Appliance Quick Start Guide

v7.2

Table of Contents

Loadbalancer.org terminology

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

What is a Virtual IP address?

Most load balancer vendors use the term virtual IP address (VIP) to describe the address that the cluster is accessed from. It is important to understand that the virtual IP (VIP) refers both to the physical IP address and also to the logical load balancer configuration. Likewise the real IP (RIP) address refers both to the real servers physical IP address and its representation in the logical load balancer configuration.

What is a Floating IP address?

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

What are Your Objectives?

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

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

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

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

In order to achieve all three objectives of performance, reliability & maintenance in a web based application, your application must handle persistence correctly (see page 34 for more details).

What Is The Difference Between a One-Arm and a Two-Arm Configuration?

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

One-Arm	The load balancer has **one** physical network card connected to **one** subnet
Two-Arm	The load balancer has **two** network interfaces connected to **two** subnets - this may be achieved by using two physical network cards or by assigning two addresses to one physical network card

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 or 2 ARM
Layer 7	SNAT (Source Network Address Translation: HAProxy)	Layer 7 allows great flexibility including full SNAT and WAN load balancing, cookie insertion and URL switching ***Not as fast as Layer 4***	1 or 2 ARM

Key:

Recommended for high performance fully transparent and scaleable solutions

Recommended if HTTP cookie persistence is required, also used for numerous Microsoft applications such as Terminal Services (RDP cookie persistence) and Exchange, that require SNAT mode

Only required for Direct Routing implementation across routed networks (rarely used)

Loadbalancer.org Recommendation:

Where feasible, one-arm direct routing (DR) mode is our recommended method because it's a very high performance solution with little change to your existing infrastructure.

Sometimes it's not possible to use DR mode. The two most common reasons being: if the application cannot bind to the RIP & VIP at the same time; or if the host operating system cannot be modified to handle the ARP problem (see page 20-29 for more details)

A second option is Network Address Translation (NAT) mode. This is 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.

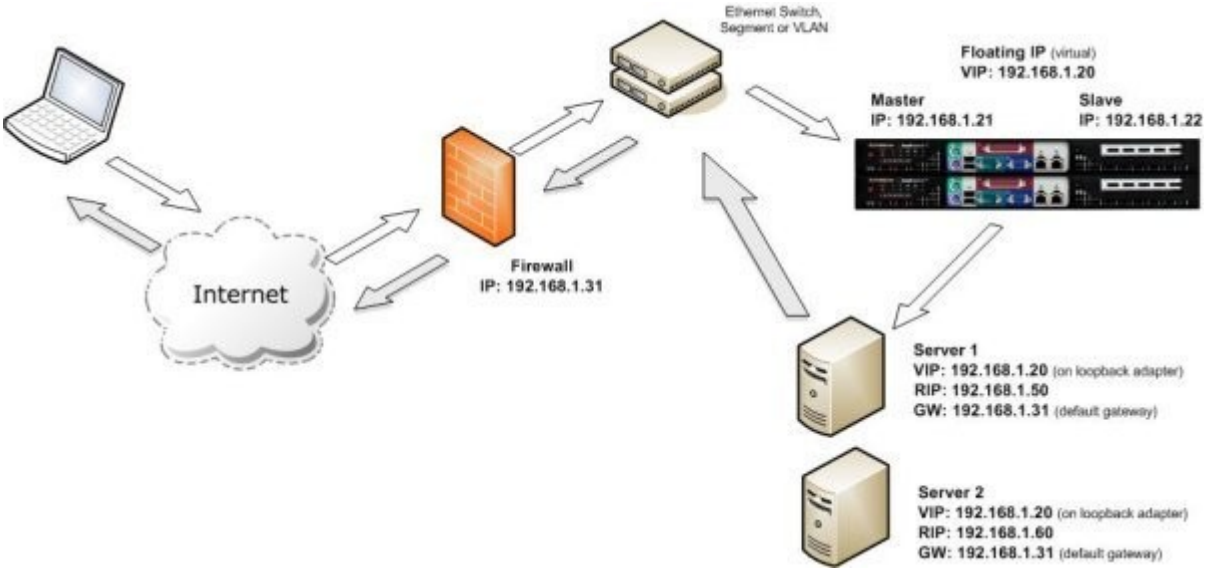
The third option is Source Network Address Translation (SNAT) mode using HAproxy. If your application requires that the load balancer handles cookie insertion, RDP cookies, Session Broker integration or SSL termination then this option is appropriate. This can be deployed in one-arm or two-arm mode and does not require any changes to the application servers. HAproxy is a high-performance solution that operates as a full proxy, but due to this it cannot perform as fast as the layer 4 solutions.

If your application doesn't maintain its own state information then you may need to use cookie insertion to maintain server persistence (affinity)

The following sections describe these configurations in more details.

Direct Routing (DR)

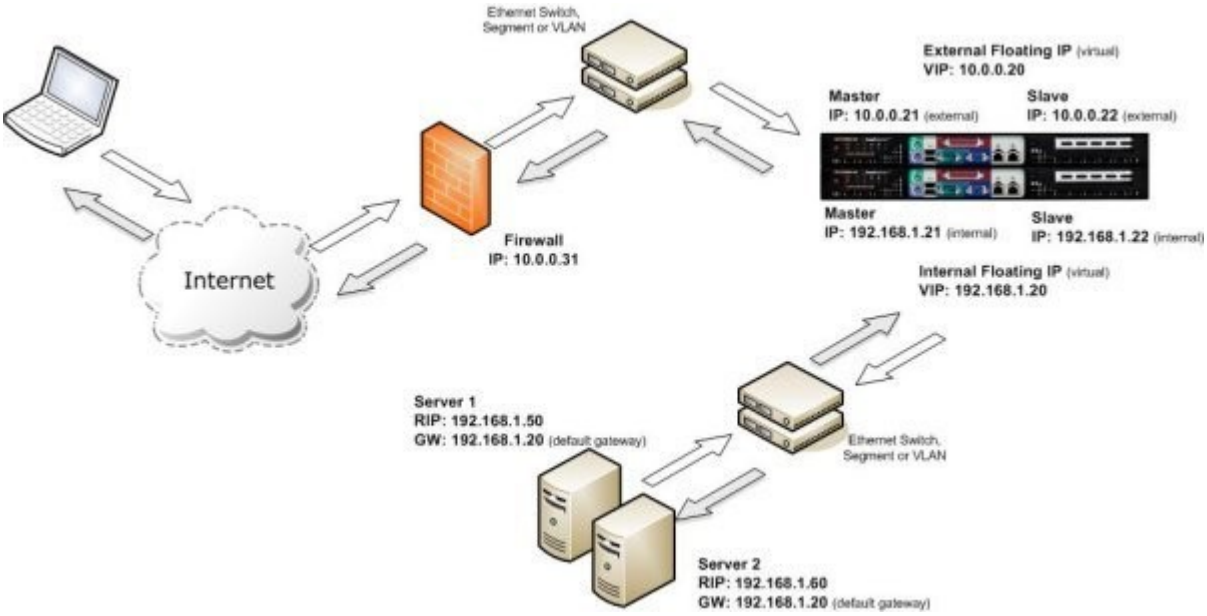
The one-arm direct routing (DR) mode is the recommended mode because it's a very high performance solution with little change to your existing infrastructure. *NB. Foundry networks call this Direct Server Return and F5 call it N-Path.*



- Direct routing works by changing the destination MAC address of the incoming packet on the fly which is very fast
- However, this means that when the packet reaches the real server it expects it to own the VIP. This means you need to make sure the real server responds to both its own IP and the VIP, but does not respond to ARP requests for the VIP. Please refer to page 20-29 for more details on resolving the ARP problem
- 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
- Load balanced services can be configured directly on the interface (normally eth0) with no additional IP address. However, when using a clustered pair, all load balanced virtual services **must** be configured on a floating IP to enable failover & failback between master & slave
- The virtual server and real servers must be in the same switch fabric / logical network. They can be on different subnets, provided there are no router hops between them. If multiple subnets are used, an IP address in each subnet must be defined on the load balancer
- Port translation is not possible in DR mode i.e. having a different RIP port than the VIP port
- DR mode is transparent , i.e. the real server will see the source IP address of the client
- Administration of the load balancer is via any active IP address (on port 9080)

