Load Balancing
RabbitMQ

Deployment Guide
v1.1.0
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 relevant Administration Manual:

- v7 Administration Manual
- v8 Administration Manual

2. Loadbalancer.org Appliances Supported
All our products can be used for load balancing RabbitMQ. The complete list of models is shown below:

<table>
<thead>
<tr>
<th>Discontinued Models</th>
<th>Current Models *</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise R16</td>
<td>Enterprise R20</td>
</tr>
<tr>
<td>Enterprise VA R16</td>
<td>Enterprise MAX</td>
</tr>
<tr>
<td>Enterprise VA</td>
<td>Enterprise 10G</td>
</tr>
<tr>
<td>Enterprise R320</td>
<td>Enterprise 40G</td>
</tr>
<tr>
<td></td>
<td>Enterprise Ultra</td>
</tr>
<tr>
<td></td>
<td>Enterprise VA R20</td>
</tr>
<tr>
<td></td>
<td>Enterprise VA MAX</td>
</tr>
<tr>
<td></td>
<td>Enterprise AWS **</td>
</tr>
<tr>
<td></td>
<td>Enterprise AZURE **</td>
</tr>
</tbody>
</table>

* For full specifications of these models please refer to: http://www.loadbalancer.org/products/hardware
** Some features may not be supported, please check with Loadbalancer.org support

3. Loadbalancer.org Software Versions Supported

- V7.6.4 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 section 1 of the appendix on page 15 for full details on installing and configuring the server feedback agent.

7. Deployment Concept
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.

- 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 Servers own IP address and the VIP.
- The Real Server 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. Please refer to page 22 for more information.
- On average, DR mode is 8 times quicker than NAT for HTTP, 50 times quicker for Terminal Services and much, much faster for other services.
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 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

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 request to the chosen Real Server. As a result, Layer 7 is a slower technique than DR or NAT mode at Layer 4. 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.

This mode can be deployed in a one-arm or two-arm configuration and does not require any changes to the Real Servers. However, since the load balancer is acting as a full proxy it doesn’t have the same raw throughput as the layer 4 methods.

The load balancer proxies the application traffic to the servers so that the source of all traffic becomes the load balancer.

• SNAT mode is a full proxy and therefore load balanced Real Servers do not need to be changed in any way

• Because SNAT mode is a full proxy any server in the cluster can be on any accessible subnet including across the
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 balancers 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 clients IP address can be passed through either by enabling TProxy on the load balancer, or for HTTP, using X-forwarded-For headers. Please refer to chapter 6 in the administration manual for more details.

SNAT mode can be deployed using either a 1-arm or 2-arm configuration.

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 section 3 of the appendix on page 22 for more information.

For a detailed explanation of DR mode and the nature of the ARP problem, please refer to the section that covers layer 4 DR mode on page 6.

10. Loadbalancer.org Appliance – the Basics

Virtual Appliance Download & Deployment
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 and XEN and has been optimized for each Hypervisor. By default, the VA is allocated 1 CPU, 2GB of RAM and has an 8GB 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 Administration Manual and the ReadMe.txt text file included in the VA download for more detailed information on deploying the VA using various Hypervisors.

Initial Network Configuration
The IP address, subnet mask, default gateway and DNS settings can be configured in several ways as detailed below:

Method 1 - Using the Network Setup Wizard at the console
After boot up, follow the instructions on the console to configure the IP address, subnet mask, default gateway and DNS settings.

Method 2 - Using the WebUI
Using a browser, connect to the WebUI on the default IP address/port: https://192.168.2.21:9443
To set the IP address & subnet mask, use: Local Configuration > Network Interface Configuration
To set the default gateway, use: Local Configuration > Routing
To configure DNS settings, use: Local Configuration > Hostname & DNS

Accessing the Web User Interface (WebUI)

1. Browse to the following URL: https://192.168.2.21:9443/lbadmin/ (replace with your IP address if it’s been changed)
   * Note the port number → 9443

2. Login to the WebUI:
   
   Username: loadbalancer
   Password: loadbalancer

Note: To change the password, use the WebUI menu option: Maintenance > Passwords.
Once logged in, the WebUI will be displayed as shown below:

**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 slave unit is covered in section 2 of the appendix on page 19.

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](image)

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 HAProxy

**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

1 - 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 $(92/100 \times \text{requested weight})$ to find the new optimized weight.

Note: The 'Requested Weight' is the weight set in the WebUI for each Real Server. For more information please also refer to the following blog article: [http://www.loadbalancer.org/blog/open-source-windows-service-for-reporting-server-load-back-to-haproxy-load-balancer-feedback-agent](http://www.loadbalancer.org/blog/open-source-windows-service-for-reporting-server-load-back-to-haproxy-load-balancer-feedback-agent)

Windows Agent

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

![Setup - LoadBalancer.org Feedback Agent Monitor](image)

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
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'.

Note: The agent should be installed on all Real Serves in the cluster.
• 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:

- **readme file**: [http://downloads.loadbalancer.org/agent/linuxv4.1/readme.txt](http://downloads.loadbalancer.org/agent/linuxv4.1/readme.txt)
- **xinetd file**: [http://downloads.loadbalancer.org/agent/linuxv4.1/lb-feedback](http://downloads.loadbalancer.org/agent/linuxv4.1/lb-feedback)
- **feedback script**: [http://downloads.loadbalancer.org/agent/linuxv4.1/lb-feedback.sh](http://downloads.loadbalancer.org/agent/linuxv4.1/lb-feedback.sh)

**Installation & Testing**

```bash
# Install xinetd
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.
```

Note: The agent should be installed on all Real Serves in the cluster.
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.

