



Loadbalancer.org Appliance Administration v5.9



Copyright © Loadbalancer.org Limited 2002-2007

Table of Contents

Loadbalancer.org Appliance Administration v5.9.....	1
Introduction.....	4
Console configuration.....	5
Remote configuration.....	6
Edit Configuration.....	7
Logical Load balancer configuration.....	7
Modify Logical Virtual Servers	7
Modify Logical Real Servers	12
Modify Global Settings	14
Modify logical Virtual Servers (Layer 7 HAProxy).....	15
Modify logical Real Servers (Layer 7 HAProxy).....	16
Enable Web Based Statistics (Layer 7 HAProxy).....	17
Physical Load Balancer Configuration.....	18
Modify the physical Real IP(s)	18
Modify the physical network configuration	18
Modify the physical Virtual IP(s).....	18
Modify logical Virtual Servers (SSL Termination Pound).....	19
Manage this SSL certificate.....	20
Adding an Intermediate key to the certificate chain.....	21
How do I import certificates exported from Windows Server?.....	22
Windows.....	22
UNIX.....	22
Services.....	23
Restart Heartbeat.....	23
Restart Ldirectord.....	23
Restart Pound-SSL.....	23
Restart HAProxy.....	23
Power Control.....	23
Shut down and restart server.....	23
Shut down and halt server.....	23
Advanced.....	24
Execute a shell command.....	24
Maintenance.....	25
Maintain Real Servers.....	25
Take a real server offline or online	25
Backup & Recovery.....	26
Make a configuration backup.....	26
Restore configuration from backup.....	26
Disaster Recovery Options.....	26
Security & Maintenance.....	26
Change passwords.....	26
Modify the maintenance (holding) page of this load balancer	27
Firewall lock down wizard.....	28
Modify the firewall script of this load balancer.....	29
Using the firewall script to NAT real servers.....	30
Initialise statistics tracking database (rrdtool)	31

Re-Initialise statistics tracking database (rrdtool)	31
Online Update & Security Patches.....	31
Using a recovery ISO image.....	31
Reports.....	32
Graphical stats over time.....	33
Advanced Topics.....	34
Configuring the web interface port binding as 9080&9443.....	34
Firewall Marks	35
FTP.....	35
Terminal Server RDP.....	36
Persistence > 15 minutes?.....	36
Server maintenance when using persistence.....	36
Persistence State Table Replication.....	37
Load balancing based on URL match.....	38
NIC Bonding and High-Availability.....	39
Example 1: Bonding for bandwidth.....	39
Example 2: Bonding for High-Availability (recommended).....	39
Example 3: Bonding for High-Availability & Bandwidth.....	39
8021q VLAN support.....	40
Fail over when network fails.....	41
Heartbeat over network as well as fail-over cable.....	41
Feedback Agents.....	42
Installing the Windows agent.....	42
Installing the Linux/Unix agent.....	43
Custom HTTP agent.....	43
Changing the local Date,Time,Time zone & Keyboard settings.....	44
SNMP installation & configuration.....	45
Round Up.....	46

Introduction

The Loadbalancer.org appliance is a standard x86 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: Slackware 10, Linux 2.4.32, 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

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/rc.d/rc.inet1.conf
                        /etc/resolv.conf
                        /etc/HOSTNAME
                        /etc/hosts
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

INFO : *You will be prompted for a password and the default username and password are both 'loadbalancer'*

Usually you would just use links to navigate to *Edit Configuration / Modify the physical Real IP(s)* 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.

Remoteconfiguration

Remote configuration is recommended in most cases, but be very cautious if you are changing the physical IP address. 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 WinSCP2 Secure File Transfer
- HTTP or HTTPS Web based Administration

NB. The default IP address for the Loadbalancer.org appliance is 10.0.0.21/255.255.0.0

For SSH and SCP login as *root* password *loadbalancer*

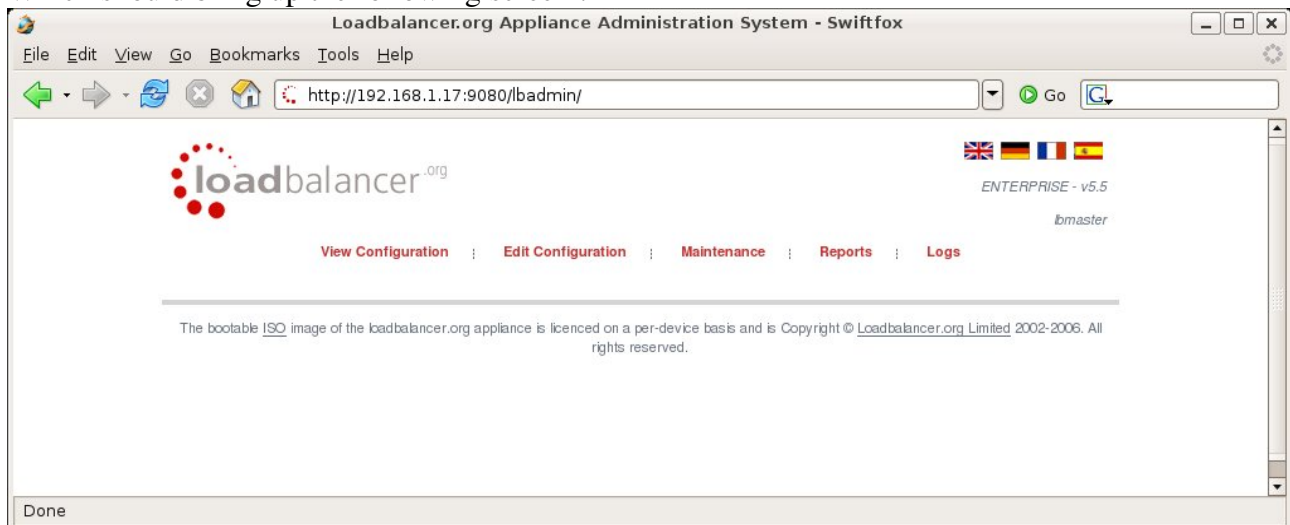
The Web based Administration interface uses a different set of user accounts and passwords based on the simple .htaccess files. This allows you to set up users in three groups configuration, maintenance and reports.

To access the web based administration interface use :

http://ApplianceIPaddress/lbadmin/

INFO : You will be prompted for a password and the default username and password are both '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

Edit Configuration

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

Logical Load balancer configuration

The logical load balancer configuration controls how the incoming traffic is broken down into virtual servers and real servers.

