

Load Balancing SFTPGo

Version 1.0.0



Table of Contents

1. About this Guide	4
2. Loadbalancer.org Appliances Supported	4
3. Software Versions Supported	4
3.1. Loadbalancer.org Appliance	4
3.2. SFTPGo	4
4. SFTPGo	4
5. Load Balancing SFTPGo	5
5.1. Load Balancing & HA Requirements	5
5.1.1. Prerequisites for High Availability	5
5.2. Virtual Service (VIP) Requirements	6
5.3. Persistence (aka Server Affinity)	7
5.4. Health Checks	7
5.5. SSL/TLS Termination	8
6. Deployment Concept	8
7. Load Balancer Deployment Methods	9
7.1. Layer 4 SNAT Mode	9
7.2. Layer 7 SNAT Mode	10
7.3. Our Recommendation	10
8. Loadbalancer.org Appliance – the Basics	11
8.1. Virtual Appliance	11
8.2. Initial Network Configuration	11
8.3. Accessing the Appliance WebUI	11
8.3.1. Main Menu Options	13
8.4. Appliance Software Update	14
8.4.1. Online Update	14
8.4.2. Offline Update	14
8.5. Ports Used by the Appliance	15
8.6. HA Clustered Pair Configuration	16
9. Configuring SFTPGo for Load Balancing	16
9.1. Step 1 – Install SFTPGo on Each Node	16
9.2. Step 2 – Deploy the Shared PostgreSQL Data Provider	17
9.3. Step 3 – Point SFTPGo at the Shared Database	18
9.4. Step 4 – Create the Administrator Account	18
9.5. Step 5 – Configure the Shared Defender	18
9.6. Step 6 – Set Up Shared File Storage	19
9.7. Step 7 – Synchronize the SSH Host Keys	20
9.8. Step 8 – Enable the FTP Service	21
9.9. Step 9 – Enable the WebDAV Service	21
9.10. Step 10 – Enabling Client IP Visibility (Optional)	22
9.10.1. SFTP/SCP and FTP – the PROXY Protocol	22
9.10.2. WebDAV – the X-Forwarded-For Header	23
10. Appliance Configuration for SFTPGo	23
10.1. Layer 7 – Advanced Configuration	23
10.2. VIP 1 – SFTPGo-SFTP	24
10.2.1. Virtual Service (VIP) Configuration	24
10.2.2. Define the Associated Real Servers (RIPs)	25
10.3. VIP 1 (Alternative) – SFTPGo-SFTP Using the PROXY Protocol	25
10.3.1. Virtual Service (VIP) Configuration	25

10.3.2. Define the Associated Real Servers (RIPs)	27
10.4. VIP 2 – SFTPGO-FTP	27
10.4.1. Virtual Service (VIP) Configuration	28
10.4.2. Define the Associated Real Servers (RIPs)	29
10.5. VIP 3 – SFTPGO-WebDAV	29
10.5.1. Virtual Service (VIP) Configuration	29
10.5.2. Define the Associated Real Servers (RIPs)	31
10.6. Finalizing the Configuration	31
11. Testing & Verification	32
11.1. Verifying the SFTPGO Cluster	32
11.2. Using System Overview	32
11.3. Accessing SFTPGO via the Load Balancer	33
11.4. Verifying Client IP Visibility	34
11.5. Testing Failover	35
12. Technical Support	35
13. Further Documentation	35
14. Appendix	37
14.1. SFTPGO Environment File Reference	37
14.2. Useful SFTPGO Service Commands	38
14.3. Additional Considerations	38
14.4. Configuring HA - Adding a Secondary Appliance	39
14.4.1. Non-Replicated Settings	39
14.4.2. Configuring the HA Clustered Pair	40
15. Document Revision History	42

1. About this Guide

This guide details the steps required to configure a load balanced SFTPGo environment utilizing Loadbalancer.org appliances. It covers the configuration of the load balancers and also the SFTPGo 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 with SFTPGo. For full specifications of available models please refer to <https://www.loadbalancer.org/products/enterprise>.

Some features may not be available or fully supported in all cloud platforms due to platform specific limitations. For more details, please refer to the "Main Differences to our Standard (Non-Cloud) Product" section in the appropriate cloud platform [Quick Start Guide](#) or check with Loadbalancer.org support.

3. Software Versions Supported

3.1. Loadbalancer.org Appliance

- V8.13.6 and later

Note

The screenshots used throughout this document aim to track the latest Loadbalancer.org software version. If you're using an older version, or the very latest, the screenshots presented here may not match your WebUI exactly.

3.2. SFTPGo

- The latest stable release. Please refer to <https://github.com/drakkan/sftpgo/releases> to determine the current release at the time of deployment.

4. SFTPGo

SFTPGo is an event driven, multi-protocol file transfer server. It presents the same set of users and the same set of files over SFTP, SCP, FTP/S, WebDAV and HTTP/S, and it supports a number of storage backends including the local filesystem, S3, Azure Blob Storage and Google Cloud Storage.

SFTPGo is developed by the SFTPGo Authors (Nicola Murino) and is published as open source software under the AGPL-3.0 license. Commercial enterprise editions are available via <https://sftpgo.com>.

The product presents the following user facing components:

- **SFTP & SCP** – file transfer over SSH. SCP is served by the same SSH listener as SFTP.
- **FTP/FTPS** – classic file transfer, with a configurable passive data port range.



- **WebDAV** – HTTP based file access, used for browser access and for mounting the server as a network drive.
- **WebAdmin** – the browser based administration interface.
- **WebClient** – the browser based end user interface.
- **REST API** – the programmatic management and file transfer interface.

5. Load Balancing SFTPGo

Note

It's highly recommended that you have a working SFTPGo environment first before implementing the load balancer.

