# Table of Contents

1. About this Guide ............................................................................. 3
2. Loadbalancer.org Appliances Supported ........................................... 3
3. Loadbalancer.org Software Versions Supported ................................. 3
4. RabbitMQ Software Versions Supported ........................................... 3
5. RabbitMQ ..................................................................................... 3
6. Load Balancing RabbitMQ ............................................................... 3
   Persistence (aka Server Affinity) ....................................................... 3
   Virtual Service (VIP) Requirements .................................................. 3
   Server Feedback Agent .................................................................... 4
7. Deployment Concept ....................................................................... 4
8. Load Balancer Deployment Methods ............................................... 4
   Layer 4 DR Mode ........................................................................... 4
   Layer 7 SNAT Mode ....................................................................... 5
   Our Recommendation ...................................................................... 6
9. Configuring RabbitMQ for Load Balancing ...................................... 7
   Server Configuration ...................................................................... 7
   Layer 4 DR Mode – Solving the ARP Problem ................................. 7
10. Loadbalancer.org Appliance – the Basics ....................................... 7
    Virtual Appliance .......................................................................... 7
    Initial Network Configuration ......................................................... 7
    Accessing the WebUI ...................................................................... 7
    Main Menu Options ...................................................................... 9
    HA Clustered Pair Configuration .................................................... 10
11. Appliance Configuration for RabbitMQ – Using Layer 4 DR Mode .... 10
    Configuring the Virtual Service (VIP) ............................................. 10
    Defining the Real Servers (RIPs) ..................................................... 10
12. Appliance Configuration for RabbitMQ – Using Layer 7 SNAT Mode ... 11
    Configuring the Virtual Service (VIP) ............................................. 11
    Defining the Real Servers (RIPs) ..................................................... 11
    Finalizing the Configuration ........................................................... 12
13. Testing & Verification .................................................................... 12
    Using System Overview ................................................................ 12
14. Technical Support .......................................................................... 12
15. Further Documentation .................................................................. 13
16. Conclusion .................................................................................... 13
17. Appendix ...................................................................................... 14
    Server Feedback Agent ................................................................ 14
    Windows Agent ............................................................................ 14
    Linux/Unix Agent ......................................................................... 16
    Custom HTTP Agent ..................................................................... 17
    Configuring HA - Adding a Secondary Appliance ........................... 17
    Solving the ARP Problem .............................................................. 20
    Solving the ARP Problem for Linux .............................................. 20
    Windows Server 2012, 2016 & 2019 ................................................. 22
18. Document Revision History ............................................................ 28
1. About this Guide

This guide details the steps required to configure a load balanced RabbitMQ environment utilizing Loadbalancer.org appliances. It covers the configuration of the load balancers and also any RabbitMQ configuration changes that are required to enable load balancing.

For more information about initial appliance deployment, network configuration and using the Web User Interface (WebUI), please also refer to the Administration Manual.

2. Loadbalancer.org Appliances Supported

All our products can be used for load balancing RabbitMQ. For full specifications of available models please refer to https://www.loadbalancer.org/products.

Some features may not be supported in all cloud platforms due to platform specific limitations, please check with Loadbalancer.org support for further details.

3. Loadbalancer.org Software Versions Supported

- V8.3.8 and later

4. RabbitMQ Software Versions Supported

- RabbitMQ – all versions

5. RabbitMQ

RabbitMQ is an open source message broker. It uses a publish-subscribe model to route data from publishers to consumers. It is scalable and can be load balanced, acting as a reliable and highly available intermediary. It has support for management and monitoring, and has a range of tools and plugins available.

6. Load Balancing RabbitMQ

Note: It’s highly recommended that you have a working RabbitMQ environment first before implementing the load balancer.

Persistence (aka Server Affinity)

RabbitMQ does not require session affinity at the load balancing layer by default.

Virtual Service (VIP) Requirements

To provide load balancing and HA for RabbitMQ, one virtual service is required.

The virtual service must be set to listen on the same port as the RabbitMQ service, which listens on port 5672 by default.

Our recommended configuration uses a layer 4 DR mode VIP. Using a layer 7 SNAT mode VIP is also supported.
Server Feedback Agent

It may be useful to adjust how much traffic is passed to the RabbitMQ servers depending on their CPU load. This can be done by installing the Loadbalancer.org server feedback agent on each RabbitMQ server and then re-configuring the Virtual Service to make use of the agent. The feedback agent is available for both Linux and Windows servers.