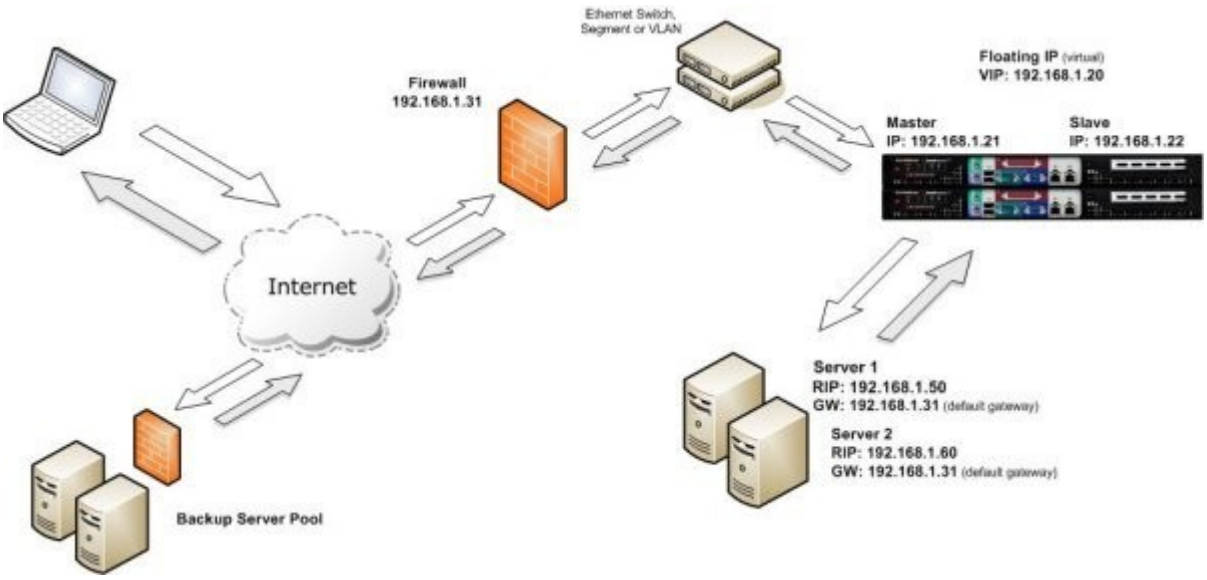
Network Address Translation (NAT)

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 problem. 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).



- In two-arm NAT mode the load balancer translates all requests from the external virtual server to the internal real servers
- It is a good idea to use *eth1* as your external network and *eth0* as internal, otherwise you will need to change the *autonat* setting in *Edit Configuration > Layer 4 – Advanced Configuration*
- The real servers must have their default gateway configured to point at the load balancer. When master & slave units are used, a floating IP **must** be used to enable failover
- Real servers are automatically given access to the Internet through the load balancer (via autonat)
- Load balanced services can be configured directly on the interface (normally eth0) with no additional IP address. However, when using a clustered pair all load balanced virtual services must be configured on a floating IP to enable failover & failback between master & slave
- Normally the virtual server and real servers should be located on different subnets within the same logical network (i.e. no router hops) and the load balancer should have an IP address in each subnet. *Note-1: It is possible to have real and virtual servers in the same subnet – please refer to the Advanced NAT topic in Section F of the administration manual. Note-2: It is possible to have the real servers located on routed subnets, but this would require a customized routing configuration on the real servers and is not recommended*
- If you want real servers to be accessible on their own IP address for non-load balanced services, e.g. SMTP or RDP, you will need to setup individual SNAT and DNAT firewall script rules for each real server. Please refer to the Advanced NAT Considerations section in the administration manual for more details
- 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 the routing configuration on the real servers. Please refer to the Advanced NAT Considerations section in the administration manual for more details.
- NAT mode is transparent , i.e. the real server will see the source IP address of the client
- Administration of the load balancer is via any active IP address (on port 9080)

Source Network Address Translation (SNAT)



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.

The network diagram for the Layer 7 HAProxy SNAT 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.

- As with other modes a single unit does not require a Floating IP, although it is recommended to make adding a slave unit easier
- SNAT is a full proxy and therefore load balanced real servers do not need to be changed in any way
- Because SNAT is a full proxy any server in the cluster can be on any accessible subnet including across the Internet or WAN
- SNAT is not transparent by default , i.e. the real servers will not see the source IP address of the client, they will see the load balancers IP address. If required, this can be solved by either enabling TPROXY on the loadbalancer, or leveraging the X-forwaded-For header. See the administration manual for more details.

For detailed configuration examples, please refer to section D in the administration manual

High-Availability Configuration of Two Loadbalancer.org Appliances

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

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

Clustered Pair Configuration Methods

There are two ways to configure a clustered pair; either by using the wizard or configuring the units manually.

Using the Wizard

If the wizard is used, the slave is configured first and then the master. This ensures that both units can first communicate via the selected link (via a serial cable or over the network) , and also that settings that are setup on the master and correctly replicated to the slave.

For more details on using the wizard and an example, refer to page 16

Manual configuration

If the master is configured first without using the wizard and the slave is added later, the following points should be considered:

- The IP address for the slave must be configured in the master using *Edit Configuration* > *Hostname DNS* in the WUI
- The **Force full slave sync** option in *Edit Configuration* > *Hostname DNS* should be checked prior to clicking **Update** - this will ensure that all configured services are correctly replicated over to the slave unit
- Once the IP address is set and synchronization has occurred, its important to restart heartbeart on both units to ensure heartbeat starts cleanly. This can be done via *Maintenance* > *Restart Services* in the WUI

For more details refer to configuration Example-2 on page 48 of the admin manual, for more details on adding a slave and verifying failover refer to page 127 and for more details on setting up heartbeat, refer to page 99

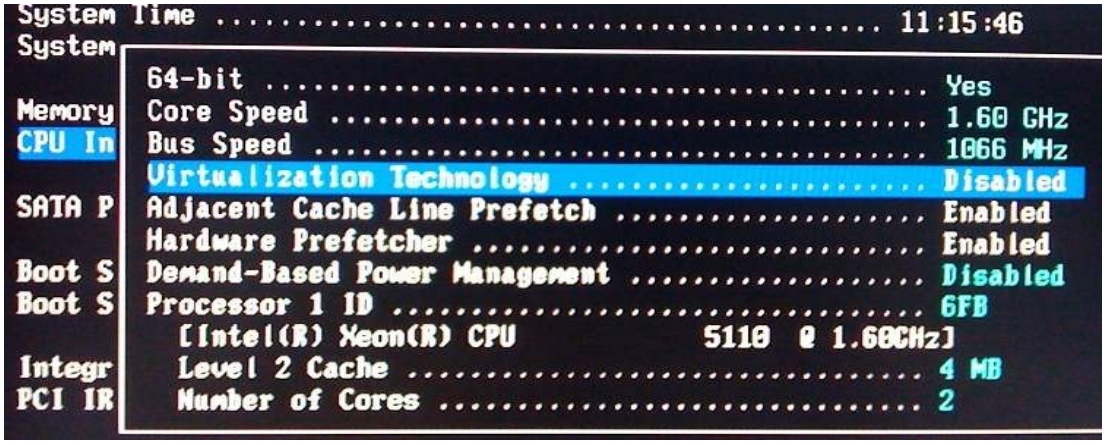
VMware Virtual Appliance

Host Server requirements & preparation for VMware

To be able to successfully run the Loadbalancer.org Enterprise VA under VMware, the following basic server specifications must be met:

- A compatible 64bit CPU
- Virtual Technology hardware support – either Intel-VT or AMD-V compliant CPU's

For an Intel based server, VT may need to be enabled in BIOS as shown in the example below:



If your server is unable to support 64bit guests, a message similar to the following message will be displayed when trying to start the VA:

Appliance download formats

VMware has a number of formats and versions for systems and files. It can get confusing which type / version is needed for your specific environment. The section below explains what is needed for various versions of VMware. All files can be downloaded from the downloads page on our website.

VMware Player, Workstation & Server

- Download file LBVMv7.zip (virtualHW.version = 4)
- For VMware server v2.x you can highlight the VA after import and select Upgrade to Hardware v7, for VMware Player, Workstation & Server v1.x no further steps are required

Vsphere Client 4.x & ESX 4.x / ESXi 4.x

- Download file LBVMESXv7.zip (ovf v1.0, hardware v7)

Virtual Infrastructure Client 2.5.x & ESX 3.x / ESXi 3.x

In this case you have two choices:

- Download file LBVMESXv7_ovf0.9.zip (ovf v0.9, hardware v4) from the Quick Download Links section of the downloads page

  or

- Download LBVMv7.zip from our downloads page and use the converter for your environment to convert to a compatible VA

NOTE: Due to Vmxnet3 driver compatibility limitations with the various versions of ESX & ESXi only the LBVMESXv7.zip download uses the Vmxnet3 network drivers. The other downloads use E1000 drivers

Setting up the Loadbalancer.org Virtual Appliance

1. Download & extract the appropriate file (see previous section)
2. Import the VA:
   - For VMware Server use: ***Virtual Machine > Add VM to Inventory***
   - For Vsphere use: ***File > Deploy ovf Template***
   - For Virtual Infrastructure use: ***File > Virtual Appliance > Import***
3. Start the Virtual Appliance, allow a minute for booting

VMware Tools

With the exception of the Vmxnet3 network drivers used in the LBVMESXv7.zip download, our appliance doesn't strictly need any of the extra VMware tools functionality.

