FlbXiSvBgX
hdS0ci5lUVoEl/oulKAHP4EiAisfAXSoNi+HcRVEh4vg0O3MXEjYC2vm+wIDAQAB
AoGAVzszvoMfogMZsgyZ7ONR/eHhwpOnkUIzNWJKsnHrUBXXjYfKN/YY9KSpKtJb
OGJFrGnzlL4QjJqmfqPrxhhYYIVYtTRFNrNp53gRTMC5QCq9RMsKDzszSIzoUZ3t
xyDJhkr9blyqxDilryw+uPhO6ZSgS49tyH8+dbUOkoC9bkECQQDuoE7l5lIRLm2c
7hSeNlaasXu5ZmMvU7nsrKLVLiwXqrafgFrDeT7BXhQtaLxe/FaNNe63LUHlK+nI
nwjYbqM3AkEAwVn2KQFDbgWg4nJ6fa4u0Lt40uzkZXZsmzGU+xxftjeF/OwTUcvV
```

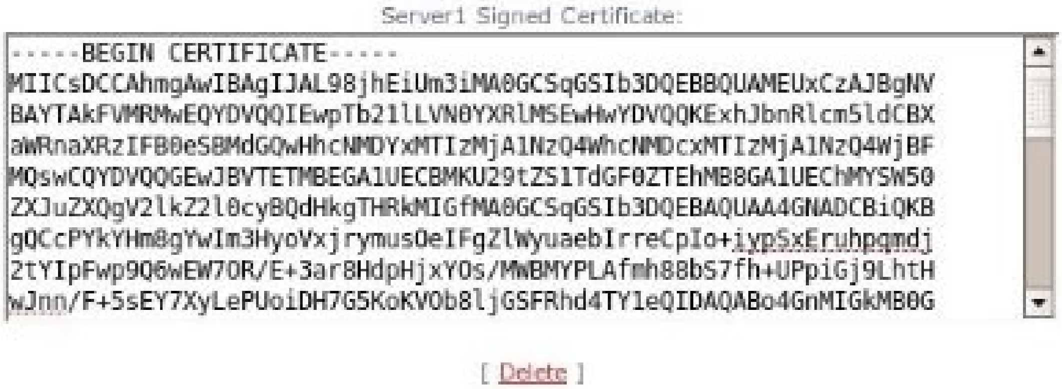
Advanced SSL considerations

Adding an Intermediate key to the certificate chain

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

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

When you select *Manage this SSL Certificate* on a pre-configured certificate it will show a copy of the full signed PEM file.



```
-----BEGIN CERTIFICATE-----
MIICsDCCAhmgAwIBAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIEwpTb21lLVN0YXRlMSEwHwYDVQQKExhJbnRlcm5ldCBX
aWRnaXRzIFB0eSBMdGQwHhcNMDYxMTIzMjA1NzQ4WhcNMDcxMTIzMjA1NzQ4WjBF
MQswCQYDVQQGEwJBVTETMBEGA1UECBMKU29tZS1TdGF0ZTEhMB8GA1UEChMYSW50
ZXJuZXQgV2lkZ2l0cyBQdHkgTHRkMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCcPYkYHm8gYwIm3HyoVxjrymusOeIFgZlWyuaebIrreCpIo+iypSxEruhpqmdj
2tYIpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPLAfmh88bS7fh+UPpiGj9LhtH
wJnn/F+5sEY7XyLePUoiDH7G5KoKVOb8ljGSFRhd4TY1eQIDAQABo4GnMIGkMB0G
```

Select the whole of the text and paste it into a text editor such as notepad, not Word!

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-----
MIICsDCCAhmgAwIBAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIEwpTb21lLVN0YXRlMSEwHwYDVQQKExhJbnRlcm5ldCBX
kU6DJupvN6U6PRi7+zcKqd8wUiY8+3CyYKHtJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBNkyHYbQTvU8MeR3iIhe2fw+qVE2pgxWYWaGm8QwTsxQKgbxiG
SXUWIWb0+k2j2L1z2PszFxwCIwQ=
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwIm3HyoVxjrymusOeIFgZlWyuaebIrreCpIo+iy
pSxEruhpqmdj2tYIpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPLAfmh88bS7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgRUQPiLosSmUCZ/9Ec
SlqR7x/WAQUnFKVxQAMDatpeXSp3FGgXF+mpffusjEw=
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIEwDCCBCmgAwIBAgIQY7GlzcWfeIAdoGNs+XVGezANBgkqhkiG9w0BAQUFADCB
Ai9TXvRlcD5q0mB+nyK9fB2aBzOiaiHSiIWzAJeQjuqA+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.

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
```

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.