5.1. Load Balancing & HA Requirements

SFTPGo can be installed on multiple servers and load balanced to provide load sharing, high availability and resilience. SFTPGo does not include a clustering layer of its own – instead, each node is configured to use the same shared state. Once that shared state is in place, any node can serve any request and the load balancer is free to distribute connections across all of them.

The following SFTPGo services are load balanced in this guide:

Service	Port(s)	Notes
SFTP / SCP	TCP 2022	The primary file transfer protocols. SCP is served by the same SSH listener as SFTP, so a single VIP serves both.
FTP / FTPS	TCP 2121 + passive range 50000-50100	Passive mode only.
WebDAV	TCP 10085	HTTP based file access, used for browser access and for mounting the server from Windows Explorer.

Note

Active mode FTP is deliberately **not** supported by this deployment. Active mode requires the load balancer to be made transparent at layer 7 using TProxy, which in turn imposes a two subnet topology and requires the SFTPGo servers to use the load balancer as their default gateway. It also requires a wide inbound ephemeral port range to be opened. If active mode FTP is a hard requirement, please refer to the "FTP" section of the Administration Manual and contact Loadbalancer.org support.

Note

WebAdmin (TCP 8080) is **not** load balanced in this deployment. Administrators access it directly per node, e.g. <http://10.11.40.65:8080/web/admin>. Because all nodes share one data provider, an administrative change made on any node is immediately visible on all of them. If highly available administrative access is required, an additional layer 7 HTTP VIP can be created for port 8080 following the same pattern as the WebDAV VIP.

5.1.1. Prerequisites for High Availability



All of the following must be true before the SFTPGo cluster will behave correctly behind a load balancer. Each item is covered in [Configuring SFTPGo for Load Balancing](#).

- All SFTPGo nodes must use the same shared data provider (a PostgreSQL database in this guide). Users, folders, admins and API keys are all held there.
- All SFTPGo nodes must present the **same SSH host keys**. The host keys are generated per node on first start, so they must be copied from the first node to all of the others.
- All SFTPGo nodes must have access to the **same shared file storage**, so that a file uploaded through one node is immediately readable through any other.
- The SFTPGo **Defender** (brute force protection) must use the [provider](#) driver, so that ban events are written to the shared database and are therefore honoured by every node.

5.2. Virtual Service (VIP) Requirements

To provide load balancing and HA for SFTPGo, the following VIPs are required:

Ref.	VIP Name	Mode	Port(s)	Persistence Mode	Health Check
VIP 1	SFTPGo-SFTP	Layer 4 SNAT	2022	Source IP, 1200s	Connect to port 2022
VIP 1 (alt.)	SFTPGo-SFTP	Layer 7 TCP	2022	Source IP, 20 mins	Connect to port 2022
VIP 2	SFTPGo-FTP	Layer 7 TCP	2121, 50000-50100	Source IP, 20 mins	Connect to port 2121
VIP 3	SFTPGo-WebDAV	Layer 7 HTTP	10085	HTTP Cookie	Negotiate HTTP (GET)

Note

The persistence timeouts above all express the same 20 minute window. They are written differently because the fields differ: the layer 4 **Timeout** is in seconds (1200) and the layer 7 **Persistence timeout** is in minutes (20).

Important

The two variants of VIP 1 are **mutually exclusive alternatives** – configure one or the other, never both.

Use the **layer 4 SNAT** variant for maximum throughput. The SFTPGo nodes will log the load balancer's IP address as the client address.

Use the **layer 7 TCP** variant if the real client IP addresses must be visible to SFTPGo. This variant sends the PROXY protocol, which requires a matching configuration change on the SFTPGo nodes (see [Step 10 – Enabling Client IP Visibility \(Optional\)](#)).

They cannot be run side by side for two reasons. Firstly, a layer 4 SNAT mode VIP and a layer 7 SNAT mode VIP must not share the same RIP:PORT combination because the required firewall rules conflict. Secondly, once [proxy_protocol](#) is enabled on the SFTPGo nodes with the load balancer listed in [proxy_allowed](#), SFTPGo expects a PROXY header on connections arriving from the load balancer – so connections arriving from the layer 4 VIP, which does not send one, would fail.



Note

In this guide a single VIP address, **10.11.40.60**, is used for all three Virtual Services. They are distinguished by port. Separate VIP addresses can be used instead if preferred.

5.3. Persistence (aka Server Affinity)

Source IP persistence is applied to the SFTP and FTP Virtual Services, with a 20 minute (1200 second) timeout. This serves two purposes:

- **FTP passive mode requires it.** The client establishes a control connection, and the server then hands the client a port on which to open a separate data connection. Both connections must reach the same SFTPGO node.
- **It keeps long file transfers intact.** Source IP persistence prevents a client that reconnects mid session from being sent to a different node.

The persistence timeout is reset on every new connection from that client, so an actively used client will remain pinned for as long as it stays active.

Important

If large transfers or long lived sessions are expected, increase the persistence timeout accordingly – 1200 seconds is a starting point, not an upper limit.



Tip

For the SFTP and FTP Virtual Services, consider enabling **Clear Stick on Drain** (modify the VIP, *Other > [Advanced]*). SSH sessions are long lived with large connection timeouts, so without this setting a Real Server that has been drained via System Overview will continue to receive reconnecting clients until their stick table entries expire.

5.4. Health Checks

- **SFTP and FTP** – a simple TCP connect check is used. SFTPGO has no SFTP or FTP level probe endpoint, and a successful TCP handshake on the listener is a reliable indication that the service is accepting connections.



Note

The TCP probe opens and immediately closes a connection, which causes SFTPGO to write a debug level **connection_failed / no_auth_tried** entry to its log every check interval on every node. This is harmless. Setting the SFTPGO log level to **info** in production suppresses it.