Please refer to Server Feedback Agent for full details on installing and configuring the server feedback agent.

7. Deployment Concept

VIPs = Virtual IP Addresses

Note: The load balancer can be deployed as a single unit, although Loadbalancer.org recommends a clustered pair for resilience & high availability. Please refer to Configuring HA - Adding a Secondary Appliance for more details on configuring a clustered pair.

8. Load Balancer Deployment Methods

The load balancer can be deployed in one of 4 fundamental ways: Layer 4 DR mode, Layer 4 NAT mode, Layer 4 SNAT mode, or Layer 7 SNAT mode. For RabbitMQ, layer 4 DR mode is recommended. Layer 7 SNAT mode is also supported. Both supported modes are described below.

Layer 4 DR Mode

One-arm direct routing (DR) mode is a very high performance solution that requires little change to your existing infrastructure.

Note: Kemp, Brocade, Barracuda & A10 Networks call this Direct Server Return and F5 call it N-Path.
DR mode works by changing the destination MAC address of the incoming packet to match the selected Real Server on the fly which is very fast.

When the packet reaches the Real Server it expects the Real Server to own the Virtual Services IP address (VIP). This means that you need to ensure that the Real Server (and the load balanced application) respond to both the Real Server’s own IP address and the VIP.

The Real Servers should not respond to ARP requests for the VIP. Only the load balancer should do this. Configuring the Real Servers in this way is referred to as Solving the ARP Problem. For more information please refer to DR Mode Considerations.

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.

The load balancer must have an Interface in the same subnet as the Real Servers to ensure layer 2 connectivity required for DR mode to work.

The VIP can be brought up on the same subnet as the Real Servers, or on a different subnet provided that the load balancer has an interface in that subnet.

Port translation is not possible with DR mode, e.g. VIP:80 → RIP:8080 is not supported.

DR mode is transparent, i.e. the Real Server will see the source IP address of the client.

Layer 7 SNAT Mode

Layer 7 SNAT mode uses a proxy (HAProxy) at the application layer. Inbound requests are terminated on the load balancer and HAProxy generates a new corresponding request to the chosen Real Server. As a result, Layer 7 is typically not as fast as the Layer 4 methods. Layer 7 is typically chosen when either enhanced options such as SSL termination, cookie based persistence, URL rewriting, header insertion/deletion etc. are required, or when the network topology prohibits the use of the layer 4 methods.
Layer 7 SNAT mode is a full proxy and therefore load balanced Real Servers do not need to be changed in any way.

Because layer 7 SNAT mode is a full proxy, any server in the cluster can be on any accessible subnet including across the Internet or WAN.

Layer 7 SNAT mode 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 balancer’s own IP address by default, or any other local appliance IP address if preferred (e.g. the VIP address). This can be configured per layer 7 VIP.

If required, the load balancer can be configured to provide the actual client IP address to the Real Servers in 2 ways. Either by inserting a header that contains the client’s source IP address, or by modifying the Source Address field of the IP packets and replacing the IP address of the load balancer with the IP address of the client. For more information on these methods please refer to Transparency at Layer 7.

Layer 7 SNAT mode can be deployed using either a one-arm or two-arm configuration.

Port translation is possible with Layer 7 SNAT mode, e.g. VIP:80 → RIP:8080 is supported.

You should not use the same RIP:PORT combination for layer 7 SNAT mode VIPs and layer 4 SNAT mode VIPs because the required firewall rules conflict.

Our Recommendation
Where possible we recommend that Layer 4 Direct Routing (DR) mode is used. This mode offers the best possible performance since replies go directly from the Real Servers to the client, not via the load balancer. It’s also relatively simple to implement. Ultimately, the final choice does depend on your specific requirements and infrastructure.

If DR mode cannot be used, for example if the real servers are located in remote routed networks, then SNAT mode is recommended.

If the load balancer is deployed in AWS or Azure, layer 7 SNAT mode must be used as layer 4 direct routing is not currently possible on these platforms.
9. Configuring RabbitMQ for Load Balancing

Server Configuration
RabbitMQ servers need to be configured for load balancing and high availability. This configuration is specific to the RabbitMQ service and is beyond the scope of this deployment guide.