Advanced

Execute a shell command

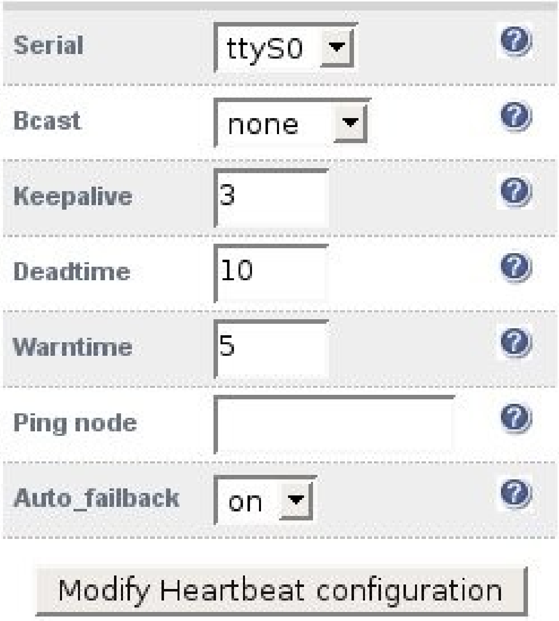
This allows you to remotely execute a shell command as a root user. Useful if you accidentally kill your SSH server or something.

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.



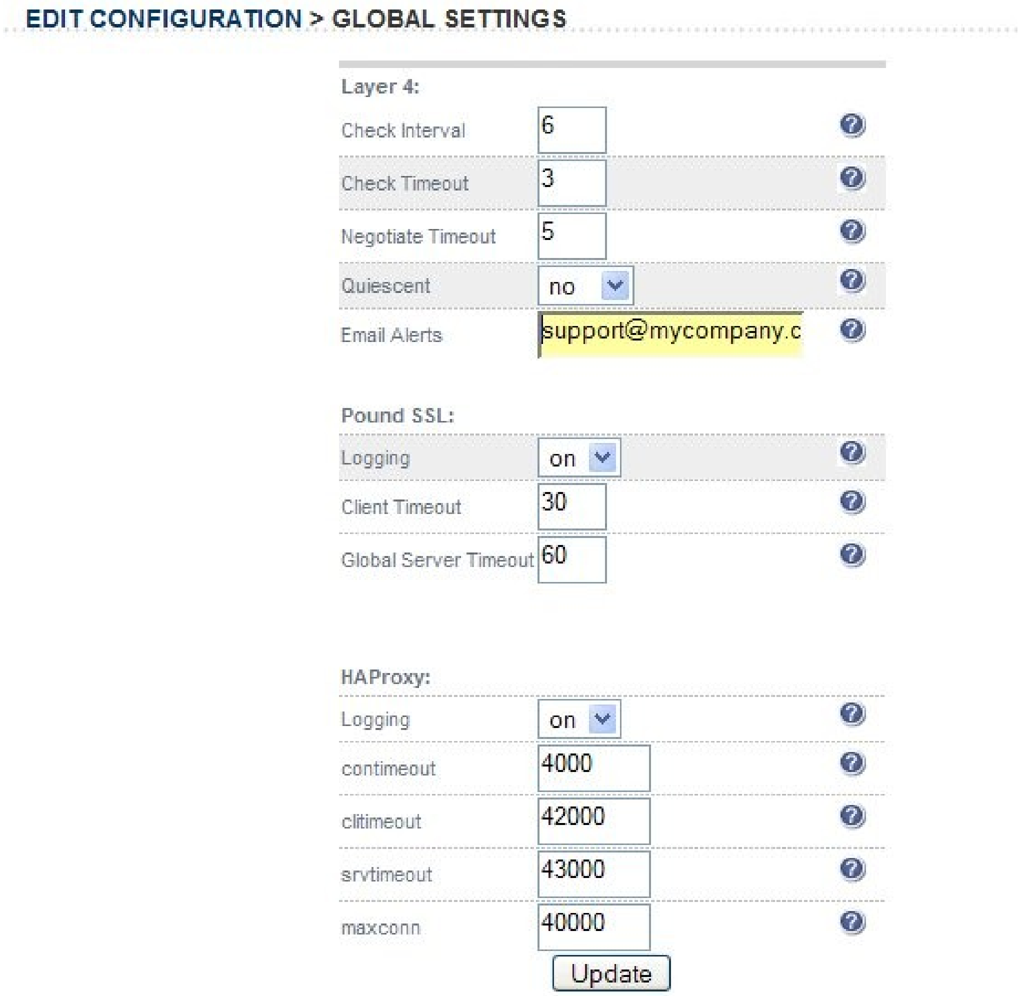
Serial	ttyS0
Bcast	none
Keepalive	3
Deadtime	10
Warntime	5
Ping node	
Auto_failback	on