Modify Logical Virtual Servers

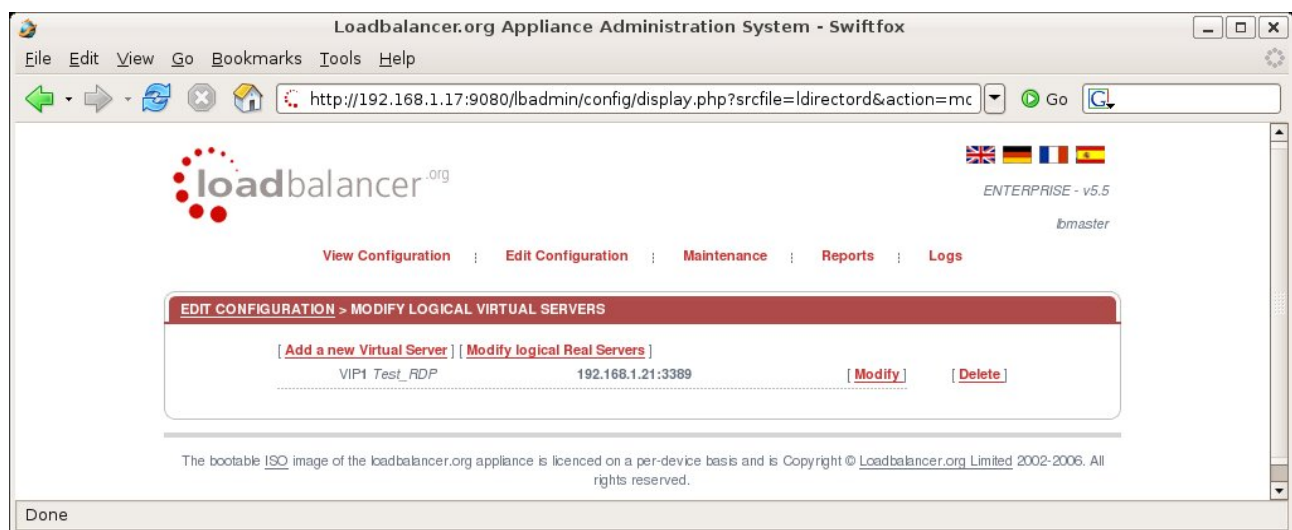
This menu option allows you to add, remove or modify virtual servers from your cluster.

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 :

192.168.1.30:80 & 192.168.1.30:443

NB. Assuming that 192.168.1.30 is the Physical Virtual IP address shared between the master and slave load balancer.



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

The screenshot shows a web browser window titled "Loadbalancer.org Appliance Administration System - Swiftfox". The address bar displays the URL "http://192.168.1.17:9080/lbadmin/config/display.php?srcfile=ldirectord&action=modvirtu:". The page features the "loadbalancer.org" logo and navigation links: "View Configuration", "Edit Configuration", "Maintenance", "Reports", and "Logs". A red banner at the top of the main content area reads "EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER". Below this, a form contains the following fields: "Label" (empty), "VIP Name" (empty), "Virtual Server (ipaddress:port)" (filled with "10.0.0.20:80"), and "Persistent" (a dropdown menu set to "no"). An "Add a new Virtual Server" button is positioned below the form. On the right side of the page, there are flags for the United Kingdom, Germany, France, and Spain, along with the text "ENTERPRISE - v5.5" and "lbmaster". A sidebar on the right shows a "VIP" list with the entry "10.". The browser's status bar at the bottom indicates "Done".

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

The screenshot shows the 'EDIT CONFIGURATION > MODIFY LOGICAL VIRTUAL SERVERS' page in the Loadbalancer.org Appliance Administration System. The browser window title is 'Loadbalancer.org Appliance Administration System - Swiftfox'. The address bar shows 'http://192.168.1.17:9080/lbadmin/config/display.php?srcfile=ldirectord&action=modv'. The page has a navigation bar with links: View Configuration, Edit Configuration, Maintenance, Reports, and Logs. The main content area contains a form for configuring a virtual server. The form fields are as follows:

Field	Value
Label	Test_RDP
Virtual Server (ipaddress:port)	192.168.1.21:3389
Persistent	no
Persistence Timeout	300
Scheduler	wrr
Fallback Server	127.0.0.1:80
Check Type	connect
Service to check	none
Check Port	
Virtual Host	
Login	
Password	
Protocol	tcp
Granularity	255.255.255.255
File to check	check.txt
Response expected	TESTOK
Forwarding Method	masq
Feedback Method	none

At the bottom of the form is a button labeled 'Modify logical Virtual Servers'. The status bar at the bottom of the browser window shows 'Done'.

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 fire wall 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
- 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

Modify Logical Real Servers

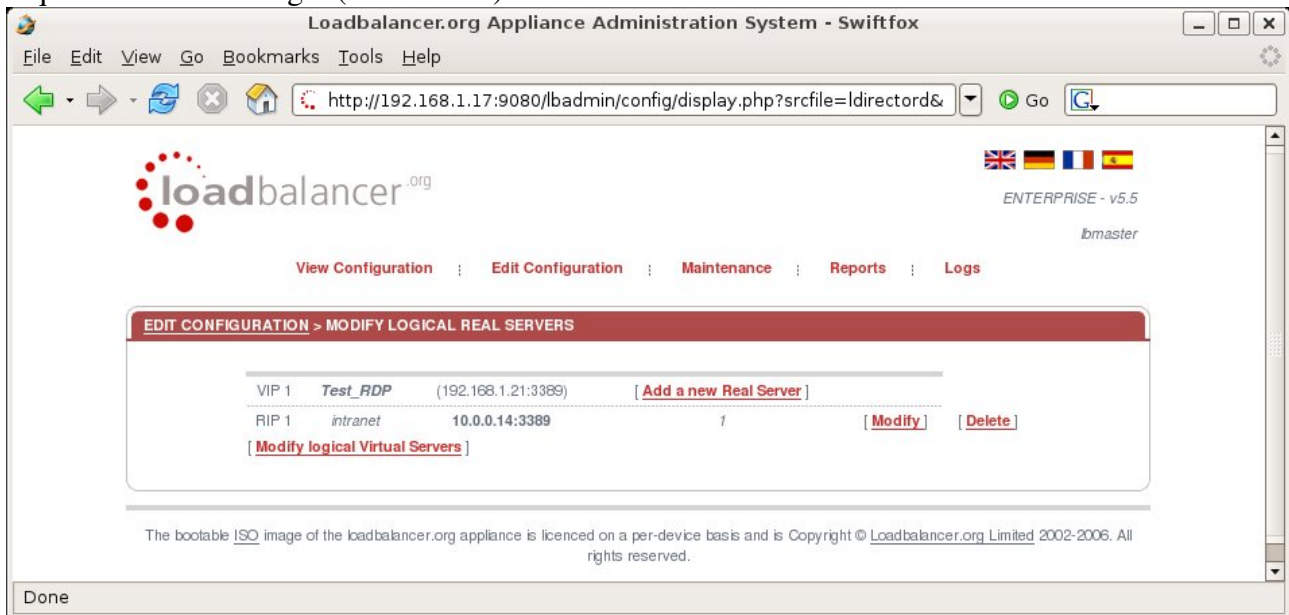
This menu option allows you to add, remove or modify real servers from your cluster.