Configuration
As mentioned, both layer 4 and layer 7 VIPs can be configured to use the feedback agent. To Configure Virtual Services to use Agent/HTTP Feedback follow the steps below:

1. Using the WUI, navigate to:
   - Cluster Configuration > Layer 4 - Virtual Services
   or
   - Cluster Configuration > Layer 7 - Virtual Services
2. Click Modify next to the relevant Virtual Service

   ![Feedback Method](image)

   Feedback Method
   Feedback Method: Agent
   Feedback Agent Port: 3333

3. Change the Feedback Method to either Agent or HTTP for layer 4 VIPs
4. Change the Feedback Method to Agent for layer 7 VIPs
5. Click Update
6. Reload/restart services as prompted

2 – Clustered Pair Configuration – Adding a Slave Unit
If you initially configured just the master unit and now need to add a slave - our recommended procedure, please refer to the relevant section below for more details:

Note: A number of settings are not replicated as part of the master/slave pairing process and therefore must be manually configured on the slave appliance. These are listed below:

- Hostname & DNS settings
- Network settings including IP addresses, bonding configuration and VLANs
- Routing configuration including default gateways and static routes
- Date & time settings
- Physical – Advanced Configuration settings including Internet Proxy IP address & port, Firewall table
size, SMTP relay and Syslog server
- SNMP settings
- Graphing settings
- Firewall Script & Firewall Lockdown Script settings
- Software updates

Version 7:
Please refer to Chapter 8 – Appliance Clustering for HA in the v7 Administration Manual.

Version 8:
To add a slave node – i.e. create a highly available clustered pair:

- Deploy a second appliance that will be the slave and configure initial network settings
- Using the WebUI, navigate to: Cluster Configuration > High-Availability Configuration

Specify the IP address and the loadbalancer users password (the default is 'loadbalancer') for the slave (peer) appliance as shown above

Click Add new node

The pairing process now commences as shown below:
Once complete, the following will be displayed:

To finalize the configuration, restart heartbeat and any other services as prompted in the blue message box at the top of the screen.

Note: Clicking the Restart Heartbeat button on the master appliance will also automatically restart heartbeat on the slave appliance.

Note: Please refer to chapter 9 – Appliance Clustering for HA in the Administration Manual for more detailed information on configuring HA with 2 appliances.
3 – 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 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 -d <VIP> -j REDIRECT
```

**e.g.**
```
iptables -t nat -A PREROUTING -d 10.0.0.21 -j REDIRECT
```

*(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.

Also, 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.

**Step 1: 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:
Step 2: 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:

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

Step 3: apply these settings
Either reboot the Real Server or run the following command to apply these settings:

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

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

```
ip addr del dev lo <Ipv4-VIP>/32
```

Note: 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:

```
echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
echo 1 > /proc/sys/net/ipv4/conf/eth0/arp_ignore
echo 1 > /proc/sys/net/ipv4/conf/eth1/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/eth0/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/eth1/arp_announce
echo 0 > /proc/sys/net/ipv6/conf/lo/dad_transmits
echo 0 > /proc/sys/net/ipv6/conf/lo/accept_dad
```

---

**Solving The ARP Problem For Windows Servers**

When using Layer 4 DR mode, the ARP problem must be solved. This involves configuring each Real Server to be able to receive traffic destined for the VIP, and ensuring that each Real Server does not respond to ARP requests for the VIP address – only the load balancer should do this.

The steps below are for Windows 2012/2016, for other versions of Windows please refer to chapter 6 in the [administration manual](#).

**Step 1: 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: 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. Un-check 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.

**Important:** When configuring the Loopback Adapter to solve the ARP Problem, the following options must also be checked (ticked):

- Client for Microsoft Networks
- File & Printer Sharing for Microsoft Networks

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.

   192.168.220.255/255.255.255.255 as shown below:

![](image)

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:

![](image)
7. Click **OK** on TCP/IP Properties, then click **Close** on Ethernet Properties to save and apply the new settings.
8. Now repeat the above process on the other Windows 2012/2016 Real Servers.

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

Windows Server 2000 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 2008 and later supports strong host sends and receives for both IPv4 and IPv6 by default. To ensure that Windows 2012/2016 is running in the correct mode to be able to respond to the VIP, the following commands must be run on each Real Server:

For IPv4 addresses:

```plaintext
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 IPv6 addresses:

```plaintext
netsh interface ipv6 set interface "net" 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
```

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:

```plaintext
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:

```plaintext
netsh interface ipv6 set interface "net" 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
```

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:

```plaintext
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.

1. Start Powershell or use a command window to run the appropriate netsh commands as shown in the example below:

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

2. Now repeat these 4 commands on the other Windows 2012/2016 Real Servers

Note: Solving the ARP problem for other version of Windows is similar. For full details, please refer to the administration manual.
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>
</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.

United Kingdom
Loadbalancer.org Ltd.
Compass House, North Harbour
Business Park, Portsmouth, PO6 4PS
UK: +44 (0) 330 380 1064
sales@loadbalancer.org
support@loadbalancer.org

Canada
Loadbalancer.org Appliances Ltd.
300-422 Richards Street, Vancouver,
BCV6B 2Z4, Canada
TEL: +1 302.213.0122
sales@loadbalancer.org
support@loadbalancer.org

United States
Loadbalancer.org, Inc.
4550 Linden Hill Road, Suite 201
Wilmington, DE 19808, USA
TEL: +1 833.274.2566
sales@loadbalancer.org
support@loadbalancer.org

Germany
Loadbalancer.org GmbH
Tengstraße 2780798,
München, Germany
TEL: +49 (0)89 2000 2179
sales@loadbalancer.org
support@loadbalancer.org