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

Modify 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	6
Check Timeout	3
Negotiate Timeout	5
Quiescent	no
Email Alerts	support@mycompany.c

Pound SSL:

Logging	on
Client Timeout	30
Global Server Timeout	60

HAProxy:

Logging	on
contimeout	4000
clitimeout	42000
srvtimeout	43000
maxconn	40000

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

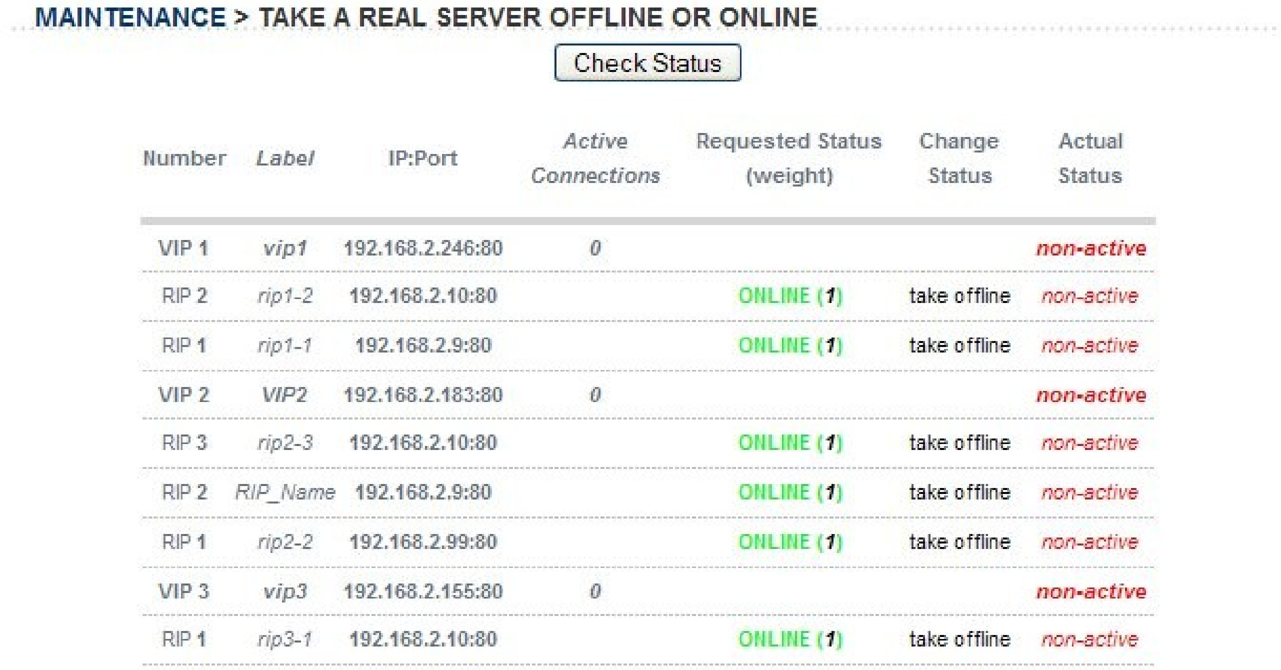
The Pound SSL & HAProxy logs will be in */var/log/poundssl* and */var/log/haproxy* respectively when enabled.