- **WebDAV** – an HTTP GET of the WebDAV root is used. An unauthenticated GET returns **HTTP 401**, which proves that the WebDAV listener is up and processing requests.

Important

A 401 is not a 2xx or 3xx response, so the **Response Expected** field **must** be populated. If it is left blank the appliance treats only 2xx and 3xx as healthy and every node will be marked down. See [VIP 3 – SFTPGO-WebDAV](#) for the value to use.



Note

SFTPGO does provide a dedicated **/healthz** endpoint, but it is served by the telemetry server rather than by any of the load balanced services. That server is disabled by default



(`telemetry.bind_port` is 0) and binds to `127.0.0.1` when enabled, so it is not reachable from the load balancer without additional configuration. Each service is therefore checked on its own listener instead.

5.5. SSL/TLS Termination

No SSL/TLS termination is performed on the load balancer in this deployment.

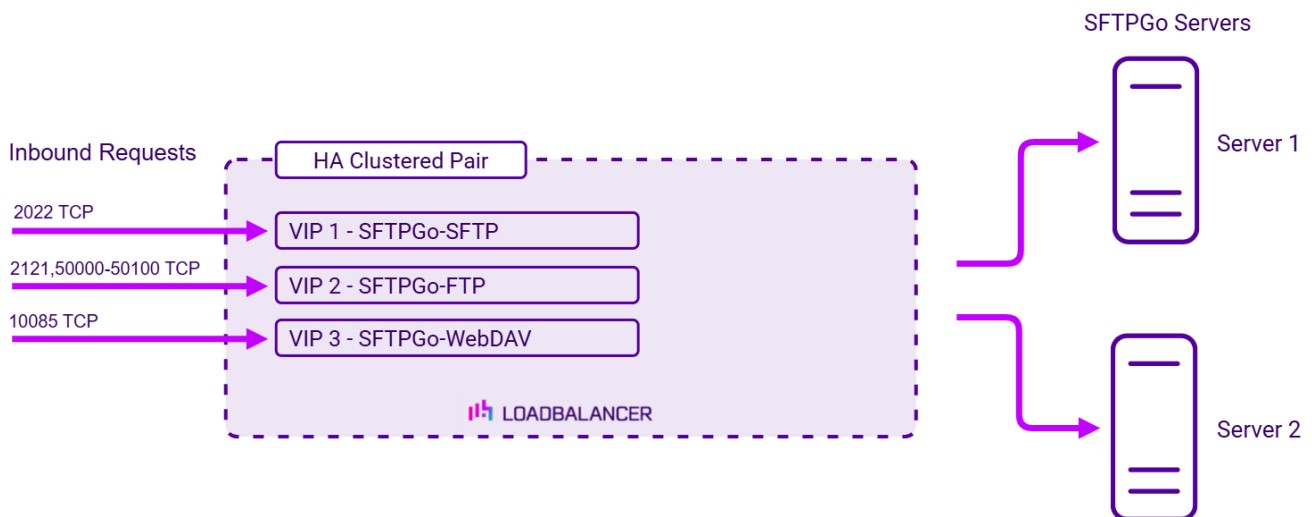
- SFTP and SCP are inherently encrypted by the SSH transport.
- FTPS and HTTPS for WebDAV can be terminated on the SFTPGo nodes themselves using SFTPGo's own TLS settings (`SFTPGO_FTPDBINDINGS0TLS_MODE` and `SFTPGO_WEBDAVDBINDINGS0ENABLE_HTTPS`), in which case the layer 7 Virtual Services operate in TCP mode and simply pass the encrypted traffic through.

Note

If you would prefer to terminate TLS for WebDAV on the load balancer instead, this is fully supported. Certificates in PEM or PFX format can be uploaded using *Cluster Configuration > SSL Certificate*, and an SSL termination is then associated with the WebDAV VIP. Please refer to *SSL Termination* in the Administration Manual.

6. Deployment Concept

Once the load balancer is deployed, clients connect to the Virtual Services (VIPs) rather than connecting directly to one of the SFTPGo nodes. These connections are then load balanced across the SFTPGo nodes to distribute the load according to the load balancing algorithm selected. All nodes share a single PostgreSQL data provider and a single file store, so each node presents an identical view of users and files.



The diagram above shows the inbound client traffic paths only. Not shown, but equally essential, is what sits behind the nodes: every SFTPGo server also connects to the **shared PostgreSQL data provider** and the **shared file storage**, and it is those two shared resources that make the nodes interchangeable. Configuring them is covered in [Step 2 – Deploy the Shared PostgreSQL Data Provider](#) and [Step 6 – Set Up Shared File Storage](#).

Note

The load balancer can be deployed as a single unit, although Loadbalancer.org recommends a