As explained on the previous page, only the LBVMESXv7.zip download uses the Vmxnet3 drivers, the other two downloads use the E1000 network driver which is part of the default kernel. Therefore, it is really only necessary to upgrade the tools on the LBVMESXv7.zip version.

To upgrade VMware tools, detailed instructions are available on our blog at:

http://blog.loadbalancer.org/how-to-upgrade-vmware-tools-on-clusterload-esx-or-loadbalancerorg-va/

Balloon Driver

We recommend leaving the balloon driver disabled.

This has been configured on the v7.3 appliance by adding `sched.mem.maxmemct=0` to the advanced configuration parameters for the VM.

For more details on this, please refer to the following Vmware Link :

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1002586

Physical Appliance (for reference)

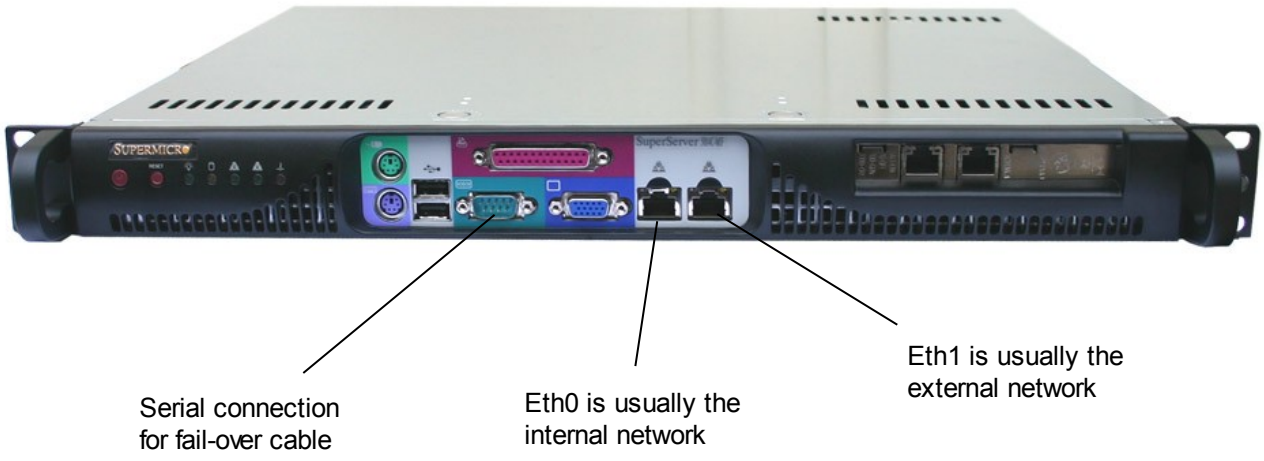
- Remove all packaging
- Rack mount the appliance if required
- The power supply is an auto sensing unit (100v to 240v)
- Connect the power lead from the power socket to the mains or UPS
- Connect your network cable from your switch or hub to the internal network port (*eth0*)
- If using a two-armed configuration connect a second network cable to the external port (*eth1*)

If two load balancers (recommended) are being used, connect a null modem cable (one cable is supplied with each appliance) between the two serial ports, then configure the slave first

- Attach a monitor to the VGA port
- Attach a keyboard to the USB or PS/2 port
- Check mains power is on
- Press the power switch to start the appliance (fans should start & front panel LEDs should light)
- Allow a minute for booting

The next few pages of this document detail the following steps:

- Configuring the load balancer using the web based wizard
- Additional appliance configuration using the web interface
- Testing the load balancer configuration



Configuring The Loadbalancer.org Appliance Using The Web Based Wizard

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

Network interface configuration

log in to the console: ***Username**:* root
***Password**:* loadbalancer

You can access the web interface either via Links at the console or from a web browser on a client connected to the same network (***recommended***). By default the IP address for eth0 on the physical appliance is set to 192.168.2.21/24. If another device already has this IP address then no address will be assigned. If you want to change or assign an IP address, the following command should be used once logged in as root:

```
ip addr add <IP address>/<mask> dev eth0
```

e.g.

```
ip addr add 192.168.1.100/24 dev eth0
```

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

Accessing the Web User Interface (WUI)

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

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

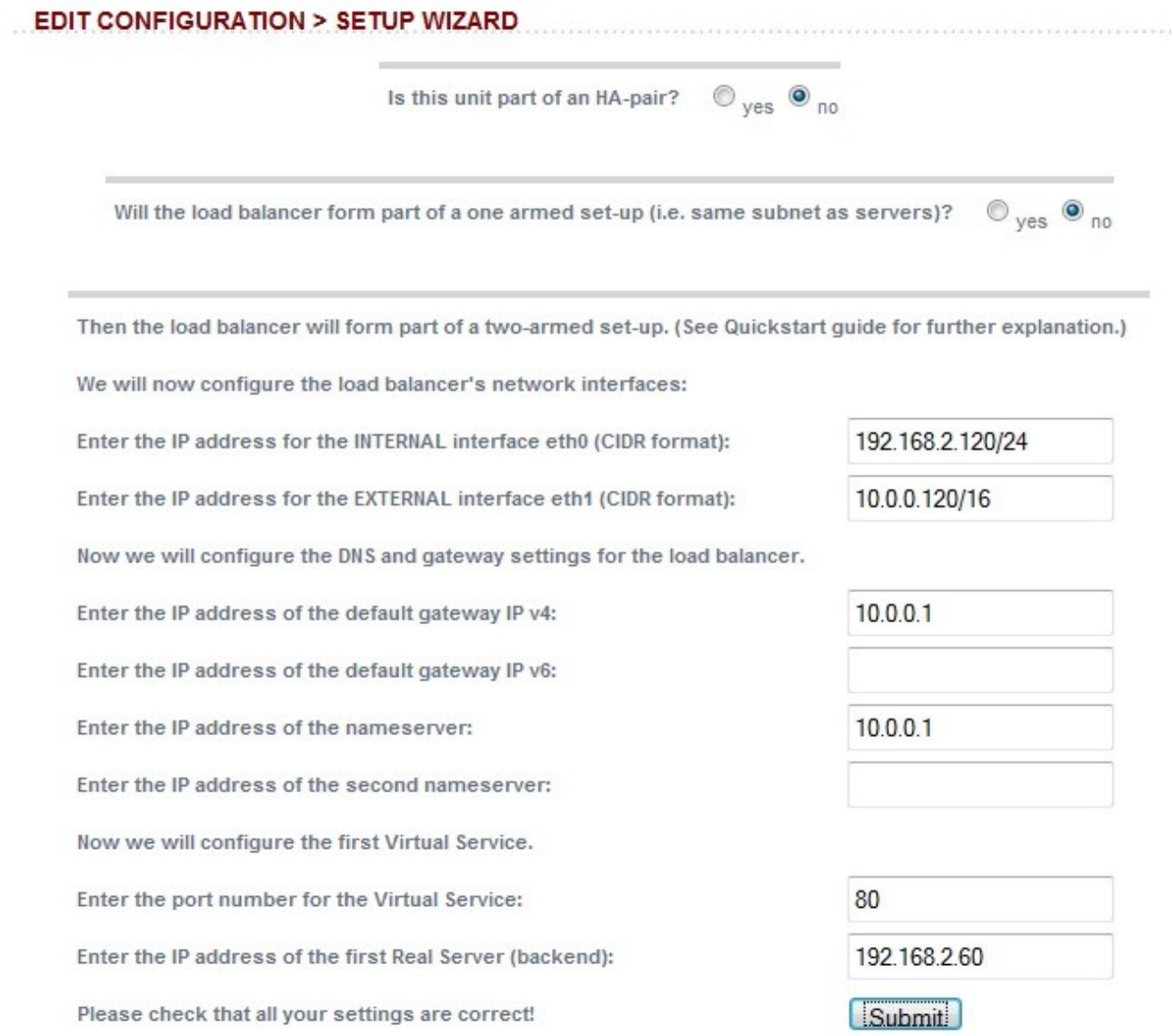
log in to the WUI: ***Username**:* loadbalancer
***Password**:* loadbalancer

NOTE: 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 configure the appliance with a basic configuration.