Please refer to the following documentation on the RabbitMQ website which details the configuration that is required: https://www.rabbitmq.com/ha.html

Layer 4 DR Mode – Solving the ARP Problem
If using layer 4 DR mode, the ‘ARP problem’ must be solved on each real server for DR mode to work. For detailed steps on solving the ARP problem for Linux and Windows, please refer to Solving the ARP Problem for more information.

For a detailed explanation of DR mode and the nature of the ARP problem, please refer to Layer 4 DR Mode.

10. Loadbalancer.org Appliance – the Basics

Virtual Appliance
A fully featured, fully supported 30 day trial is available if you are conducting a PoC (Proof of Concept) deployment. The VA is currently available for VMware, Virtual Box, Hyper-V, KVM, XEN and Nutanix AHV and has been optimized for each Hypervisor. By default, the VA is allocated 2 vCPUs, 4GB of RAM and has a 20GB virtual disk. The Virtual Appliance can be downloaded here.

Note
The same download is used for the licensed product, the only difference is that a license key file (supplied by our sales team when the product is purchased) must be applied using the appliance’s WebUI.

Note
Please refer to The Virtual Appliance - Hypervisor Deployment and the ReadMe.txt text file included in the VA download for more detailed information on deploying the VA using various Hypervisors.

Note
For the VA, 4 NICs are included but only eth0 is connected by default at power up. If the other NICs are required, these should be connected using the network configuration screen within the Hypervisor.

Initial Network Configuration
After boot up, follow the instructions on the console to configure the IP address, subnet mask, default gateway, DNS and other network settings.

Important
Be sure to set a secure password for the load balancer, when prompted during the setup routine.

Accessing the WebUI
The WebUI is accessed using a web browser. Appliance authentication is based on Apache .htaccess files. User admin tasks such as adding users and changing passwords can be performed using the WebUI menu option:
A number of compatibility issues have been found with various versions of Internet Explorer and Edge. The WebUI has been tested and verified using both Chrome & Firefox.

If required, users can also be authenticated against LDAP, LDAPS, Active Directory or Radius. For more information please refer to External Authentication.

1. Using a browser, access the WebUI using the following URL:


2. Log in to the WebUI:

   Username: loadbalancer
   Password: <configured-during-network-setup-wizard>

   To change the password, use the WebUI menu option: Maintenance > Passwords.

Once logged in, the WebUI will be displayed as shown below:
The WebUI for the VA is shown, the hardware and cloud appliances are very similar. The yellow licensing related message is platform & model dependent.

3. You'll be asked if you want to run the Setup Wizard. If you click **Accept** the Layer 7 Virtual Service configuration wizard will start. If you want to configure the appliance manually, simple click **Dismiss**.

**Main Menu Options**

- **System Overview** - Displays a graphical summary of all VIPs, RIPs and key appliance statistics
- **Local Configuration** - Configure local host settings such as IP address, DNS, system time etc.
- **Cluster Configuration** - Configure load balanced services such as VIPs & RIPs
- **Maintenance** - Perform maintenance tasks such as service restarts and taking backups
- **View Configuration** - Display the saved appliance configuration settings
- **Reports** - View various appliance reports & graphs
HA Clustered Pair Configuration

Loadbalancer.org recommend that load balancer appliances are deployed in pairs for high availability. In this guide a single unit is deployed first, adding a secondary unit is covered in Configuring HA - Adding a Secondary Appliance.

11. Appliance Configuration for RabbitMQ – Using Layer 4 DR Mode

Configuring the Virtual Service (VIP)

1. Using the web user interface, navigate to Cluster Configuration > Layer 4 – Virtual Services and click on Add a new Virtual Service.

2. Define the Label for the virtual service as required, e.g. RabbitMQ HA.

3. Set the Virtual Service IP Address field to the required IP address, e.g. 192.168.87.5.

4. Set the Ports field to the port that the RabbitMQ service is listening on, which by default is port 5672.

5. Leave the Protocol set to TCP.


7. Click Update to create the virtual service.

---

### Layer 4 - Add a new Virtual Service

<table>
<thead>
<tr>
<th>Label</th>
<th>RabbitMQ HA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Service</td>
<td>192.168.87.5</td>
</tr>
<tr>
<td>IP Address</td>
<td></td>
</tr>
<tr>
<td>Ports</td>
<td>5672</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP</td>
</tr>
<tr>
<td>Forwarding Method</td>
<td>Direct Routing</td>
</tr>
</tbody>
</table>

---

Defining the Real Servers (RIPs)

1. Using the web user interface, navigate to Cluster Configuration > Layer 4 – Real Servers and click on Add a new Real Server next to the newly created VIP.