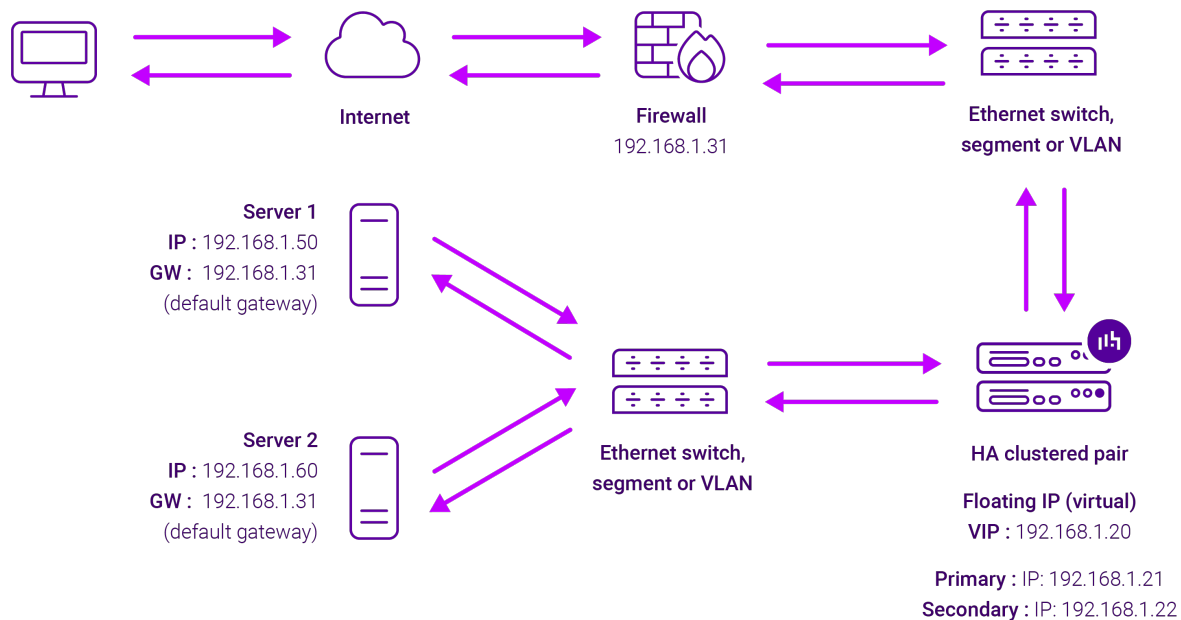
7. Load Balancer Deployment Methods

The load balancer can be deployed in 4 fundamental ways: *Layer 4 DR mode*, *Layer 4 NAT mode*, *Layer 4 SNAT mode*, and *Layer 7 SNAT mode*.

For SFTPGo, layer 4 SNAT mode and layer 7 SNAT mode are used. Both modes are described below.

7.1. Layer 4 SNAT Mode

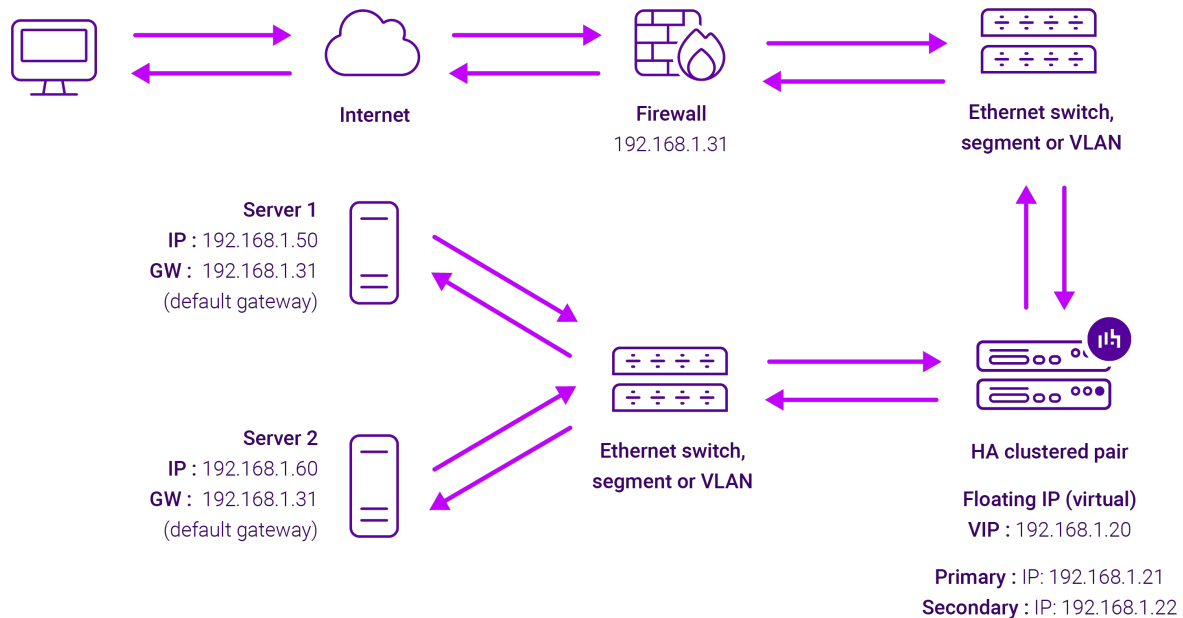
Layer 4 SNAT mode is a high performance solution, although not as fast as Layer 4 NAT mode or Layer 4 DR mode. The image below shows an example network diagram for this mode.



- Real Servers in the cluster can be on any accessible network including across the Internet or WAN.
- Layer 4 SNAT mode is not transparent, an iptables SNAT rule translates the source IP address to be the load balancer rather than the original client IP address.
- Layer 4 SNAT mode can be deployed using either a one-arm or two-arm configuration. For two-arm deployments, **eth1** is typically used for client side connections and **eth0** is used for Real Server connections, although this is not mandatory since any interface can be used for any purpose.
- Requires no mode-specific configuration changes to the load balanced Real Servers.
- Port translation is possible with Layer 4 SNAT mode, e.g. VIP:80 → RIP:8080 is supported.
- You should not use the same RIP:PORT combination for layer 4 SNAT mode VIPs and layer 7 SNAT (Proxy) mode VIPs because the required firewall rules conflict.

7.2. 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. The image below shows an example network diagram for this mode.



- Because layer 7 SNAT mode is a full proxy, Real Servers in the cluster can be on any accessible network 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. For two-arm deployments, **eth1** is typically used for client side connections and **eth0** is used for Real Server connections, although this is not mandatory since any interface can be used for any purpose.
- Requires no mode-specific configuration changes to the load balanced Real Servers.
- 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.