Example answers using the wizard for a two-arm NAT configuration (single unit)

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



Check that your settings are correct and click **Submit**. Once the wizard is complete the load balancer is configured and ready to use.

For NAT mode, you also need to configure the real server to ensure that it uses the internal IP address of the load balancer as its default gateway. Once this is done you can test the virtual server form the external network. By default, the wizard uses the IP address of the external interface for the first virtual server, 10.0.0.120 in this example.

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

To restore manufacturer's settings – at the console use the command **lbrestore** or in the WUI goto *Maintenance* > *Backup & Restore* > *Restore Manufacturer's Defaults*. This will set the address to 192.168.2.21 if this address is available.

Additional Appliance Configuration Using The Web Interface

When using a Clustered Pair, all configuration must be done via the master unit, the slave unit will then be synchronized automatically. If for some reason the master is down and changes are reqired to the setup, please contact support@loadbalancer.org for advice

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

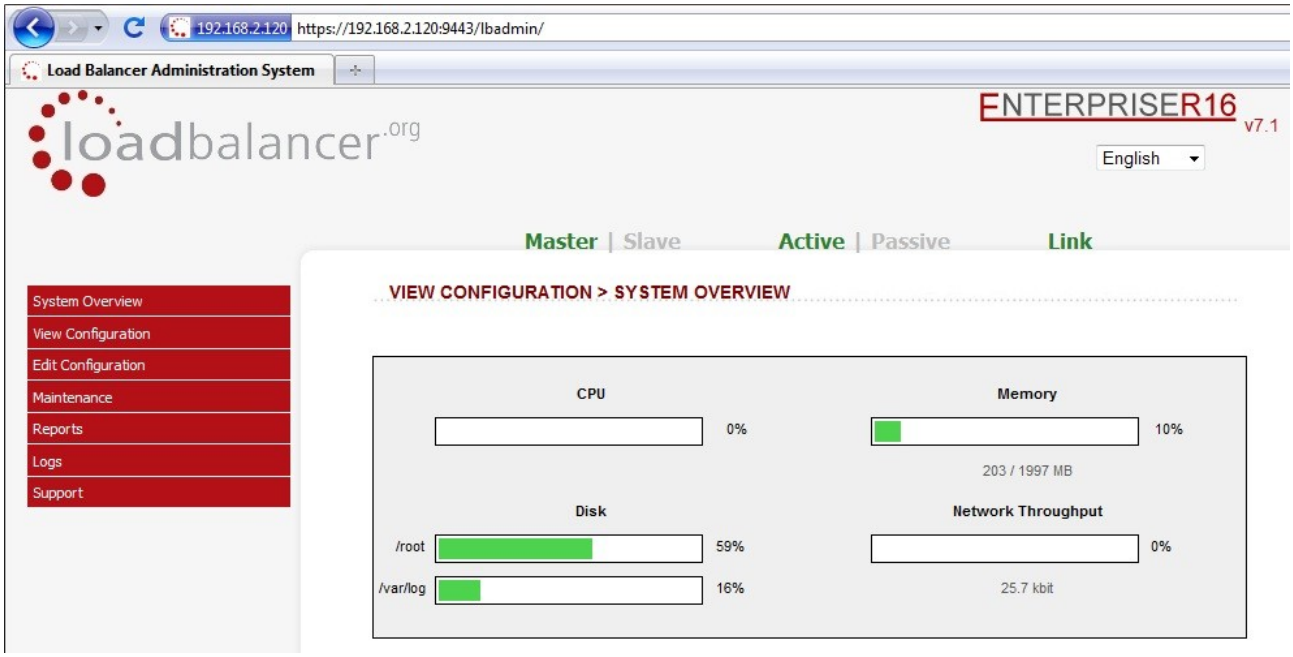
If you have already used the web based wizard, then you will already be using the WUI. From here all administration tasks can be carried out. If not, access the WUI as follows:

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

(replace 192.168.2.21 with the correct address)

log in to the WUI: ***Username***: loadbalancer
Password: loadbalancer

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

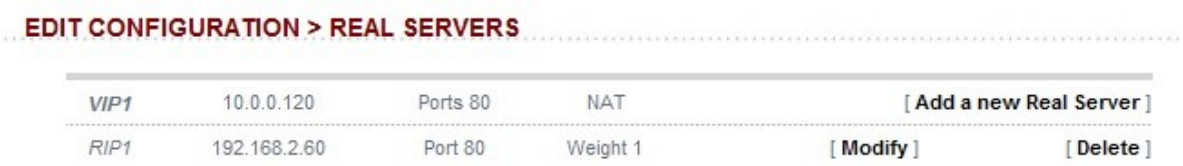


All administration tasks can be carried out through the web interface.

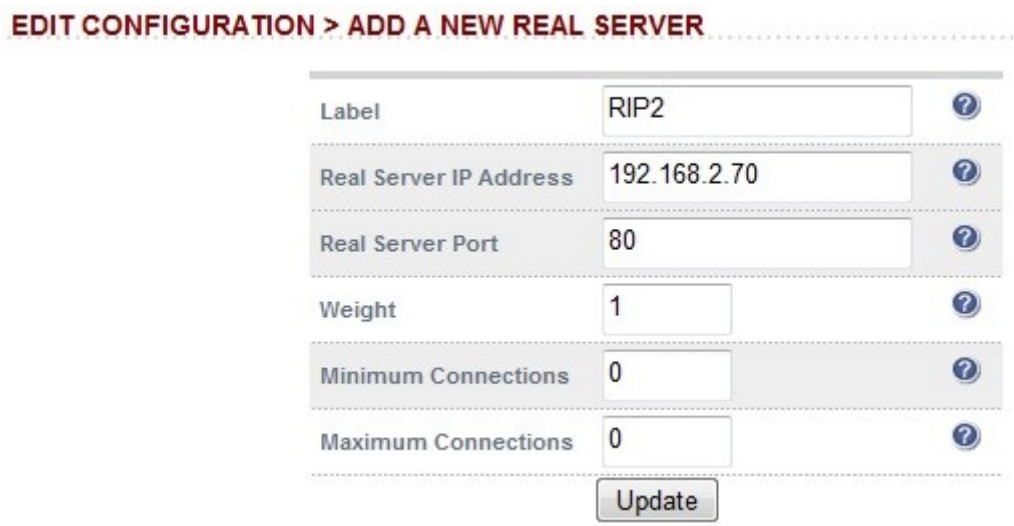
Adding additional real servers

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

- Use *Edit Configuration* > *Layer 4 Configuration* > *Real Servers* , you'll see the first Real Server that was created by the wizard



- Click [**Add a new Real Server**]



- Enter the label, IP address and port number of your additional real server
- The weight defaults to 1 making the real server active immediately and equal weight to the first real server added by the wizard. If the real servers have different performance specifications, then the weight can be adjusted – a higher number means more traffic is sent to that server
- Leave the minimum & maximum connections as 0 for unrestricted

Configuring the Real Servers

Depending on the deployment method (DR, NAT or SNAT) used, the actual physical servers may need to be configured to allow the load balancer to operate correctly. The following sections define what is needed for the various modes.

Configuring the real servers 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. Normally, a floating IP address is added using *Edit Configuration* > *Floating IPs.* This is important when a master / slave configuration is used to allow failover & failback of the default gateway address.

Failure to correctly configure the real servers default gateway is the most common mistake when using NAT mode

Configuring the real servers for DR mode (Linux)

If you are using the one-arm DR load balancing method, each real server requires the ARP problem to be solved. All real servers 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, whilst other traffic such as health-checks, administration traffic etc. use the real server's IP address.

Detecting the ARP problem

You can use *Reports* > *Layer 4 Current Connections* to check whether the ARP problem has been solved. If not, the connection state will be SYN_RECV as shown below when a client connection to the VIP is attempted:

Solving for Linux – method 1 (using iptables)

You can use iptables (netfilter) on each real server to re-direct incoming packets destined for the virtual server IP address. To make this permenant, simply add the command to an appropriate start-up script such as /etc/rc.local. If the real server is serving multiple VIPs, add additional iptables rules for each VIP.

```
iptables -t nat -A PREROUTING -p tcp -d <VIP> -j REDIRECT
```

e.g.

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

this means redirect any incoming packets destined for 10.0.0.21 (the virtual server) locally.