Each virtual server has a number of real servers, for example one 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.

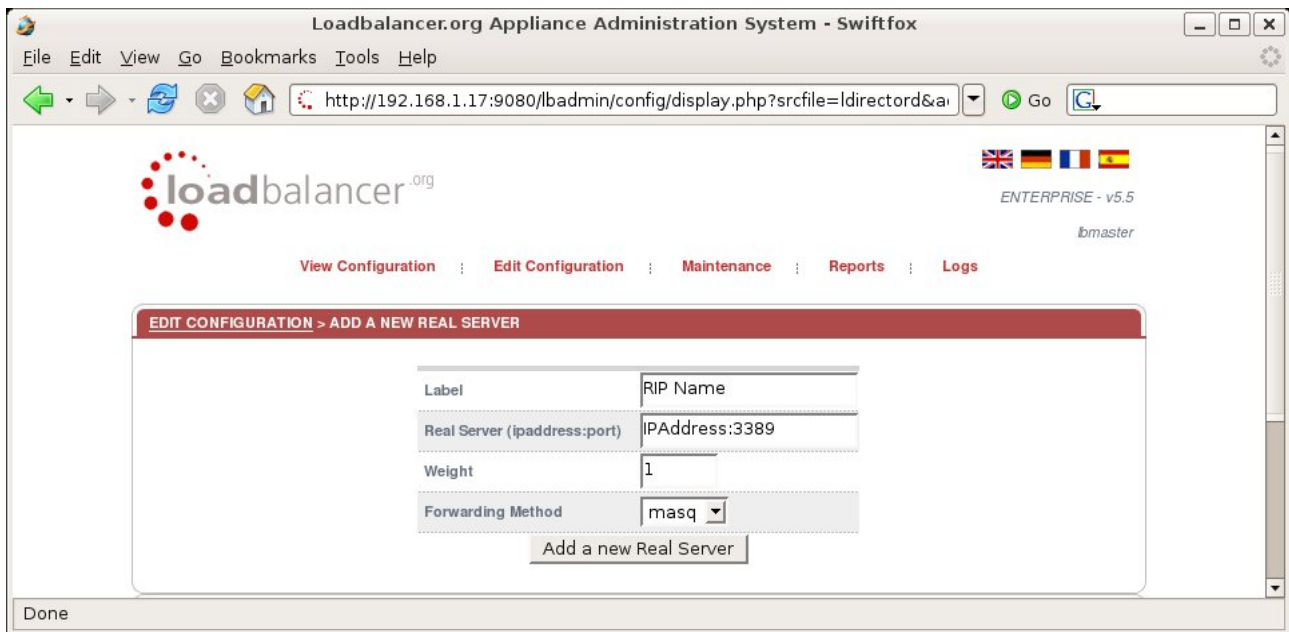
From the overview you can see each web server in the cluster, the IP address port number and the requested relative weight (0 is off line).



The screenshot shows a web browser window titled "Loadbalancer.org Appliance Administration System - Swiftfox". The address bar displays the URL "http://192.168.1.17:9080/lbadmin/config/display.php?srcfile=ldirectord&". The page features the Loadbalancer.org logo and navigation links: "View Configuration", "Edit Configuration", "Maintenance", "Reports", and "Logs". The main content area is titled "EDIT CONFIGURATION > MODIFY LOGICAL REAL SERVERS". It contains a table with the following data:

VIP 1	Test_RDP	(192.168.1.21:3389)	[Add a new Real Server]
RIP 1	intranet	10.0.0.14:3389	1

Below the table, there are links: "[Modify logical Virtual Servers]", "[Modify]", and "[Delete]". At the bottom of the page, a copyright notice states: "The bootable ISO image of the loadbalancer.org appliance is licenced on a per-device basis and is Copyright © Loadbalancer.org Limited 2002-2006. All rights reserved."



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 gate (direct routing), masq (NAT) can be used when you have two NICs and ipip (TUN) can be used to route through a tunnel across the Internet or WAN.

Selecting modify will bring up a similar dialogue 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 2 processor Xeon 4Ghz web server and a 1GHz Celeron web server, its possible you would increase the weight of the Xeon so that it took more of the load. Although in general most web server 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.

Modify Global Settings

This form allows 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 '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.

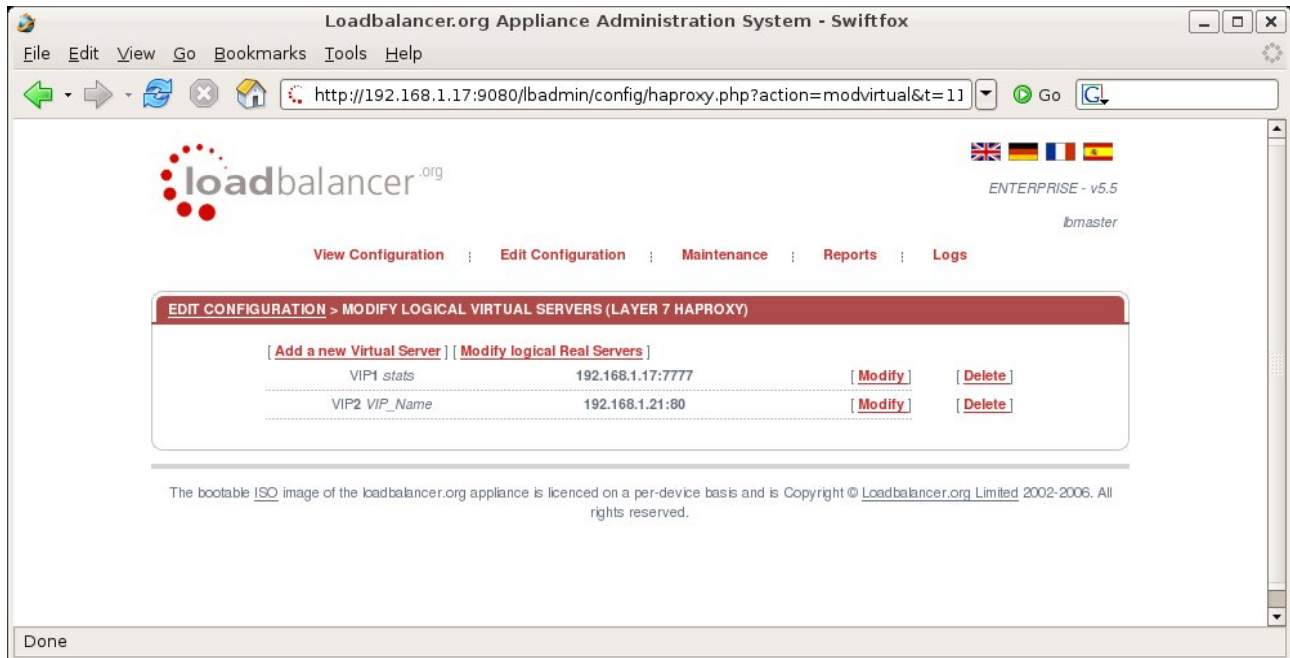
The screenshot shows a web browser window titled "Loadbalancer.org Appliance Administration System - Swiftfox". The address bar shows the URL "http://192.168.1.17:9080/lbadmin/config/display.php?srcfile=ldirectord&action=". The page features the "loadbalancer.org" logo and navigation links: "View Configuration", "Edit Configuration", "Maintenance", "Reports", and "Logs". The "Edit Configuration" link is highlighted, and the sub-header "EDIT CONFIGURATION > MODIFY GLOBAL SETTINGS" is displayed. The form contains the following settings:

Check Interval	6	?
Check Timeout	3	?
Negotiate Timeout	4	?
Quiescent	no	?

Below the form is a "Modify Global Settings" button. At the bottom of the page, a copyright notice states: "The bootable ISO image of the loadbalancer.org appliance is licenced on a per-device basis and is Copyright © Loadbalancer.org Limited 2002-2006. All rights reserved." The browser status bar at the bottom shows "Done".

Modify logical Virtual Servers (Layer7 HAProxy)

The layer 7 virtual servers are configured separately from the layer 4 ones because they use the HAProxy engine rather than the LVS engine.

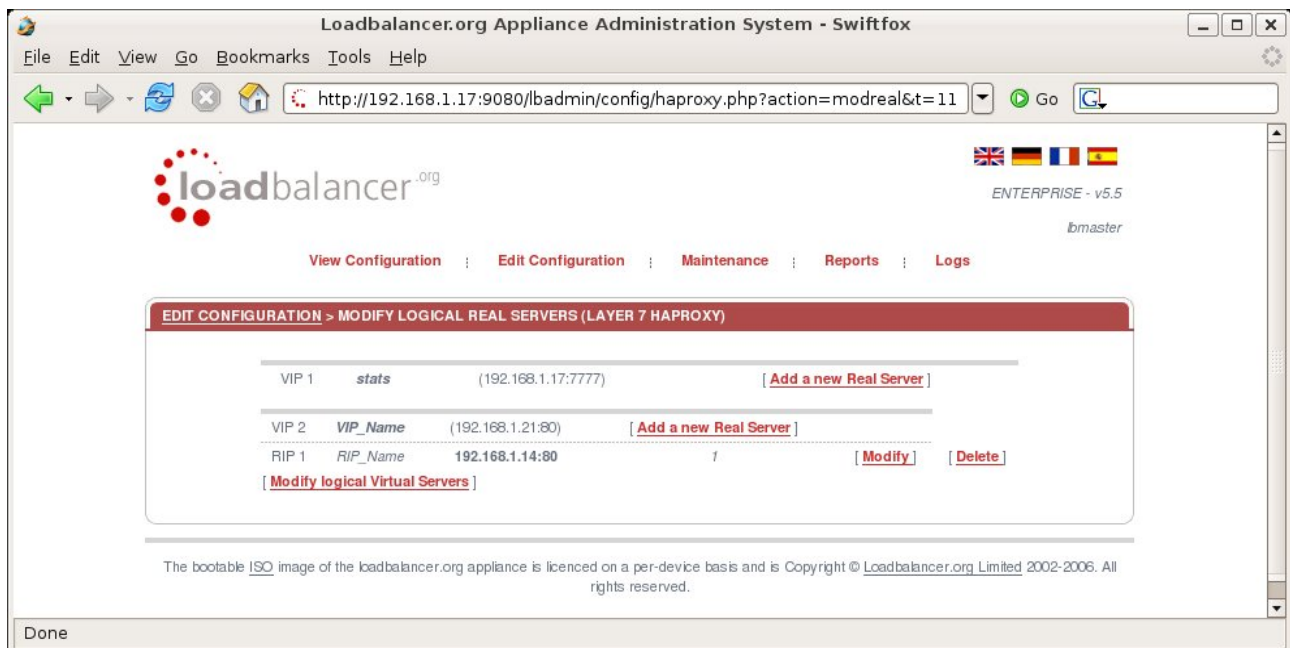


The Layer 7 HAProxy VIPs are created in the usual way by specifying a Virtual IP address and port for the service. If *persistence=no* then weighted round robin load balancing is performed. If *persistence=yes* and the *mode=tcp* then persistence by source IP is used.

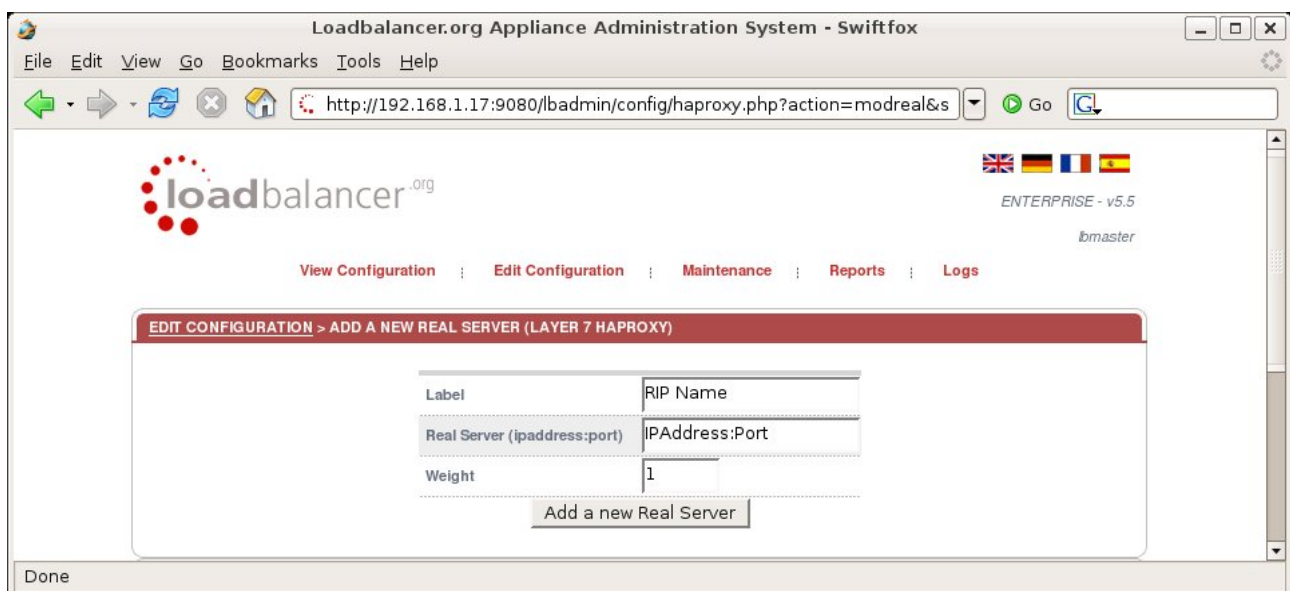
However if *persistence=yes* and the *mode=http* then the load balancer will automatically insert a cookie into each http request with the same name as the original destination server name. Therefore it is important that each real server is given a unique label when using cookie persistence.