7.3. Our Recommendation



- **Layer 4 SNAT mode** is used for the SFTP Virtual Service, where raw throughput matters most and where no application layer processing is needed.
- **Layer 7 SNAT mode** is used for the FTP and WebDAV Virtual Services. FTP passive mode needs a multi-port service with source IP persistence, and WebDAV benefits from an HTTP level health check and HTTP cookie persistence. Layer 7 is also the mode to use for SFTP if the real client IP addresses must be visible to the SFTPGo nodes, since it can send the PROXY protocol.

Neither mode requires any mode specific configuration changes on the SFTPGo nodes – there is no "ARP problem" to solve, and the SFTPGo nodes can be on any network reachable from the load balancer.

8. Loadbalancer.org Appliance – the Basics

8.1. 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 [Virtual Appliance Installation](#) and the ReadMe.txt text file included in the VA download for additional information on deploying the VA using the various Hypervisors.

Note

The VA has 4 network adapters. For VMware only the first adapter (**eth0**) is connected by default. For HyperV, KVM, XEN and Nutanix AHV all adapters are disconnected by default. Use the network configuration screen within the Hypervisor to connect the required adapters.

8.2. Initial Network Configuration

After boot up, follow the instructions on the appliance console to configure the management IP address, subnet mask, default gateway, DNS servers and other network and administrative settings.

Important

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

8.3. Accessing the Appliance WebUI

The WebUI is accessed using a web browser. By default, users are authenticated using Apache authentication. Users can also be authenticated against LDAP, LDAPS, Active Directory or Radius - for more information, please refer to [External Authentication](#).

Note

There are certain differences when accessing the WebUI for the cloud appliances. For details, please refer to the relevant [Quick Start / Configuration Guide](#).



1. Using a browser, navigate to the following URL:

https://<IP-address-configured-during-the-network-setup-wizard>:9443/lbadmin/

 Note

You'll receive a warning about the WebUI's SSL certificate. This is due to the default self signed certificate that is used. If preferred, you can upload your own certificate - for more information, please refer to [Appliance Security Features](#).

 Note

If you need to change the port, IP address or protocol that the WebUI listens on, please refer to [Service Socket Addresses](#).

2. Log in to the WebUI using the following credentials:

Username: loadbalancer

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

 Note

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

Once logged in, the WebUI will be displayed as shown below:

Primary | Secondary Active | Passive Link 8 Seconds ↻

System Overview
Local Configuration
Cluster Configuration
Maintenance
View Configuration
Reports
Logs
Support
Live Chat

WARNING: YOUR TRIAL IS DUE TO EXPIRE IN 30 DAYS.
Buy with confidence. All purchases come with a 90 day money back guarantee.
Already bought? Enter your license key [here](#)
[Buy Now](#)

System Overview ? 2025-05-08 12:37:21 UTC

Would you like to run the Setup Wizard?
[Accept](#) [Dismiss](#)

VIRTUALL SERVICE ▾ IP ▾ PORTS ▾ CONNS ▾ PROTOCOL ▾ METHOD ▾ MODE ▾
No Virtual Services configured.

Network Bandwidth

RX 28 Min, 2713 Avg, 27344772 Total,
TX 0 Min, 13777 Avg, 138872181 Total,

System Load Average

1m average 0.00 Min, 0.08 Avg, 0.68 Max
5m average 0.00 Min, 0.04 Avg, 0.30 Max
15m average 0.00 Min, 0.02 Avg, 0.12 Max

Memory Usage

3. You'll be asked if you want to run the Setup Wizard. Click **Dismiss** if you're following a guide or want to configure the appliance manually. Click **Accept** to start the Setup Wizard.



Note

The Setup Wizard can only be used to configure Layer 7 services.

8.3.1. 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 creating backups

View Configuration - Display the saved appliance configuration settings

Reports - View various appliance reports & graphs

Logs - View various appliance logs

Support - Create a support download, contact the support team & access useful links

8.4. Appliance Software Update

We recommend that the appliance is kept up to date to ensure that you benefit from the latest bug fixes, security updates and feature improvements. Both online and offline update are supported.

 Note

For full details, please refer to [Appliance Software Update](#) in the Administration Manual.

 Note

Services may need to be restarted/reloaded after the update process completes or in some cases a full appliance restart may be required. We therefore recommend performing the update during a maintenance window.

8.4.1. Online Update

The appliance periodically contacts the Loadbalancer.org update server (**update.loadbalancer.org**) and checks for updates. This is the default behavior and can be disabled if preferred. If an update is found, a notification similar to the example below will be displayed at the top of the WebUI:

Information: Update 8.13.10 is now available for this appliance.

Online Update

Click **Online Update**. A summary of all new features, improvements, bug fixes and security updates included in the update will be displayed. Click **Update** at the bottom of the page to start the update process.

 Important

Do not navigate away whilst the update is ongoing, this may cause the update to fail.

The update can take several minutes depending on download speed and upgrade version. Once complete, the following message will be displayed:

Information: Update completed successfully. Return to **system overview**.

If services need to be reloaded/restarted or the appliance needs a full restart, you'll be prompted accordingly.

8.4.2. Offline Update

If the appliance does not have access to the Internet, offline update can be used.

To check for the latest version, please refer to our product roadmap page available [here](#). To obtain the latest offline update files contact support@loadbalancer.org.

To perform an offline update:



1. Using the WebUI, navigate to: **Maintenance > Software Update**.
2. Select **Offline Update**.
3. The following screen will be displayed:

Software Update

Offline Update

The following steps will lead you through offline update.