Method 1 does not work with IPv6 Virtual Servers, use method 2 below instead

Solving for Linux – method 2 (using arp_ignore sysctl values)

This is the preferred method as it supports both IPv4 and IPv6. Each real server needs the loopback adapter to be configured with the Virtual Servers IP address. This address must not respond to ARP requests and the web server also needs to be configured to respond to this address. To set this up follow steps 1-3 below.

Step 1 : re-configure ARP on the real servers (this step can be skipped for IPv6 virtual servers)

To do this add the following lines to /etc/sysctl.conf:

```
net.ipv4.conf.all.arp_ignore=1
net.ipv4.conf.eth0.arp_ignore=1
net.ipv4.conf.eth1.arp_ignore=1
net.ipv4.conf.all.arp_announce=2
net.ipv4.conf.eth0.arp_announce=2
net.ipv4.conf.eth1.arp_announce=2
```

Step 2 : apply these settings

Either reboot the real server or run the following command to apply these settings:

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

Step 3 : add the virtual servers' IP address to the loopback adapter

run the following command for each VIP. To make this permanent, simply add the command to an appropriate startup script such as /etc/rc.local.

```
ip addr add dev lo <IPv4-VIP>/32
```

for IPv6 addresses use:

```
ip addr add dev lo <IPv6-VIP>/128
```

Alternatively, modify the appropriate interface script to add the additional IP address(es).

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

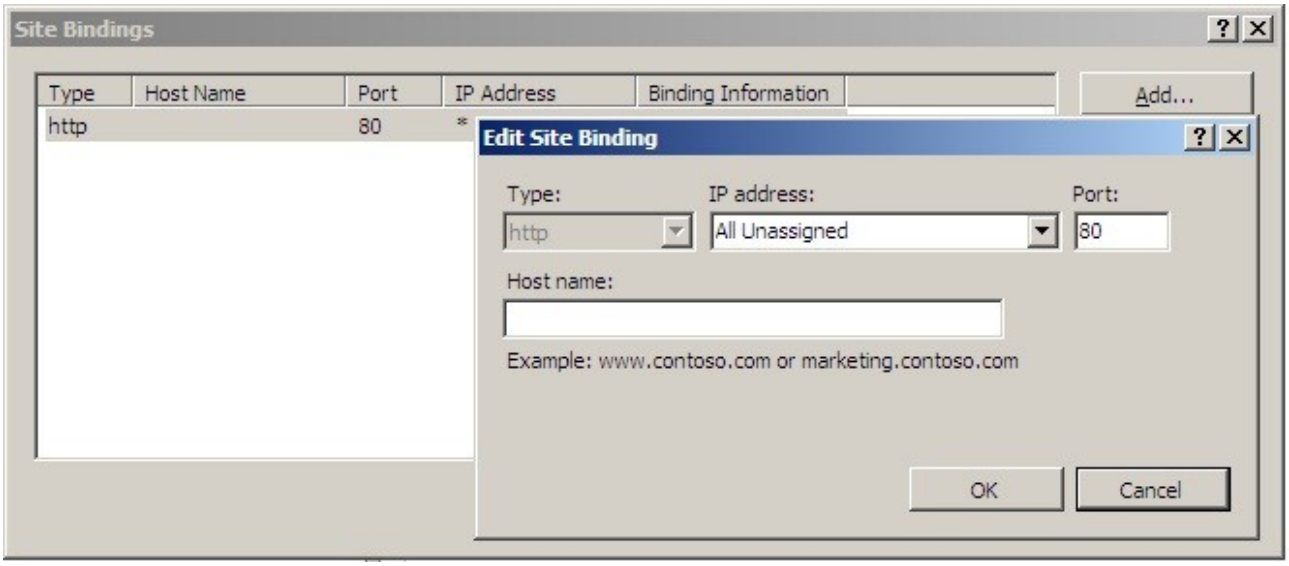
Configuring the real servers for DR mode (Windows)

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

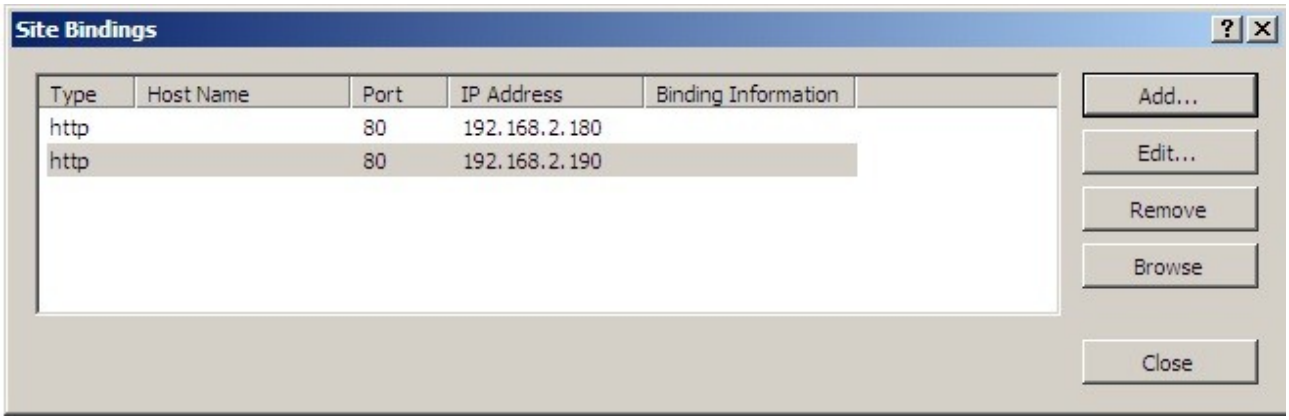
- for all real servers in Direct Routing mode the load balanced application must respond to both the virtual IP as well as the servers real IP. With Windows IIS the IP address must either be set to (All Unassigned) or use the Advanced tab to add a second IP address as shown below
- Each real server must have the Microsoft loopback adapter installed and configured
- The Microsoft loopback adapter must be configured to deal with the ARP problem

Configuring IIS to respond to both the RIP and VIP

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



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

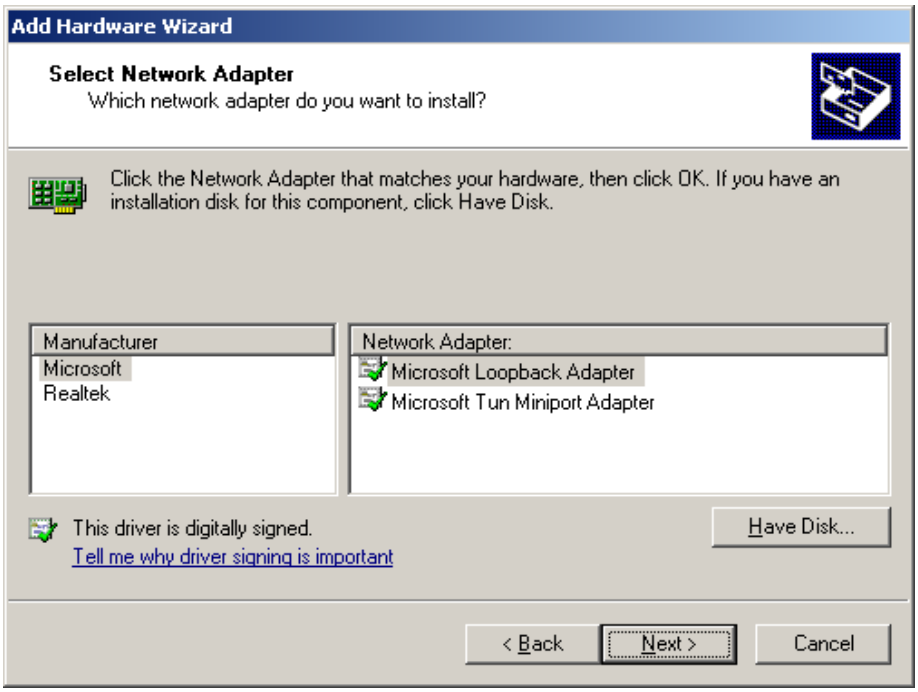


Resolving ARP issues for Windows server 2000 / 2003 (DR mode only)