Modify logical Real Servers (Layer7 HAProxy)

The real servers in a layer 7 configuration can be on any subnet in any network as long as they are accessible from the load balancer.



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



NB. Any changes to the layer 7 configuration 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 (Layer7 HAProxy)

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

File Edit View Go Bookmarks Tools Window Help Debug QA

Back Forward Reload Stop

http://127.0.0.1/haproxy?stats

The Mozilla Organiza... Latest Builds

HAProxy

Statistics Report for pid 11389

> General process information

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

active UP
active UP, going down
active DOWN, going up
active or backup DOWN

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

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

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

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

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

To enable the HaProxy web based statistics just add a new layer 7 VIP that is called 'stats'.

NB. The name is important.

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

EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER (LAYER 7 HAProxy)

Label	stats
Virtual Server (ipaddress:port)	10.0.0.21:7777
Persistent	no
Mode	tcp
Fallback	127.0.0.1:80

Add a new Virtual Server

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://10.0.0.21:7777/>.

Physical Load Balancer Configuration

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

Modify the physical Real IP(s)

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

***WARNING:** Obviously its 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.

Modify the physical network configuration

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.

Modify the physical Virtual 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 physical virtual IP(s) are controlled by heartbeat to ensure that only one of the load balancers (normally the master) owns the VIP(s). You can add as many virtual IP addresses 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.

Modify logical Virtual Servers (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

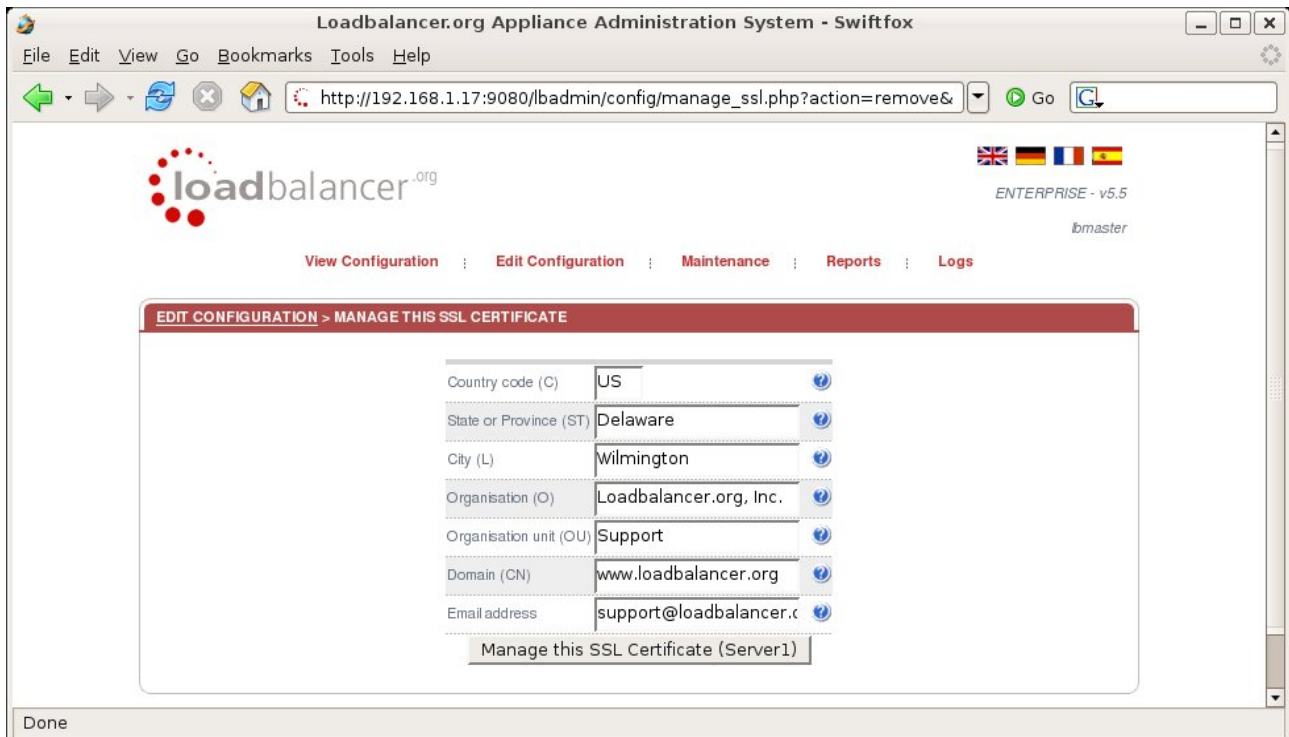
Virtual Server (ipaddress:port)	192.168.1.31:443
Backend	192.168.1.31:80
Add a new Virtual Server	

- 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. Edit Configuration > 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.

Managethis 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.



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.