1. Contact **Loadbalancer.org support** to obtain the offline update archive and checksum.
2. Save the archive and checksum to your local machine.
3. Select the archive and checksum files in the upload form below.
4. Click **Upload and Install** to begin the update process.

Archive: No file chosen

Checksum: No file chosen

4. Select the **Archive** and **Checksum** files.
5. Click **Upload and Install**.
6. If services need to be reloaded/restarted or the appliance needs a full restart, you'll be prompted accordingly.

8.5. Ports Used by the Appliance

By default, the appliance uses the following TCP & UDP ports:

Protocol	Port	Purpose
TCP	22	SSH
TCP & UDP	53	GSLB
TCP & UDP	123	NTP
TCP & UDP	161	SNMP
UDP	6694	Heartbeat between Primary & Secondary appliances in HA mode
TCP	7778	HAProxy persistence table replication
TCP	9000	Gateway service for ADC Portal comms
TCP	9080	WebUI - HTTP (disabled by default)
TCP	9081	Nginx fallback server
TCP	9443	WebUI - HTTPS
TCP	25565	Shuttle service for ADC Portal comms

Note

All ports listed above except port 123 (NTP) can be changed if required.



- To change the port used for heartbeat, refer to [Configuring High Availability](#)
- To change the port used for HAProxy replication, refer to [Layer 7 - Advanced Configuration](#)
- To change other ports, refer to [Service Socket Addresses](#)

8.6. 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 the section [Configuring HA - Adding a Secondary Appliance](#) of the appendix.

9. Configuring SFTPGo for Load Balancing

The steps in this section must be completed before the load balancer is brought into service. They convert two (or more) independent SFTPGo installations into a cluster of interchangeable nodes.

The example environment used throughout is as follows:

Role	IP Address	Notes
Load balancer (primary)	10.11.40.55	
Load balancer (secondary)	10.11.40.56	Optional, for an HA clustered pair
Virtual Service address (VIP)	10.11.40.60	Shared by all three Virtual Services
SFTPGo node 1	10.11.40.65	
SFTPGo node 2	10.11.40.66	
PostgreSQL & file share	10.11.40.5	

Note

All of the SFTPGo settings below are applied using environment files placed in `C:\ProgramData\SFTPGo\env.d\` on each node. SFTPGo reads every file in that directory at start up, which keeps the change set separate from the main `sftpgo.json` configuration file and makes it trivial to replicate onto additional nodes. A consolidated listing of all of the files is provided in [SFTPGo Environment File Reference](#).

9.1. Step 1 – Install SFTPGo on Each Node

Repeat the following on each node:

1. Download the Windows installer from <https://github.com/drakkan/sftpgo/releases> – use the `.exe` installer for the latest release.
2. Run the installer. It registers SFTPGo as a Windows service and starts it automatically.
3. Open PowerShell as Administrator and allow SFTPGo through the Windows Firewall:

```
netsh advfirewall firewall add rule name="SFTPGo SFTP" dir=in action=allow protocol=TCP
localport=2022
netsh advfirewall firewall add rule name="SFTPGo WebUI" dir=in action=allow protocol=TCP
```



```
localport=8080
```

Alternatively, allow the executable itself:

```
netsh advfirewall firewall add rule name="SFTPGo Service" dir=in action=allow program
="C:\Program Files\SFTPGo\sftpg.exe"
```

❗ Important

Do **not** create an administrator account yet, on either node. A fresh installation uses SFTPGo's built-in SQLite data provider, and administrators are stored **in the data provider**. Any account created now would be left behind in SQLite when the nodes are switched to PostgreSQL in [Step 3 – Point SFTPGo at the Shared Database](#), and SFTPGo would present its initial setup screen again. The administrator is created once the shared database is in place – see [Step 4 – Create the Administrator Account](#).

9.2. Step 2 – Deploy the Shared PostgreSQL Data Provider

SFTPGo supports several data providers. PostgreSQL is used here because it is widely deployed and well documented. A single PostgreSQL instance is sufficient for the cluster – PostgreSQL handles concurrent reads and writes from all SFTPGo nodes natively, so no application level synchronization or database replication is required for the cluster to function.

Install PostgreSQL on a server reachable by all SFTPGo nodes. This can be one of the SFTPGo servers or, as in this example, a separate machine at 10.11.40.5. Installers are available from <https://www.postgresql.org/download/windows/>.

1. Using pgAdmin or the `psql` command line, create the database and the user:

```
CREATE DATABASE sftpgo WITH ENCODING='UTF8' CONNECTION LIMIT=-1;
CREATE USER sftpgo WITH ENCRYPTED PASSWORD '<database-password>';
GRANT ALL PRIVILEGES ON DATABASE sftpgo TO sftpgo;
```

2. Edit `pg_hba.conf` (typically `C:\Program Files\PostgreSQL\<version>\data\pg_hba.conf`) and add a line for each SFTPGo node:

host	sftpgo	sftpgo	10.11.40.65/32	scram-sha-256
host	sftpgo	sftpgo	10.11.40.66/32	scram-sha-256

3. Edit `postgresql.conf` and set `listen_addresses = '*'`, or specify the appropriate interface.
4. Allow PostgreSQL through the Windows Firewall on TCP port 5432.
5. Restart the PostgreSQL service so that the changes take effect.

📌 Note

A single PostgreSQL instance is a single point of failure. This is acceptable for a lab or proof of concept, but for production Loadbalancer.org recommends deploying PostgreSQL in a highly available configuration such as streaming replication with Patroni, or a managed cluster.