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

Installing the Microsoft loopback adapter

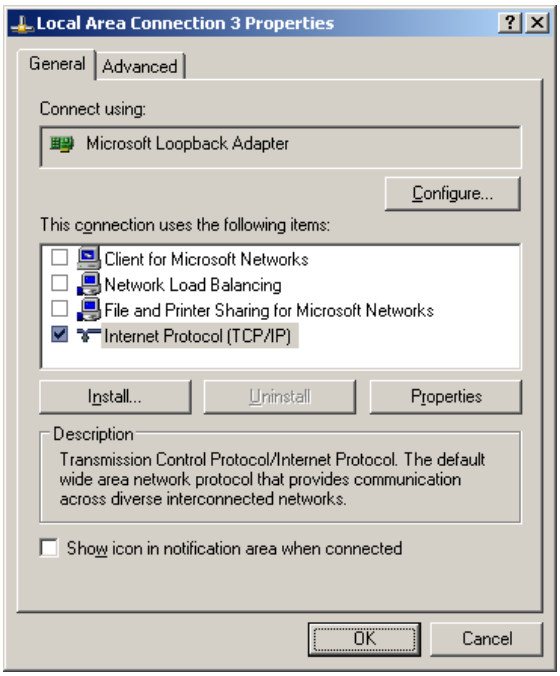
1. Open the Control Panel and double-click Add Hardware
2. Once the Hardware Wizard opens, click Next
3. Select 'Yes, I have already connected the hardware', click Next
4. Scroll to the bottom of the list, select 'Add a new hardware device' and click Next
5. Select 'Install the hardware that I manually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



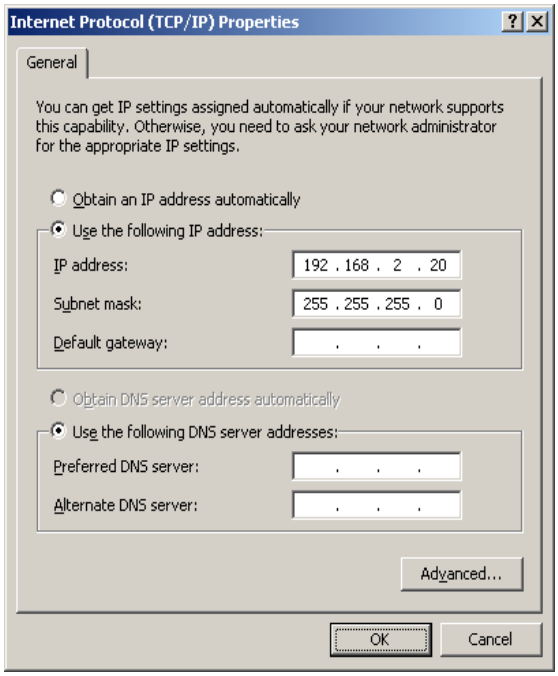
8. Click Next to start the installation, when complete click Finish

Configuring the loopback adapter

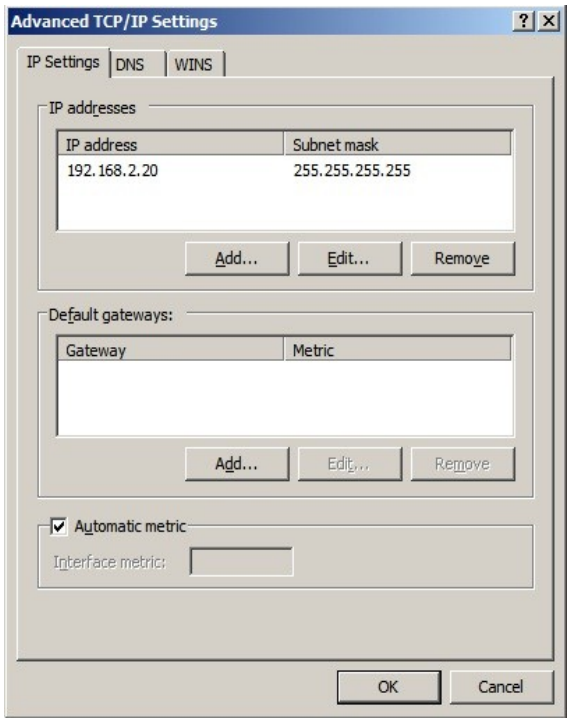
1. Open the Control Panel and double-click Network Connections
2. Right click the new loopback adapter and select properties



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



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



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

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

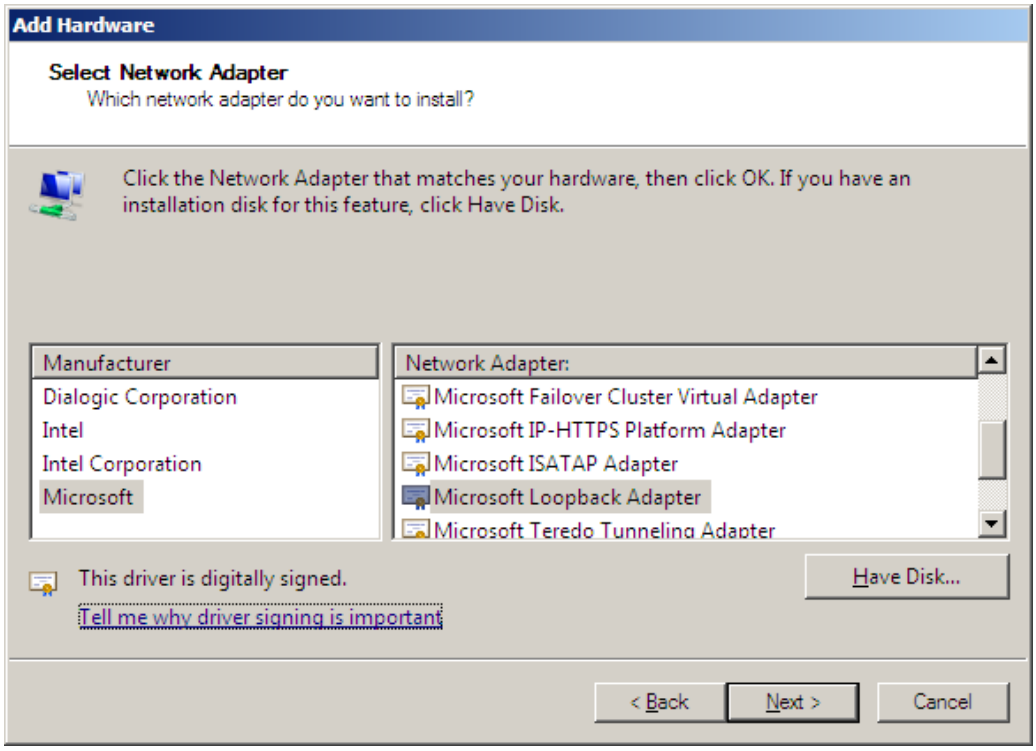
For Windows server 2003 SP1 & above, if you have enabled the built-in firewall, you will need to create an exception to enable access to the web server. This exception by default will allow traffic on both the network and loopback adapters

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

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

Installing the Microsoft loopback adapter

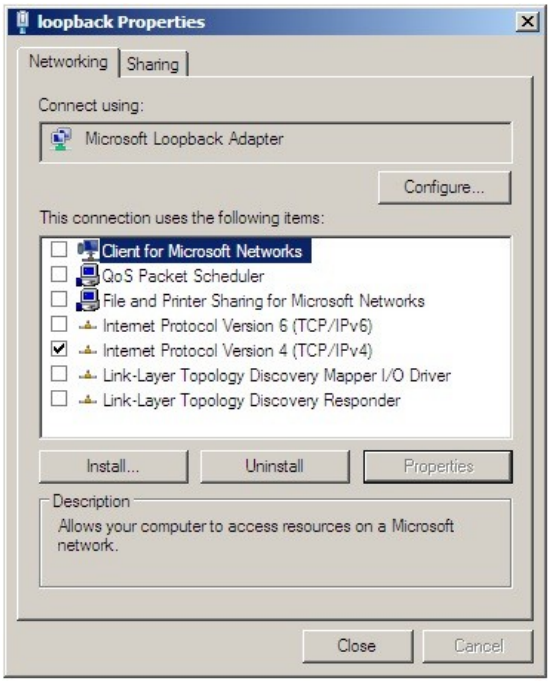
1. Click Start, select Run and enter ***hdwwiz*** to start the Hardware Installation Wizard
2. When the Wizard has started, click Next
3. Select 'Install the hardware that I manually select from a list (Advanced)', click Next
4. Select 'Network adapters', click Next
5. Select 'Microsoft' & 'Microsoft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