Maintenance

Maintain Real Servers

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.



MAINTENANCE > TAKE A REAL SERVER OFFLINE OR ONLINE

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

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 Idirectord
- 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.

Restore Configuration

This quickly restores the configuration from a previous local backup, use this if you have made a big mistake when re-configuring the device. In order to ensure that any changes take effect cleanly you may need to restart.

Disaster Recovery Options

This section gives you the following options :

- Restore manufacturers settings – Handy if you want to start all over again
- Download XML Config – Allows you to download all settings in one XML file
- Upload XML Config file – To upload a previous XML file and activate the settings

Services

Restart HAProxy

Any configuration changes to the Layer 7 (HAProxy) configuration, including server weights, will require a restart of 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 you will need to restart heartbeat (*on a properly configured cluster this will also force a heartbeat restart on the slave*).

NB. Adding a Floating IP address only requires a heartbeat restart on non-clustered systems.

With a simple restart the old virtual addresses may be left active so you may wish to do a full re-boot to be sure all changes are clean.

Restart Ldirectord

It is unlikely that you will ever need to use this function. It just re-loads the health check configuration file.

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 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.

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, it 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.

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 move the Apache binding port from 9080 to 80. This can be done with the console command *lbloports*

WARNING: If you are using localhost as your holding page and your web servers are offline then the local Apache 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 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
```

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.

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.

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.

Reports

The reports are broken down into real time and statistical.

The real time reports are:

- Status (Current Active and Inactive connections)
- Traffic rate per second
- Traffic Qty (Since last counter reset with 64 bit counters)
- Current Connections (with or without DNS lookup)

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.