9.3. Step 3 – Point SFTPGo at the Shared Database

On **each** SFTPGo node, create the file `C:\ProgramData\SFTPGo\env.d\database.env` with the following contents:

```
SFTPGO_DATA_PROVIDER__DRIVER=postgresql
SFTPGO_DATA_PROVIDER__NAME=sftpgo
SFTPGO_DATA_PROVIDER__HOST=10.11.40.5
SFTPGO_DATA_PROVIDER__PORT=5432
SFTPGO_DATA_PROVIDER__USERNAME=sftpgo
SFTPGO_DATA_PROVIDER__PASSWORD=<database-password>
```

Then restart the service on each node:

```
Restart-Service sftpgo
```

9.4. Step 4 – Create the Administrator Account

With every node now pointed at the same PostgreSQL database, that database is empty and contains no administrators, so SFTPGo presents its initial setup screen. Creating the administrator at this point writes it to the shared database, which makes it valid on every node.

1. On **one** node only, browse to `http://127.0.0.1:8080/web/admin`. SFTPGo will display its initial setup screen.
2. Create the administrator account.
3. Confirm that it worked on a **different** node: browse to that node's WebAdmin, e.g. `http://10.11.40.66:8080/web/admin`, and log in with the same credentials. They must be accepted without any further setup.

Note

If you see the initial setup screen again on the second node, that node is not reading the shared database. Re-check its `database.env` against [Step 3 – Point SFTPGo at the Shared Database](#), confirm the SFTPGo service was restarted, and check `pg_hba.conf` on the PostgreSQL server for a line covering that node's address.

If you created an administrator before switching data providers, it still exists in the original SQLite database but is invisible to PostgreSQL. Simply create the account again here – there is nothing to clean up.

9.5. Step 5 – Configure the Shared Defender

The SFTPGo Defender provides brute force protection. By default it holds its state in memory, which means each node would maintain its own independent list of banned hosts. Switching it to the `provider` driver stores ban events in PostgreSQL so that they are shared by every node.

On **each** SFTPGo node, create the file `C:\ProgramData\SFTPGo\env.d\defender.env`:



```
SFTPGO_COMMON__DEFENDER__ENABLED=true
SFTPGO_COMMON__DEFENDER__DRIVER=provider
SFTPGO_COMMON__DEFENDER__BAN_TIME=30
SFTPGO_COMMON__DEFENDER__THRESHOLD=15
```

Restart the SFTPGo service on each node after saving the file.

Important

If the SFTP Virtual Service is configured in layer 4 SNAT mode, all connections appear to SFTPGo to originate from the load balancer. The Defender will therefore see the load balancer's IP address rather than the client's, and a single misbehaving client could cause the load balancer's address to be banned. If the Defender is to be relied upon, use the layer 7 PROXY protocol variant of VIP 1 described in [VIP 1 \(Alternative\) – SFTPGo-SFTP Using the PROXY Protocol](#) so that SFTPGo sees the true client addresses.

9.6. Step 6 – Set Up Shared File Storage

All SFTPGo nodes must read and write the same file store. The example below uses an SMB share hosted on the database server. The SFTPGo nodes in this example are standalone (workgroup) servers, so matching local accounts are created on each machine to authenticate to the share.

Note

If the servers are domain joined, use a single domain service account instead of creating matching local accounts, and skip step 1 below.

1. On **all three** servers (10.11.40.5, 10.11.40.65 and 10.11.40.66), create a local service account using the **same username and the same password** on each:

```
$Password = ConvertTo-SecureString "<service-account-password>" -AsPlainText -Force
New-LocalUser -Name "sftpgosvc" -Password $Password -PasswordNeverExpires
Add-LocalGroupMember -Group "Administrators" -Member "sftpgosvc"
```

2. On the file server (10.11.40.5), create and share the data folder:

```
New-Item -Path "C:\SFTPGo-Data\users" -ItemType Directory -Force
New-SmbShare -Name "SFTPGo-Data" -Path "C:\SFTPGo-Data" -FullAccess "sftpgosvc"
$Acl = Get-Acl "C:\SFTPGo-Data"
$Rule = New-Object System.Security.AccessControl.FileSystemAccessRule("sftpgosvc",
"FullControl", "ContainerInherit, ObjectInherit", "None", "Allow")
$Acl.SetAccessRule($Rule)
Set-Acl "C:\SFTPGo-Data" $Acl
Enable-NetFirewallRule -DisplayGroup "File and Printer Sharing"
```

3. From **each** SFTPGo node, verify that the share is reachable:

```
net use Z: \\10.11.40.5\SFTPGo-Data /user:sftpgosvc <service-account-password>
net use Z: /delete
```

- On **each** SFTPGO node, grant the service account the *Log on as a service* right: run `secpol.msc` and navigate to *Local Policies > User Rights Assignment > Log on as a service*, then add `sftpgosvc`.
- On **each** SFTPGO node, change the log-on account of the SFTPGO service: run `services.msc`, open the properties of the **SFTPGO** service, select the *Log On* tab, choose *This account* and enter `sftpgosvc` with its password. Then grant the account access to SFTPGO's own data directory and restart the service:

```
icacls "C:\ProgramData\SFTPGO" /grant "sftpgosvc:(OI)(CI)F" /T
Restart-Service sftpgo
```

- On **each** SFTPGO node, create the file `C:\ProgramData\SFTPGO\env.d\storage.env`:

```
SFTPGO_COMMON__UPLOAD_MODE=1
SFTPGO_DATA_PROVIDER__USERS_BASE_DIR=\\10.11.40.5\SFTPGO-Data\users
```



Note

`UPLOAD_MODE=1` causes uploads to be written to a temporary file and atomically renamed on completion. This prevents other nodes from reading a partially written file.

- Restart the SFTPGO service on each node.

9.7. Step 7 – Synchronize the SSH Host Keys

❗ Important

This is the single most commonly missed step in an SFTPGO load balancing deployment. Each node generates its own SSH host keys on first start. If the keys differ between nodes, SFTP clients will be presented with a different host key whenever the load balancer sends them to a different node, and will report a "host key changed" warning or refuse to connect altogether.