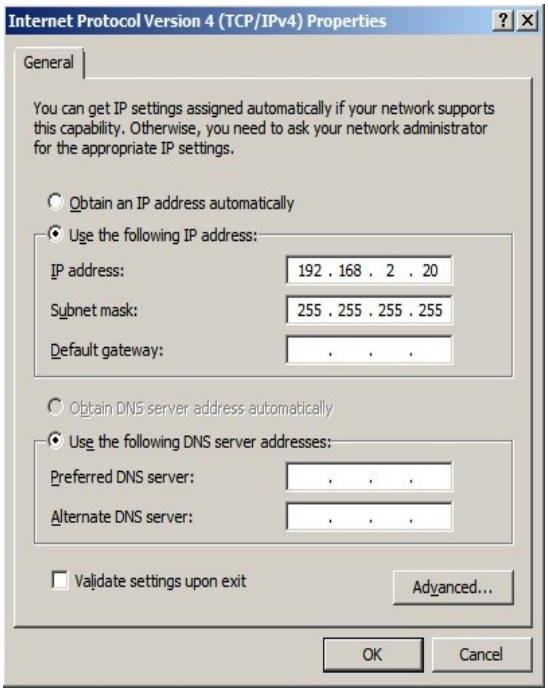
6. Click Next to start the installation, when complete click Finish

Configuring the loopback adapter

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



4. Un-check all items except Internet Protocol Version 4 (TCP/IPv4)
5. Select Internet Protocol Version (TCP/IPv4), click Properties and configure the IP address to be the same as the Virtual Server (VIP) with a full subnet mask, e.g. 192.168.2.20/32



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

7. For Windows 2008, its not necessary to modify the interface metric on the advanced tab and should be left set to Automatic

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

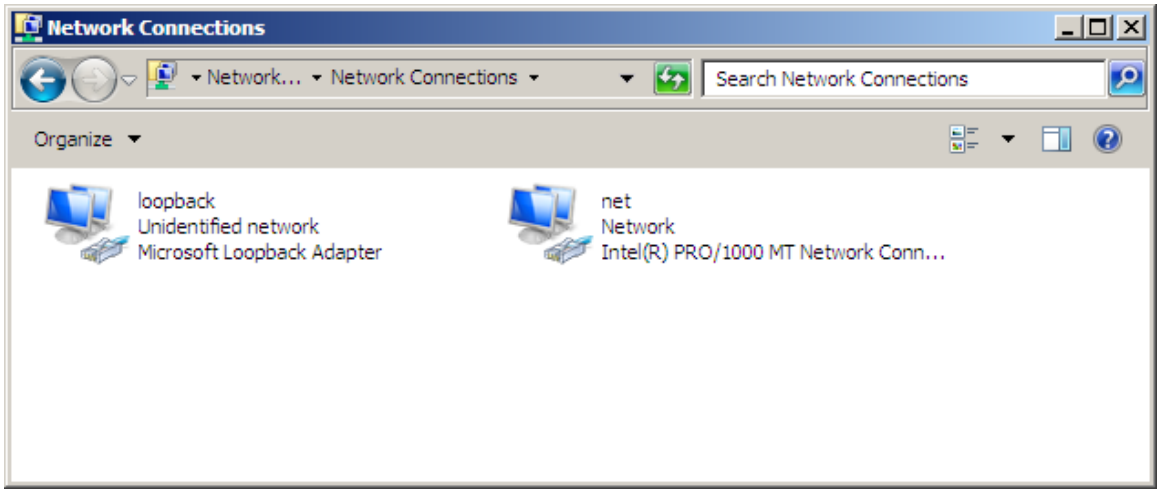
Configuring strong / weak host behavior

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

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

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

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



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

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

```
netsh interface ipv4 show interface
```

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

Verifying netsh Settings

To verify that settings have been configured correctly, run the following command on the Windows server to clearly list the settings that have been applied to the interface:

```
netsh interface ipv4 show interface <interface name>
```

i.e. for the loopback adapter run : `netsh interface ipv4 show interface loopback`
i.e. for the Net adapter run : `netsh interface ipv4 show interface net`

e.g.

```
C:\Users\Administrator>netsh interface ipv4 show interface loopback

Interface loopback Parameters
----------------------------------------------
IfLuid                                 : ethernet_9
IfIndex                                : 15
State                                  : connected
Metric                                 : 30
Link MTU                               : 1500 bytes
Reachable Time                         : 28500 ms
Base Reachable Time                    : 30000 ms
Retransmission Interval                : 1000 ms
DAD Transmits                          : 3
Site Prefix Length                     : 64
Site Id                                : 1
Forwarding                             : disabled
Advertising                            : disabled
Neighbor Discovery                     : enabled
Neighbor Unreachability Detection      : enabled
Router Discovery                       : dhcp
Managed Address Configuration          : enabled
Other Stateful Configuration           : enabled
Weak Host Sends                        : enabled
Weak Host Receives                     : enabled
Use Automatic Metric                   : enabled
Ignore Default Routes                  : disabled
Advertised Router Lifetime             : 1800 seconds
Advertise Default Route                : disabled
Current Hop Limit                      : 0
Force ARPND Wake up patterns           : disabled
Directed MAC Wake up patterns          : disabled


C:\Users\Administrator>
```

the above screen shot shows that the settings have been applied correctly.

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

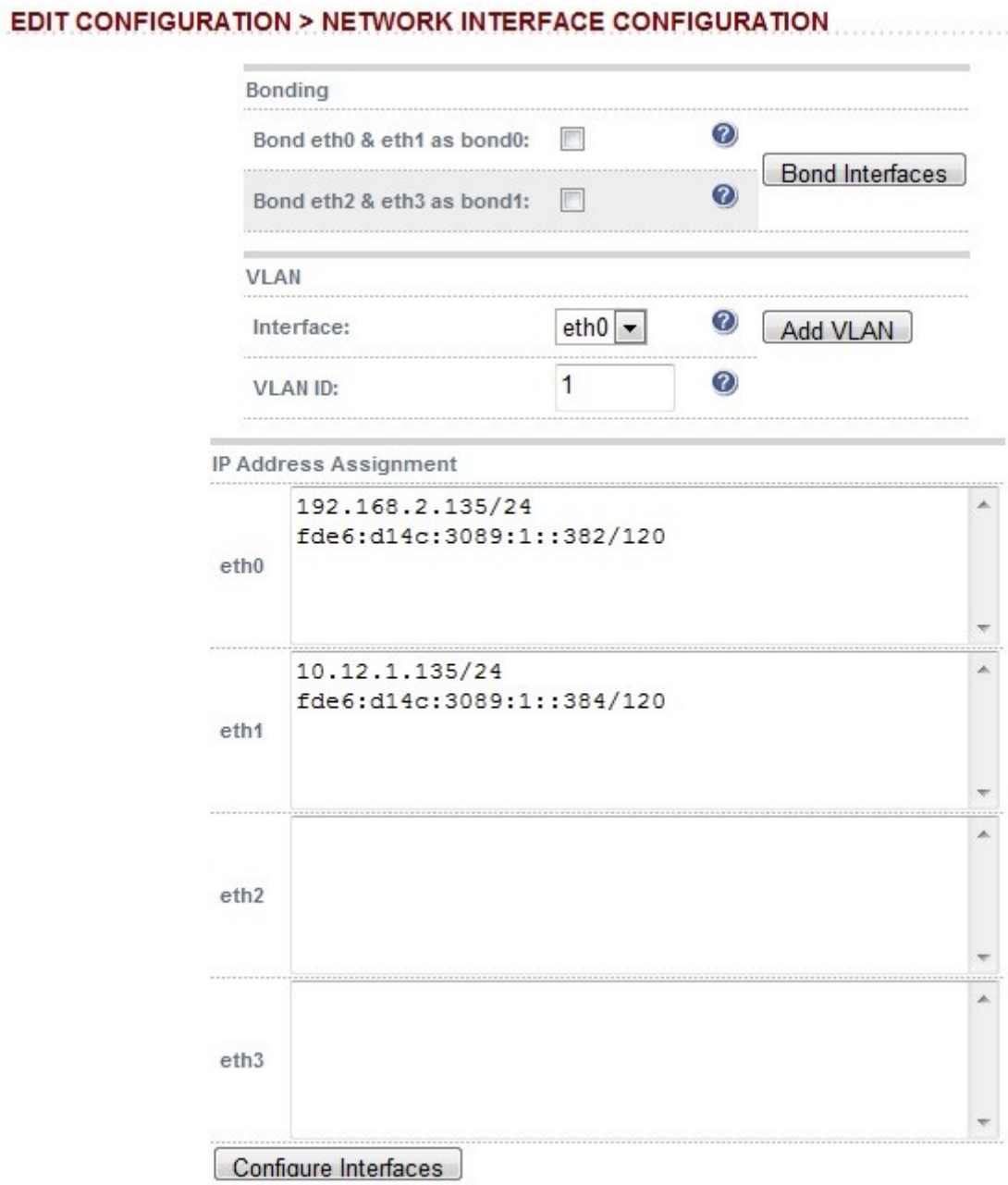
Failure to correctly configure the real servers to handle the ARP problem is the most common mistake in DR configurations

Configuring the real server for SNAT mode

When using Layer7 (HAproxy) Virtual Servers, no changes are required to the real servers.