- Status (Layer 7 HAProxy)

This is a dynamic link to the load balancers eth0:7777, created when you modify the load balancers eth0 IP Address. This takes you to a real time report showing all of the Layer 7 HAProxy instances and their current status. If this report does not appear check that you have a Layer 7 Virtual IP configured with a label of *stats* on eth0:7777. i.e.

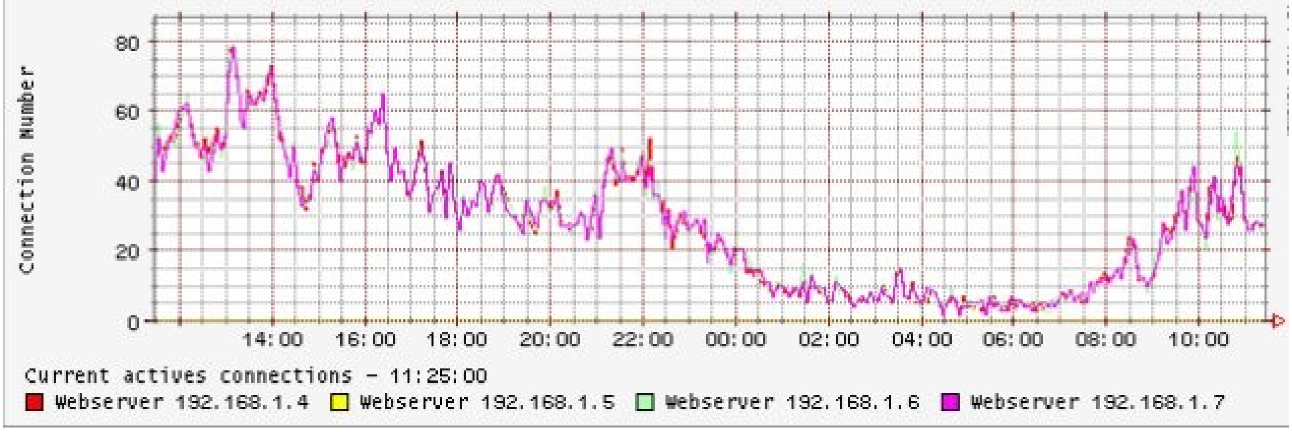
```
listen  stats 10.0.0.20:80
        stats   enable
        stats   uri /
        server  dummy 127.0.0.1:80
```

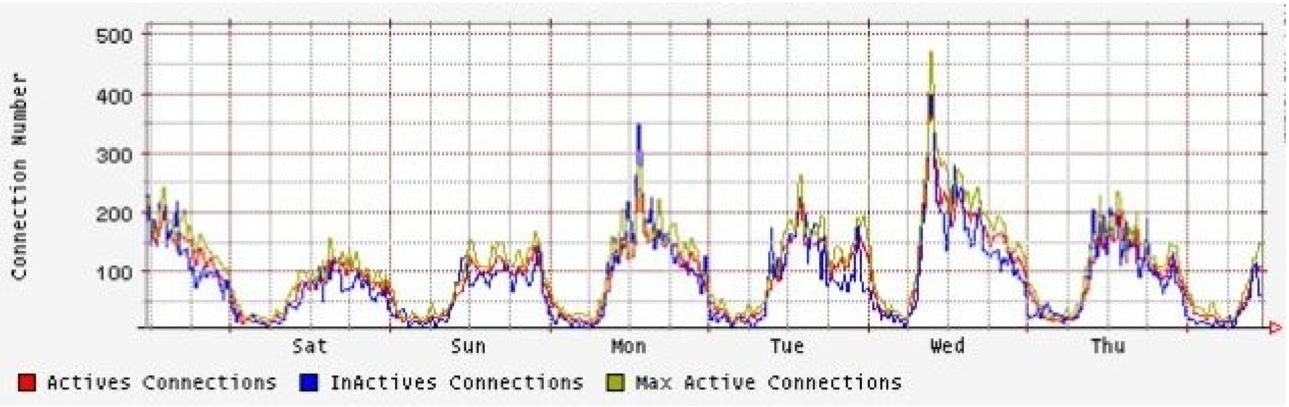
Graphical Stats Over Time

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.





Advanced Topics

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.

FTP

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

active 20,21
passive 21,high_port

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

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

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

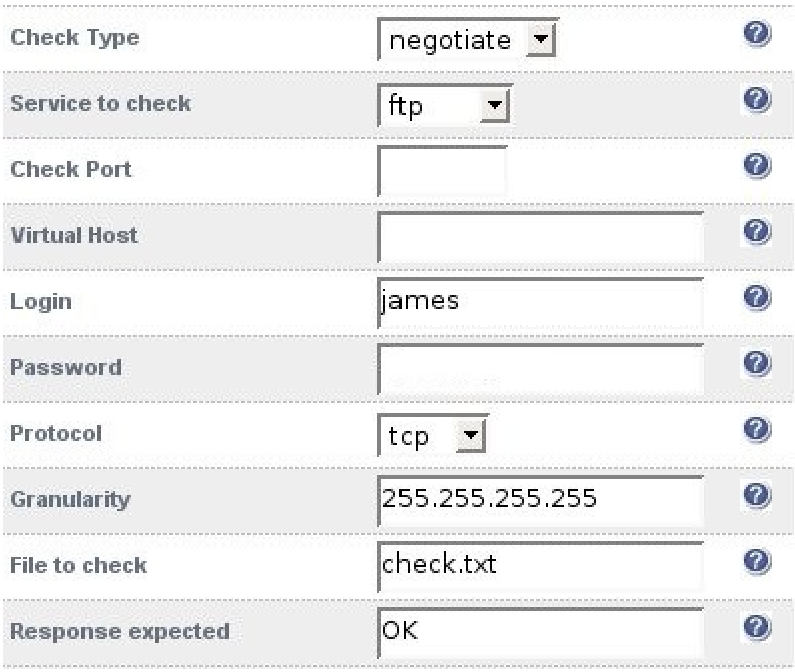
```
# This example marks groups the active FTP ports
VIP1="192.168.0.66"
# First two rule are for Active connections
 iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 21 -j MARK --set-mark 1
 iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 20 -j MARK --set-mark 1
# Third additional rule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

 iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 1024: -j MARK --set-mark 1
```

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

WARNING: *Windows Server FTP & DR mode requires the use of the firewall marks method.*

FTP Negotiate Health Check

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



Check Type	negotiate
Service to check	ftp
Check Port	
Virtual Host	
Login	james
Password	
Protocol	tcp
Granularity	255.255.255.255
File to check	check.txt
Response expected	OK

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

Terminal Server RDP

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.

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.

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 so that when a client disconnects the persistence template is cleared forcing them to re-connect to a different server.
In the unlikely event that you wish to disable this feature globally use the following commands from the console:

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

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

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.

Load balancing based on URL match

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.

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=1
```