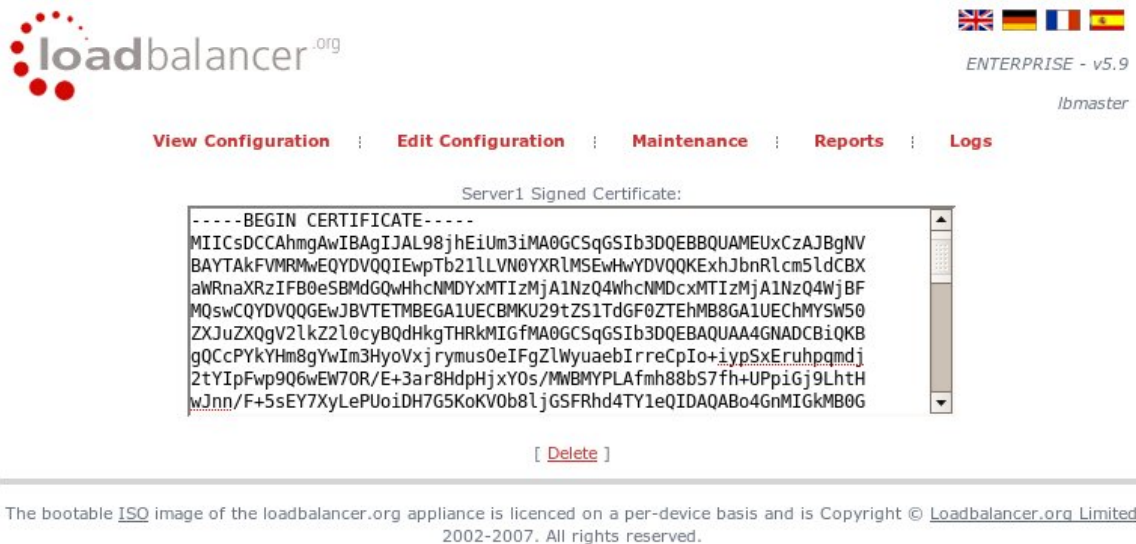
Once the signed key is uploaded you will need to restart Pound-SSL

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.



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-----
MIICsDCCAhmgAwIBAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUx
CzAJBgNV
BAYTAKFVMRMwEQYDVQIEwPb21lLVN0YXRIMSEwHwYDVQQKEzhJbnRlcm
5ldCBX
aWRnaXRzIFB0eSBMdGQwHhcNMDYxMTIzMjA1NzQ4WmcNMDcxMTIzMjA
1NzQ4WjBF
MQswCQYDVQGEwJBVTETMBEGA1UECBMKU29tZS1TdGF0ZTEhMB8GA1UE
ChMYSW50
ZXJuZXQv2lkZ2l0cyBqdHhkTHRkMIGfMA0GCSqGSIb3DQEBAQUAA4GN
ADCBiQKB
gQCCPYkYHm8gYwIm3HyoVxjrymusOeIFgZlWyuabIrreCpIo+iypSx
Eruhpmjdj
2tYIpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPfAfmh88bS7fh+UP
piGj9LhtH
wJnn/F+5sEY7XyLePUoiDH7G5KoKV0b8ljGSFRhd4TY1eQIDAQABo4
GnMIGkMB0G
-----
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwIm3HyoVxjrymusOeIFgZlWyuabIrreCpIo+iy
pSxEruhpmjdj2tYIpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPfAfmh88bS7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ/9Ec
SlqR7x/WAQUnFKVxQAMDatpeXSp3FGgXF+mpffusjEw=
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIEwDCCBCmgAwIBAgIY7GlzcWfeIAdoGNs+XVGezANBgkqhkiG9w0BAQUFADCB
Ai9TXvRIcD5q0mB+nyK9fB2aBzOiaiHSiWzAJeQjuqA+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

How do I 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.

Services

RestartHeartbeat

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 physical virtual 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.

RestartLdirectord

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

RestartPound-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 (have you used the *lbhiports* command?)

RestartHAPoxy

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 (have you used the *lbhiports* command?).

Restarts of HAProxy are completely graceful whether they succeed or not.

PowerControl

Shutdown and restartserver

Fairly self explanatory.

Shutdown and haltserver

You don't normally want to do this.

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.*

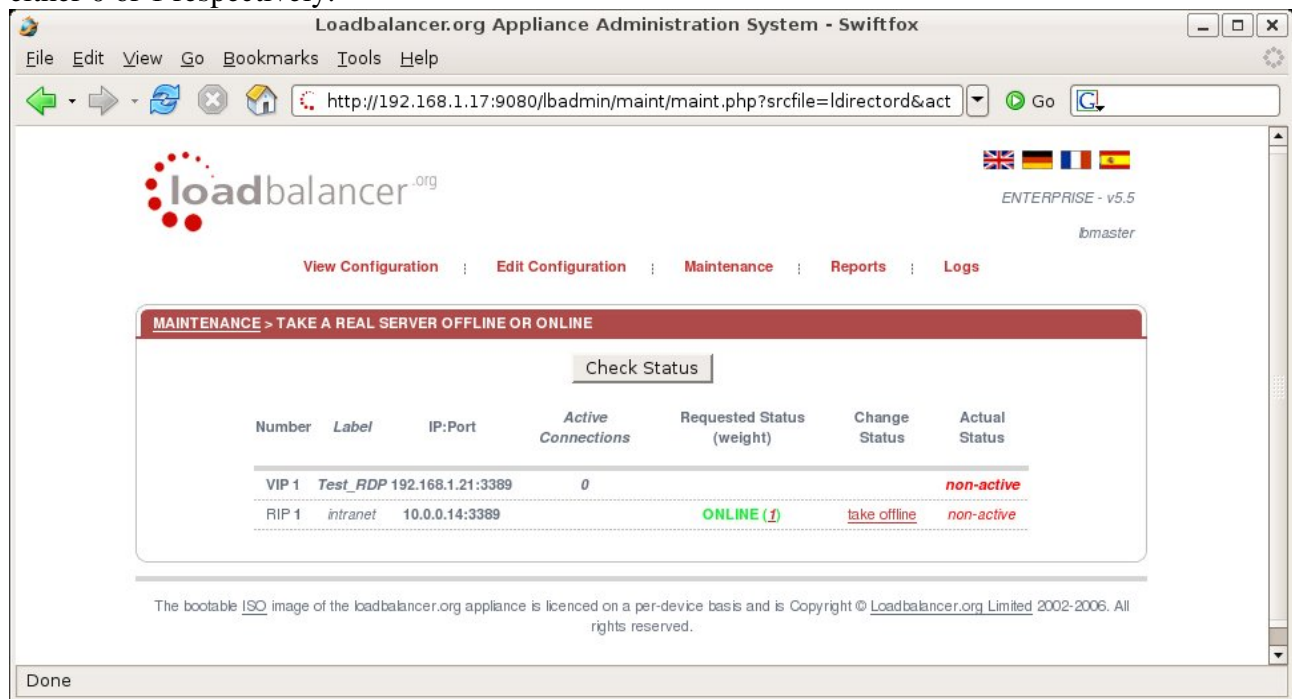
The output of the command will be displayed on screen.

Maintenance

Maintain Real Servers

Take a real server offline or online

This form allows you to view all of the virtual and associated real servers ip addresses, port numbers and weights. Clicking take offline or bring online will change the weight of the server to either 0 or 1 respectively.



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

Backup& Recovery

Your Loadbalancer.org appliance is covered by a return to base warranty, and re-configuration is simple from the default install BUT its always nice to have a backup!

Makea configurationbackup

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

Restoreconfigurationfrombackup

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 both Heartbeat and Ldirectord.