2. Define the Label for the real server as required, e.g. Rabbit1.

3. Set the Real Server IP Address field to the required IP address, e.g. 192.168.87.10.

4. Click Update.

5. Repeat these steps to add additional RabbitMQ servers as required.
12. Appliance Configuration for RabbitMQ – Using Layer 7 SNAT Mode

Configuring the Virtual Service (VIP)
1. Using the web user interface, navigate to Cluster Configuration > Layer 7 – Virtual Services and click on Add a new Virtual Service.
2. Define the Label for the virtual service as required, e.g. RabbitMQ HA.
3. Set the Virtual Service IP Address field to the required IP address, e.g. 192.168.2.5.
4. Set the Ports field to the port that the RabbitMQ service is listening on, which by default is port 5672.
5. Set the Layer 7 Protocol to TCP Mode.
6. Click Update to create the virtual service.

Defining the Real Servers (RIPs)
1. Using the web user interface, navigate to Cluster Configuration > Layer 7 – Real Servers and click on Add a new Real Server next to the newly created VIP.
2. Define the Label for the real server as required, e.g. Rabbit1.
3. Set the Real Server IP Address field to the required IP address, e.g. 192.168.2.10.
4. Leave the Real Server Port field blank.
5. Click Update.
6. Repeat these steps to add additional RabbitMQ servers as required.

### Finalizing the Configuration

To apply the new settings, HAProxy must be restarted as follows:

1. Using the WebUI, navigate to: Maintenance > Restart Services and click Restart HAProx.

### 13. Testing & Verification

#### Using System Overview

The System Overview can be viewed in the WebUI. It shows a graphical view of all VIPs & RIPS (i.e. the RabbitMQ Nodes) and shows the state/health of each server as well as the state of the cluster as a whole.

This first example, shown below, shows a layer 4 DR mode VIP where all RabbitMQ nodes are healthy and available to accept connections.

This second example, shown below, shows a layer 7 SNAT mode VIP where all RabbitMQ nodes are healthy and available to accept connections.

### 14. Technical Support

For more details about configuring the appliance and assistance with designing your deployment please don’t hesitate to contact the support team using the following email address: support@loadbalancer.org.
15. Further Documentation


16. Conclusion

Loadbalancer.org appliances provide a very cost effective solution for highly available load balanced RabbitMQ environments.
17. Appendix

Server feedback Agent

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. For layer 4 VIPs the feedback method can be set to either agent or HTTP, for Layer 7 VIPs, only the agent method is supported.

A telnet to port 3333 on a Real Server with the agent installed will return the current idle stats as an integer value in the range 0 - 100. The figure returned can be related to CPU utilization, RAM usage or a combination of both. This can be configured using the XML configuration file located in the agents installation folder (by default C:\ProgramData\LoadBalancer.org\LoadBalancer).

The load balancer typically expects a 0-99 integer response from the agent which by default relates to the current CPU idle state, e.g. a response of 92 would imply that the Real Servers CPU is 92% idle. The load balancer will then use the formula (92/100*requested_weight) to find the new optimized weight.

**Note**

The ‘Requested Weight’ is the weight set in the WebUI for each Real Server.

For more information about the feedback agent please refer to [this blog](#).

Windows Agent

The latest Windows feedback agent can be downloaded from [here](#). To install the agent, run loadbalanceragent.msi on each Real Server:

Leave the default location or change according to your requirements, click **Next**.
Leave the default location or change according to your requirements, click **Next**.

Click **Install** to start the installation process.
Click Finish.

**Note**  
The agent should be installed on all Real Servers in the cluster.

**Starting the Agent**

Once the installation has completed, you’ll need to start the service on the Real Servers. The service is controlled by the Feedback Agent monitor & control program that is also installed along with the Agent. This can be accessed on the Windows server from: Start> Loadbalancer.org > Loadbalancer.org Feedback Agent. It’s also possible to start the service using the services snap-in - the service is called LBCPUMon.