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
```

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

You can test if everything works 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.

Loadbalancer fail over when network fails

This can be configured using *Edit Configuration* > *Heartbeat Configuration*

If you want it to detect a network failure you need to define a ping node that should be accessible from both the master and slave. A good ping node would normally be a router or gateway.

The manual process is to edit /etc/ha.d/ha.cf (on both machines)
And remove the # remark from the line: (and enter the correct ping node)

```
#ping 10.10.10.254
```

And remove the # remark from the line:

```
#respawn hacluster /usr/lib/heartbeat/ipfail
```

Then restart heartbeat on both nodes.

Now if either node detects a network failure it will swap over.
NB. The other method is to set up cross-wired high-availability bonding on the NICs.

Heartbeat over network as well as fail-over cable

This can be configured using *Edit Configuration* > *Heartbeat configuration.*
By default the hardware appliance only uses the fail over (serial) cable for the high-availability heartbeat.

You can configure the heartbeat to be over either the network, fail-over cable or both:

To do this you need to edit the /etc/ha.d/ha.cf file on both nodes (using SSH/SCP/WINSCP etc.)

```
# serial serialportname ...
serial /dev/ttyS0 # Linux
#
# What interfaces to broadcast heartbeats over?
#
#bcast eth0 # Linux
```

Uncomment the bcast line (UDP broadcast will be activated)
Then restart heartbeat on both nodes.

The VMWare appliance defaults to using the network for its heartbeat.

Feedback agents

The load balancer can modify the weight (amount of traffic) of each server by gathering data from either a custom agent or an HTTP server. Just set the virtual servers feedback method to agent or http as required.
A telnet to port 3333 on a real server with the agent installed will return the current CPU idle as an integer 0-100
The load balancer expects a 0-99 integer response from the agent usually relating to the CPU idle i.e. a response of 92 would imply that the real servers CPU is 92% idle. The load balancer will then use the formula (92/10*requested_weight) to find the new optimized weight. Using this method an idle real server will get 10 times as many new connections as an overloaded server.

Installing the Windows agent

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

```
C:\>Instsrv.exe LBAGENT c:\LBCPUMon.exe

The service was successfully added!
Make sure that you go into the Control Panel and use
the Services applet to change the Account Name and
Password that this newly installed service will use
for its Security Context.
C:\>net start LBAGENT
The LBAGENT service is starting.
The LBAGENT service was started successfully.
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.
```

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
cmod 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 local time zone*

You can also use standard Linux commands:
To set the timezone use the command *timeconfig* (as root) (this will give you a list of timezones)
To set the time use the *date* command (as root)

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.

To manually set the date and time use the following commands:

```
     date --set 1998-11-02
     date --set 21:08:0
```

To save changes to the hardware clock do a:

```
hwclock --systohc
```

Using a recovery ISO image

A recovery ISO image can be downloaded on request. This will enable you to restore a completely new image of the load balancer software back onto the appliance. Once you have done this you will either need to re-configure from scratch or restore from your backups.

First burn the ISO image as a bootable CD. Then boot from the CD.

Press 2 to continue booting from the recovery CD.

Then use:

```
cd /etc/recovery
./start-restore.sh
```

Then follow the prompts to copy the image to the local hard drive.

NB. When the appliance first boots it will try to pick up an IP address by DHCP, just change this as required using the web interface.

Conclusion

You should have enough information here to be productive with your Loadbalancer.org appliance.

There are many aspects to the Loadbalancer.org appliance that have not been covered here, please contact support@loadbalancer.org if you have any questions or suggestions for improvements in the documentation.