DisasterRecoveryOptions

This section gives you the following options :

- Restore manufacturers settings – Handy if you want to start all over again
- Download Config – Allows you to download ldirectord.cf (virtual and real server config)
- Download Real IP Address Info – Your physical ip address
- Download VIPs – Your physical virtual ip address info
- Download Firewall Script – The firewall configuration (for creating firewall marks)
- Upload a previously saved config file – To upload a previous ldirectord.cf file.

Security& Maintenance

Changepasswords

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.

Modify the maintenance(holding) page of this load balancer

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

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

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

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.

Firewalllock 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
```

Modify the firewall script of this load balancer

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

WARNING: BE CAREFULL! 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 i.e. 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 admin IP addresses 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 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.

Using the firewall script to NAT real servers

If you are using layer 4 load balancing in NAT mode:

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
# i.e. Everything coming from the internal subnet should be
# automatically NATed to the external subnet
# If you don't do this you will have no Internet access from your
# real servers (which may not be required)
```

- 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
# NB. You will need a floating VIP set up for the external IP if
# you haven't got one already
$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
#NB. Obviously this should now be locked down with ACCEPT & DENY
# rules on FORWARD chain
```

Initialise statistics tracking database (rrdtool)

Once you have configured all of your virtual and real servers you will probably want to initialise 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.

Re-Initialise statistics tracking database (rrdtool)

This will allow you re-initialise small changes such as IP addresses or labels etc. to the graphs but will not pick up new VIPs or RIPv. You will need to initialise the whole database from scratch for that.

Online Update & Security Patches

If you have a valid software maintenance licence for your site (*each licence 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 authorisation 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.

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 attach an ATAPI CD-DRIVE using the SATA socket and allow the appliance to boot from the CD.

Press 2 to continue booting from the recovery CD.

Then use :

```
cd /etc/restore  
./clone-dsk.sh
```

Then follow the prompts to copy the 1GB image to the local flash card.

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 *RealIPAddress:7777* , created when you modify the load balancers physical 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 show check that you have a Layer7 Virtual IP configured with a label of *stats on RealIPAddress:7777. ie.*

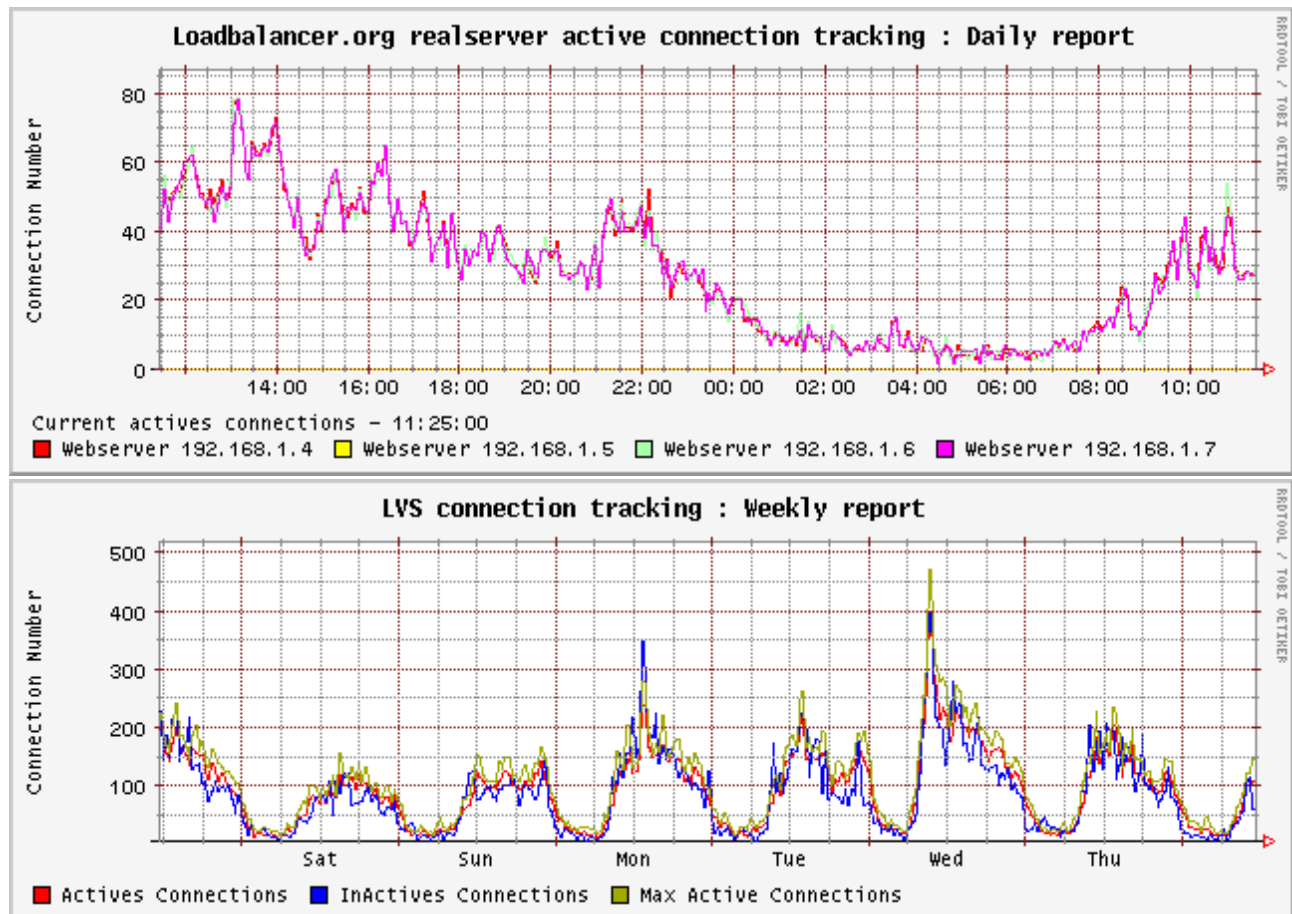
```
listen stats 10.0.0.20:80
      stats enable
      stats uri /
      server dummy 127.0.0.1:80
```


Graphical stats overtime....

This link goes to a page generated by the *initialise statistics database* command. The graphs generated are great for showing management pretty pictures they wont 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.



AdvancedTopics

Configuring the web interface port binding as 9080&9443

Follow the previous configuration instruction to set up your load balancer or clustered pair with a physical IP address.

IMPORTANT: The load balancer administration interface uses port 80 & 443 by default this will need to be changed before implementing a proxy on those ports.

From the local console, or the remote console via SSH or Putty login as *root* and:

Type the command: *lbhiports*

This will modify the local Apache server to respond to ports 9080 and 9443 respectively for the administration interface.

You can also do this from the web interface:

IMPORTANT: If you do this from the web interface the script will NOT be able to restart the Apache process. A cron job will realize the server is dead and restart it but this may take up to 7 minutes.

From the web based administration interface go to *Edit Configuration > Execute shell command*

Type the command: *lbhiports*

Then hit the 'Execute command' button.

The web browser should then **fail to connect** (as the ports have changed).

Modify your browsers URL to use port 9080 or 9443 respectively i.e.

<http://192.168.1.21:9080/lbadmin/>

FirewallMarks

You can use the modify firewall script option to group certain protocols together in one cluster. So rather than specifying VIP as Ippaddress: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 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.

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.

NB. By default a Terminal Sever connection that is not minimised 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 15mins then you will need to increase the load balancers TCP time out value. The TCP time out can be set a command in your firewall script:

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

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

Server maintenance when using persistence.

The default settings of the load balancer when persistence is enabled is to continue forwarding all connection requests from a source IP to the same server until the persistence timeout expires. This is true even if you set the weight of the server to zero for maintenance.

NB. By default the global setting 'quiescent=yes' makes sure that if a server fails a health check it is physically removed from the load balancing table hence forcing clients to change server.

With 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 time out. You can change this behaviour so that when a client disconnects the persistence template is cleared forcing them to re-connect to a different server.

This is activated globally by using the following commands from the console:

```
echo 1 > /proc/sys/net/ipv4/vs/expire_quiescent_template  
echo 1 > /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 firewall script.

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 synchronisation 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. As of v5.9 you can configure the bonding of network cards using Edit Configuration > Modify the physical Real IP(s).

If required you can change the bonding mode in the `/etc/rc.d/rc.bondX` files:

Example1: Bonding for bandwidth

```
modprobe bonding mode=0 # Bonding Link Aggregation mode
ifconfig bond0 down
ifconfig eth0 down
ifconfig eth1 down
ifconfig bond0 10.0.0.21 netmask 255.255.0.0 broadcast 10.0.255.255 up
ifenslave bond0 eth0
ifenslave bond0 eth1
```

NB. Are you really doing 1Gb/s+?

Example2: Bonding for High-Availability(recommended)

```
modprobe bonding mode=1 miimon=100 # Bonding Active-Passive mode
ifconfig bond0 down
ifconfig eth0 down
ifconfig eth1 down
ifconfig bond0 10.0.0.21 netmask 255.255.0.0 broadcast 10.0.255.255 up
ifenslave bond0 eth0
ifenslave bond0 eth1
```

NB. This works with any switch.

Example3: Bonding for High-Availability& Bandwidth

```
modprobe bonding mode=4 miimon=100 # Bonding 802.3ad mode
ifconfig bond0 down
ifconfig eth0 down
ifconfig eth1 down
ifconfig bond0 10.0.0.21 netmask 255.255.0.0 broadcast 10.0.255.255 up
ifenslave bond0 eth0
ifenslave bond0 eth1
```

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

8021qVLAN support

Native 8021qVLAN support can be enabled to load balance clusters on multiple VLANs.
Modify the rc.firewall script to make your desired physical network settings (*obviously safer to do this from the console!*)

```
#Enable 8021q VLAN support in the Kernel
modprobe 8021q
#To add a vlan to eth0
vconfig add eth0 2
#To configure its IP address.
ifconfig eth0.2 192.168.1.21 netmask 255.255.255.0 broadcast 192.168.1.255 up
```

Then create a physical floating VIP 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
```


Fail over when network fails

NB. This can now be configured via Edit Configuration > Modify 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.

Then you need 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

NB. This can now be configured via Edit Configuration > Modify Heartbeat configuration

By default the hardware appliance only use 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.)

Find the section:

```
# 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.

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

FeedbackAgents

The load balancer can modify the weight (amount of traffic) of each server by gathering data from either a custom agent or an HTTP server. Just set the virtual servers feedback method to agent or http as required.

A telnet to port 3333 on a real server with the agent installed will return the current CPU idle as an integer 0-100

The load balancer expects a 0-99 integer response from the agent usually relating to the CPU idle i.e. a response of 92 would imply that the real servers CPU is 92% idle. The load balancer will then use the formula $(92/10 * \text{requested_weight})$ to find the new optimised 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
chmod 644 /etc/xinetd.d/lb-feedback

/etc/init.d/xinetd restart

Testing:
telnet 127.0.0.1 3333

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

Custom HTTP agent

You can use any HTTP server responding on port 3333 to give feedback information to the load balancer.

The format of this information must be an integer number of 0-100 without any header information.

Using this method you can generate a custom response based on your applications requirements i.e. A mixture of memory usage, IO & CPU etc.

Changing the local Date, Time, Timezone & Keyboard settings

There isn't a GUI for changing the date & keyboard but you can 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 i.e. crontab -e

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
```

The keyboard map is loaded by the script /etc/rc.d/rc.keymap :

```
#!/bin/sh
# Load the keyboard map.  More maps are in /usr/share/kbd/keymaps.
if [ -x /usr/bin/loadkeys ]; then
    /usr/bin/loadkeys uk.map
fi
```

Just modify this script for the key map you require.

SNMP installation & configuration

SNMP is an optional installable module for the Loadbalancer.org appliance. You can find all the relevant files and instructions here:

<http://www.loadbalancer.org/download/SNMP/>

SNMP module for LVS

Compiled for Loadbalancer.org appliance with 2.4.32 Kernel.

REQUIRED FILES:

net-snmp-5.3.0.1.tar.gz

OC.txt

LVS-MIB.txt

libnetsnmplvs.so

Download net-snmp-5.3.0.1.tar.gz

copy it to /home/loadbalancer/

extract it (tar -zxvf ./net-snmp-5.3.0.1.tar.gz)

cd net-snmp-5.3.0.1

./configure

make

make install

Copy LVS-MIB.txt and OC.txt to /usr/local/share/snmp/mibs/

Check that the new MIB is visible by invoking:

```
shell> snmptranslate -m LVS-MIB -On -IR lvsServiceEntry,  
this should return .1.3.6.1.4.1.8225.4711.17.1
```

Copy libnetsnmplvs.so to /usr/local/lib/libnetsnmplvs.so

Edit /usr/local/etc/snmp/snmpd.conf:

Change the following section to give read only access:

```
-----  
#com2sec paranoid default      public  
com2sec readonly default      public  
#com2sec readwrite default     private  
-----
```

Then add the following line:

```
-----  
dlmod lvs /usr/local/lib/libnetsnmplvs.so  
-----
```

Start the SNMP daemon:

snmpd [return]

Try 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)
```

```
...
```

RoundUp

You should have enough info 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.