- To start the service, click the **Start** button
- To stop the service, click the **Stop** button

**Linux/Unix Agent**

The Linux feedback agent files can be downloaded using the following links:
Installation & Testing

Install xinetd:

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

Insert this line into /etc/services:

```bash
lb-feedback 3333/tcp # Loadbalancer.org feedback daemon
```

Then run the following commands:

```bash
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
```

to test:

```bash
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.
```

Note The agent files must be installed on all Real Servers, not the load balancer.

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.

Configuring HA - Adding a Secondary Appliance

Our recommended configuration is to use a clustered HA pair of load balancers to provide a highly available and resilient load balancing solution.

We recommend that the Primary appliance should be configured first, then the Secondary should be added. Once the Primary and Secondary are paired, all load balanced services configured on the Primary are automatically replicated to the Secondary over the network using SSH/SCP.
For Enterprise Azure, the HA pair should be configured first. In Azure, when creating a VIP using an HA pair, 2 private IPs must be specified – one for the VIP when it’s active on the Primary and one for the VIP when it’s active on the Secondary. Configuring the HA pair first, enables both IPs to be specified when the VIP is created.

The clustered HA pair uses Heartbeat to determine the state of the other appliance. Should the active device (normally the Primary) suffer a failure, the passive device (normally the Secondary) will take over.

A number of settings are not replicated as part of the Primary/Secondary pairing process and therefore must be manually configured on the Secondary appliance. These are listed by WebUI menu option in the table below:

<table>
<thead>
<tr>
<th>WebUI Main Menu Option</th>
<th>Sub Menu Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Configuration</td>
<td>Hostname &amp; DNS</td>
<td>Hostname and DNS settings</td>
</tr>
<tr>
<td>Local Configuration</td>
<td>Network Interface</td>
<td>All network settings including IP address(es), bonding configuration and VLANs</td>
</tr>
<tr>
<td>Local Configuration</td>
<td>Routing</td>
<td>Routing configuration including default gateways and static routes</td>
</tr>
<tr>
<td>Local Configuration</td>
<td>System Date &amp; time</td>
<td>All time and date related settings</td>
</tr>
<tr>
<td>Local Configuration</td>
<td>Physical – Advanced</td>
<td>Various settings including Internet Proxy, Management Gateway,</td>
</tr>
<tr>
<td></td>
<td>Configuration</td>
<td>Firewall connection tracking table size, NIC offloading, SMTP relay,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>logging and Syslog Server</td>
</tr>
<tr>
<td>Local Configuration</td>
<td>Security</td>
<td>Appliance security settings</td>
</tr>
<tr>
<td>Local Configuration</td>
<td>SNMP Configuration</td>
<td>Appliance SNMP settings</td>
</tr>
<tr>
<td>Local Configuration</td>
<td>Graphing</td>
<td>Appliance graphing settings</td>
</tr>
<tr>
<td>Local Configuration</td>
<td>License Key</td>
<td>Appliance licensing</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Software Updates</td>
<td>Appliance software update management</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Firewall Script</td>
<td>Appliance firewall (iptables) configuration</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Firewall Lockdown Wizard</td>
<td>Appliance management lockdown settings</td>
</tr>
</tbody>
</table>

To add a Secondary node - i.e. create a highly available clustered pair:

1. Deploy a second appliance that will be the Secondary and configure initial network settings.
2. Using the WebUI, navigate to: Cluster Configuration > High-Availability Configuration.
3. Specify the IP address and the loadbalancer user's password for the Secondary (peer) appliance as shown above.

4. Click Add new node.

5. The pairing process now commences as shown below:

6. Once complete, the following will be displayed:

7. To finalize the configuration, restart heartbeat and any other services as prompted in the blue message box at the top of the screen.
Clicking the **Restart Heartbeat** button on the Primary appliance will also automatically restart heartbeat on the Secondary appliance.

**Note**

For more details on configuring HA with 2 appliances, please refer to Appliance Clustering for HA.

**Solving the ARP Problem**

**Solving the ARP Problem for Linux**

**Method 1 (using iptables)**

You can use iptables (netfilter) on each Real Server to re-direct incoming packets destined for the Virtual Service IP address. To make this permanent, simply add the following command to an appropriate start-up script such as `/etc/rc.local` on each of your Real Servers. If Real Servers are serving multiple VIPs, add additional iptables rules for each VIP.

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

**Note**

Change the IP address to be the same as your Virtual Service.