IPv6 Support

New to v7.x is full IPv6 support. This allows Virtual Servers to be configured using IPv6 addresses. Its also possible to mix IPv4 and IPv6 addresses on a single appliance as illustrated below:



Testing The Load Balancer Configuration

For testing, add a page to each real web servers root directory e.g. test.html and put the server name on this page for easy identification during your tests.

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

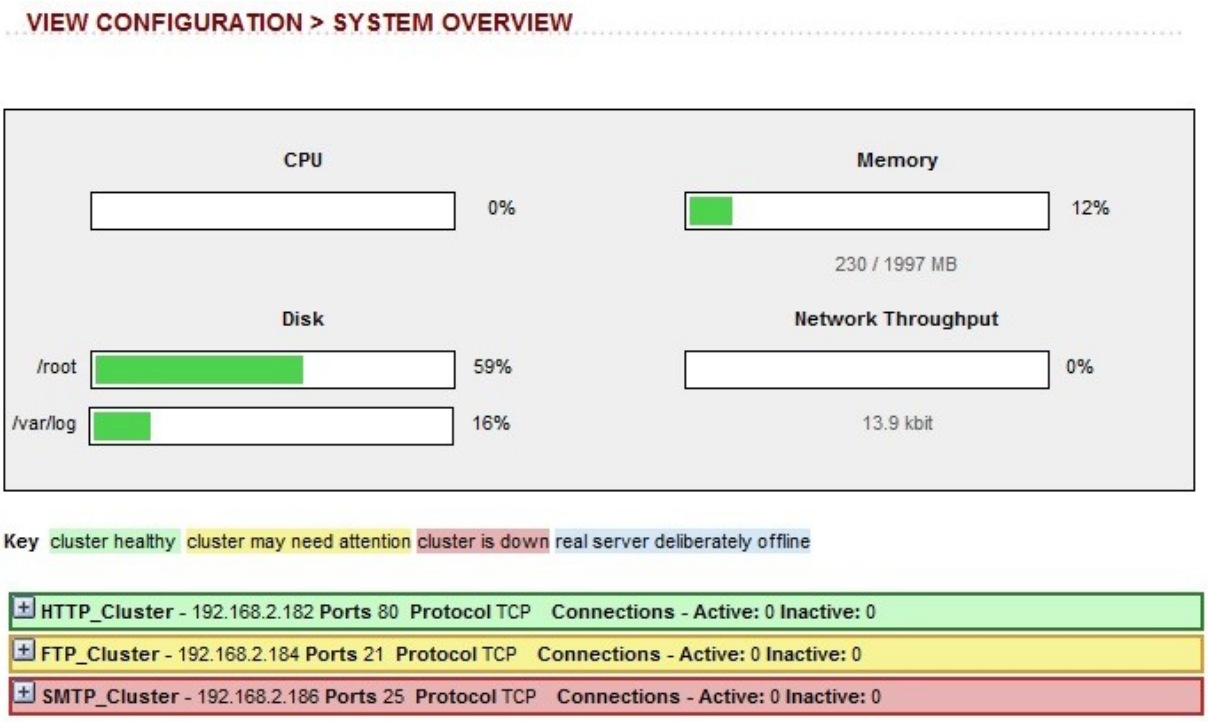
When using a two-arm NAT load balancing method the test client must be in the external subnet

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 *System Overview* and make sure none of your VIPs are highlighted in red. If they are, your cluster is down. Green indicates a healthy cluster, yellow indicates that your cluster may need attention (one real server may be down), and blue indicates a real server has been deliberately taken offline



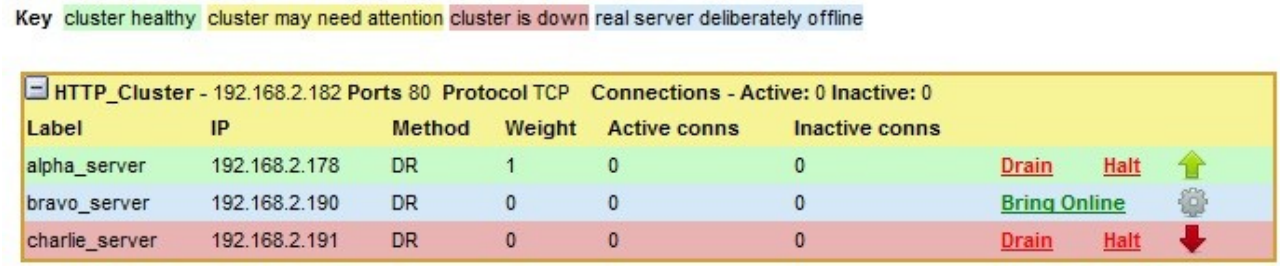
3. If the VIP is still not working then check *Reports* > *Current Connections* to see the current traffic in detail, any packets marked SYN_RECV imply incorrect real server configuration. Check that any NAT mode servers have the correct default gateway and any DR mode servers are responding to the VIP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

Health check diagnosis

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

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

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



'***alpha_server***' is green which indicates that the server is operating normally.

'***bravo_server***' is blue, this indicates that it is deliberately in maintenance mode. You can use 'Bring Online' to make it active.

'***charlie_server***' is down (red). This implies that the real server has failed a health check; you can investigate this using *Logs* > *Layer 4*. If you know the real server should be active, you may need to increase the health check time-out *Edit Configuration* > *Layer 4 – Advanced Settings* or *Layer 7 – Advanced Settings.*

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

To test fail-over of a clustered pair, once fully configured power down the master and check that the slave unit takes over all the floating IP(s). If fail-over to the slave unit does not occur correctly, check *Logs* > *Heartbeat* on both nodes for any errors.

It is very important to verify that master / slave failover occurs correctly before going live. This proves the resilience of the cluster and makes you aware of the failover / failback process. Please refer to page 116 (admin guide) for details of the hb_takeover command which can be used to force a failover and refer to page 128 for detailed steps on verifying failover / failback

When testing load balancer fail-over, don't just pull the serial cable and network cable out. This will not cause a fail-over but will cause a split brain (I.e. both units active) to occur. You can configure fail-over on network failure but this is not enabled by default. To enable this, a ping node must be configured under *Edit Configuration* > *Heartbeat Configuration*

New to v7.x is the role status at the top of each screen. For a working pair, the normal view is shown below:

This shows that the master unit is active and that the heartbeat link is up between master & slave. Other states:

Master | Slave	**Active** | Passive	Link	this is a master unit, it's active, no slave unit has been defined
Master | Slave	**Active** | Passive	**Link**	this is a master unit, it's active, a slave has been defined but the link to the slave is down. ***Action***: *check & verify the heartbeat configuration*
Master | **Slave**	**Active** | Passive	**Link**	this is a slave unit, it's active (a failover from the master has occurred) and the heartbeat link to the master has been established
Master | Slave	**Active** | **Passive**	**Link**	this is a master unit, a slave unit has been defined, but the link is down (e.g. serial cable unplugged) so the state cannot be determined. In this case the floating IP's may be active on both units. ***Action***: *check & verify the heartbeat configuration, check the serial cable (if applicable), check heatbeat logs & if required restart heartbeat on both units*
Master | Slave	Active | Passive	**Link**	this is the master unit, a slave unit has been defined on the master, but the link is down (e.g. serial cable unplugged) so the state cannot be determined. In this case the heartbeat service has probably stopped on both units. ***Action***: *check & verify the heartbeat configuration, check the serial cable (if applicable), check heatbeat logs & if required restart heartbeat on both units*

NOTE: ***Restarting heartbeat will cause a temporary outage of all load balanced services***

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***

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.

Persistence

Persistence is a feature that is required by many web applications. Once a user has interacted with a particular server all subsequent requests are sent to the same server thus persisting to that particular server. It is normally required when the session state is stored locally to the web server as opposed to a database.

What do you do if your application is not stateless?

Some applications require state to be maintained such as:

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

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

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

Loadbalancer.org persistence methods

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

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

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

Loadbalancer.org Technical Support

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

For more detailed explanations and complex configuration details please refer to our full administration manual which is available at: http://www.loadbalancer.org/pdffiles/loadbalanceradministrationv7.pdf