This means redirect any incoming packets destined for 10.0.0.21 (the Virtual Service) locally, i.e. to the primary address of the incoming interface on the Real Server.

**Note**

Method 1 may not always be appropriate if you’re using IP-based virtual hosting on your web server. This is because the iptables rule above redirects incoming packets to the primary address of the incoming interface on the web server rather than any of the virtual hosts that are configured. Where this is an issue, use method 2 below instead.

**Note**

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

**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 Services 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 1-4 below on each Real Server.

**Step 1 of 4: re-configure ARP on the Real Servers (this step can be skipped for IPv6 Virtual Services)**

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

```
Note: Adjust the commands shown above to suit the network configuration of your servers.

**Step 2 of 4: re-configure DAD on the Real Servers (this step can be skipped for IPv4 Virtual Services)**

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

```bash
net.ipv6.conf.lo.dad_transmits=0
net.ipv6.conf.lo.accept_dad=0
```

**Step 3 of 4: apply these settings**

Either reboot the Real Server or run the following command to apply these settings:

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

**Step 4 of 4: add the Virtual Services 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
```

**Note** You can check if this command added the VIP successfully using the command:

```
ip addr ls
```

You can remove the VIP from the loopback adapter using the command:

```
ip addr del dev lo <IPv4-VIP>/32
```
Steps 1, 2 & 3 can be replaced by writing directly to the required files using the following commands (run as root at the command line), this is temporary until the next reboot:

```bash
Note
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>echo 1 &gt; /proc/sys/net/ipv4/conf/all/arp_ignore</td>
<td>1</td>
</tr>
<tr>
<td>echo 1 &gt; /proc/sys/net/ipv4/conf/eth0/arp_ignore</td>
<td>2</td>
</tr>
<tr>
<td>echo 1 &gt; /proc/sys/net/ipv4/conf/eth1/arp_ignore</td>
<td>2</td>
</tr>
<tr>
<td>echo 2 &gt; /proc/sys/net/ipv4/conf/all/arp_announce</td>
<td>2</td>
</tr>
<tr>
<td>echo 2 &gt; /proc/sys/net/ipv4/conf/eth0/arp_announce</td>
<td>2</td>
</tr>
<tr>
<td>echo 2 &gt; /proc/sys/net/ipv4/conf/eth1/arp_announce</td>
<td>2</td>
</tr>
<tr>
<td>echo 0 &gt; /proc/sys/net/ipv6/conf/lo/dad_transmits</td>
<td>0</td>
</tr>
<tr>
<td>echo 0 &gt; /proc/sys/net/ipv6/conf/lo/accept_dad</td>
<td>0</td>
</tr>
</tbody>
</table>

**Method 3 (using firewalld)**

In some newer versions of Linux, iptables is being deprecated in favour of firewalld. The following command can be used on each Real Server to resolve the ARP issue using firewalld:

```bash
firewall-cmd --permanent --direct --add-rule ipv4 nat PREROUTING 0 -d <VIP> -j REDIRECT
```

**Note**
Change the IP address to be the same as your Virtual Service.

To apply the new configuration, reload the firewall rules:

```bash
firewall-cmd --reload
```

The current permanent configuration will become the new firewalld runtime configuration as well as the configuration at the next system start.

**Windows Server 2012, 2016 & 2019**

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. As with Windows 2000/2003/2008, if the Real Server is included in multiple VIPs, you can add additional IP addresses to the Loopback Adapter that correspond to each VIP.

**Step 1 of 3: Install the Microsoft Loopback Adapter**

1. Click **Start**, then run `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 KM-Test Loopback Adapter**, click **Next**.
6. Click **Next** to start the installation, when complete click **Finish**.

**Step 2 of 3: Configure the Loopback Adapter**

1. Open Control Panel and click **Network and Sharing Center**.
2. Click **Change adapter settings**.
3. Right-click the new Loopback Adapter and select **Properties**.
4. Uncheck all items except **Internet Protocol Version 4 (TCP/IPv4)** and **Internet Protocol Version 6 (TCP/IPv6)** as shown below:
Note | Leaving both checked ensures that both IPv4 and IPv6 are supported. Select one if preferred.

5. If configuring IPv4 addresses select Internet Protocol Version (TCP/IPv4), click Properties and configure the IP address to be the same as the Virtual Service (VIP) with a subnet mask of 255.255.255.255, e.g. 192.168.2.20/255.255.255.255 as shown below:

6. If configuring IPv6 addresses select Internet Protocol Version (TCP/IPv6), click Properties and configure the IP address to be the same as the Virtual Service (VIP) and set the Subnet Prefix Length to be the same as your
network setting, e.g. 2001:470:1f09:e72::15/64 as shown below:

7. Click **OK** on TCP/IP Properties, then click **Close** on Ethernet Properties to save and apply the new settings.

**Note**
For Windows 2012/2016/2019, it's not necessary to modify the interface metric on the advanced tab and should be left set to Automatic.

**Step 3 of 3: Configure the strong/weak host behavior**

To configure the correct strong/weak host behavior for Windows 2012/2016/2019, the following commands must be run on each Real Server:

For IPv4 addresses:

```bash
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" as shown below. If you prefer to leave your current NIC names, then the commands above must be modified accordingly. For example, if your network adapters are named "LAN" and "LOOPBACK", the commands required would be:

```bash
netsh interface ipv4 set interface "LAN" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostsend=enabled
```

For IPv6 addresses:
For these commands to work, the LAN connection NIC must be named "net" and the loopback NIC must be named "loopback" as shown below. If you prefer to leave your current NIC names, then the commands above must be modified accordingly. For example, if your network adapters are named "LAN" and "LOOPBACK", the commands required would be:

```
netsh interface ipv6 set interface "LAN" weakhostreceive=enabled
netsh interface ipv6 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv6 set interface "LOOPBACK" weakhostsend=enabled
netsh interface ipv6 set interface "LOOPBACK" dadtransmits=0
```

Note: 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.

- Start PowerShell or use a command window to run the appropriate netsh commands as shown in the example below:

```
Set-NetIpInterface -InterfaceAlias loopback -WeakHostReceive enabled -WeakHostSend enabled -DadTransmits 0
```

Note: This shows an IPv6 example, use the IPv4 commands if you’re using IPv4 addresses.

Repeat steps 1 - 3 on all remaining Windows 2012/2016/2019 Real Server(s).

For Windows 2012/2016/2019 you can also use the following PowerShell Cmdlets:

The following example configures both IPv4 and IPv6 at the same time:

```
Set-NetIpInterface -InterfaceAlias loopback -WeakHostReceive enabled -WeakHostSend enabled -DadTransmits 0
```
Set-NetIpInterface -InterfaceAlias net -WeakHostReceive enabled

To configure just IPv4:

Set-NetIpInterface -InterfaceAlias loopback -WeakHostReceive enabled -WeakHostSend enabled -DadTransmits 0 -AddressFamily IPv4

Set-NetIpInterface -InterfaceAlias net -WeakHostReceive enabled -AddressFamily IPv4

To configure just IPv6:

Set-NetIpInterface -InterfaceAlias loopback -WeakHostReceive enabled -WeakHostSend enabled -DadTransmits 0 -AddressFamily IPv6

Set-NetIpInterface -InterfaceAlias net -WeakHostReceive enabled -AddressFamily IPv6
## 18. Document Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change</th>
<th>Reason for Change</th>
<th>Changed By</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>29 March 2018</td>
<td>Initial version</td>
<td></td>
<td>AH</td>
</tr>
<tr>
<td>1.0.1</td>
<td>6 December 2018</td>
<td>Added the new &quot;Company Contact Information&quot; page</td>
<td>Required updates</td>
<td>AH</td>
</tr>
<tr>
<td>1.1.0</td>
<td>19 September 2019</td>
<td>Styling and layout</td>
<td>General styling updates</td>
<td>RJC</td>
</tr>
<tr>
<td>1.1.1</td>
<td>28 August 2020</td>
<td>New title page</td>
<td>Branding update</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Canadian contact details</td>
<td>Change to Canadian contact details</td>
<td></td>
</tr>
<tr>
<td>1.2.0</td>
<td>1 January 2022</td>
<td>Converted the document to AsciiDoc</td>
<td>Move to new documentation system</td>
<td>AH, RJC, ZAC</td>
</tr>
</tbody>
</table>
About Loadbalancer.org

Loadbalancer.org's mission is to ensure that its clients' businesses are never interrupted. The load balancer experts ask the right questions to get to the heart of what matters, bringing a depth of understanding to each deployment. Experience enables Loadbalancer.org engineers to design less complex, unbreakable solutions - and to provide exceptional personalized support.