- On node 2, stop the service:

```
Stop-Service sftpgo
```

- From node 1, copy the six host key files to node 2:

```
$keys = "id_rsa", "id_rsa.pub", "id_ecdsa", "id_ecdsa.pub", "id_ed25519", "id_ed25519.pub"
foreach ($k in $keys) {
    Copy-Item "C:\ProgramData\SFTPGO\$k" "\\10.11.40.66\C$\ProgramData\SFTPGO\$k" -Force
}
```

- On node 2, start the service again:

```
Start-Service sftpgo
```

- Verify that the fingerprints match. Run the following on **each** node and compare the output:



```
Get-Content "C:\ProgramData\SFTPGO\logs\sftpgo.log" | Select-String "Host key.*loaded" |  
Select-Object -Last 3
```

All three fingerprints (RSA, ECDSA and Ed25519) must be identical on every node.

Note

Repeat this step for every additional node added to the cluster.

9.8. Step 8 – Enable the FTP Service

The FTP service is disabled by default. On **each** node, create the file

`C:\ProgramData\SFTPGO\env.d\ftpd.env`:

```
SFTPGO_FTPD__BINDINGS__0__PORT=2121  
SFTPGO_FTPD__BINDINGS__0__FORCE_PASSIVE_IP=10.11.40.60  
SFTPGO_FTPD__BINDINGS__0__TLS_MODE=0  
SFTPGO_FTPD__PASSIVE_PORT_RANGE__START=50000  
SFTPGO_FTPD__PASSIVE_PORT_RANGE__END=50100
```

Important

`FORCE_PASSIVE_IP` must be set to the **Virtual Service address** (10.11.40.60 in this example), not to the node's own address. In passive mode the server tells the client which address and port to open the data connection to; if a node advertised its own address, the client would bypass the load balancer.

Open the firewall on each node and restart the service:

```
netsh advfirewall firewall add rule name="SFTPGO FTP Control" dir=in action=allow protocol=TCP  
localport=2121  
netsh advfirewall firewall add rule name="SFTPGO FTP Passive" dir=in action=allow protocol=TCP  
localport=50000-50100  
Restart-Service sftpgo
```

Note

The passive port range configured here must match the range configured on the Virtual Service in [VIP 2 – SFTPGO-FTP](#).

9.9. Step 9 – Enable the WebDAV Service

The WebDAV service is also disabled by default. On **each** node, create the file

`C:\ProgramData\SFTPGO\env.d\webdavd.env`:

```
SFTPGO_WEBDAVD__BINDINGS__0__PORT=10085  
SFTPGO_WEBDAVD__BINDINGS__0__ENABLE_HTTPS=false
```

Important

Use **10085**, not 10080. SFTPGO has no default WebDAV port – the binding is disabled until you set one – but 10080 is the value used by most SFTPGO examples, so it is a natural choice. Browsers block it: 10080 is on the restricted port list used by Chrome and Firefox (it is assigned



to the Amanda backup service), and Chrome rejects it with **ERR_UNSAFE_PORT**. Since WebDAV is reached from a browser here, a VIP on 10080 would look as though it were down. Any port outside the browsers' restricted lists will do; 10085 is used throughout this guide.

Open the firewall on each node and restart the service:

```
netsh advfirewall firewall add rule name="SFTPGo WebDAV" dir=in action=allow protocol=TCP
localport=10085
Restart-Service sftpgo
```

9.10. Step 10 – Enabling Client IP Visibility (Optional)

By default the SFTPGo nodes see the load balancer's address as the source of every connection rather than the client's. This step restores the true client addresses – for logging, for auditing, or so that the Defender bans individual clients rather than the load balancer.

There are **two independent halves** to this, using two different mechanisms. Apply either or both:

Services	Mechanism	Applies when
SFTP / SCP and FTP	PROXY protocol	Only with the layer 7 variant of VIP 1 – see VIP 1 (Alternative) – SFTPGo-SFTP Using the PROXY Protocol
WebDAV	X-Forwarded-For header	Always – it works with the standard WebDAV Virtual Service however VIP 1 is configured

On **each** node, create the file `C:\ProgramData\SFTPGo\env.d\proxy.env` containing the half (or halves) you need:

```
# SFTP/SCP and FTP - only with the layer 7 PROXY protocol variant of VIP 1
SFTPGO_COMMON__PROXY_PROTOCOL=1
SFTPGO_COMMON__PROXY_ALLOWED=10.11.40.55

# WebDAV - independent of VIP 1, safe to use on its own
SFTPGO_WEBDAVD__BINDINGS__0__PROXY_ALLOWED=10.11.40.55
SFTPGO_WEBDAVD__BINDINGS__0__CLIENT_IP_PROXY_HEADER=X-Forwarded-For
```

For a clustered pair of load balancers, list both appliance addresses as a comma separated value so that the header is trusted after a failover:

```
SFTPGO_COMMON__PROXY_ALLOWED=10.11.40.55,10.11.40.56
SFTPGO_WEBDAVD__BINDINGS__0__PROXY_ALLOWED=10.11.40.55,10.11.40.56
```

Restart the SFTPGo service on each node.

9.10.1. SFTP/SCP and FTP – the PROXY Protocol

The PROXY protocol is a small header prepended to a TCP connection by a proxy, carrying the original client's



address. It is not part of the payload protocol, so both ends must agree to use it.

❗ Important

`SFTPGO_COMMON__PROXY_PROTOCOL=1` applies to the SSH and FTP listeners. With this set and the load balancer listed in `proxy_allowed`, SFTPGo expects a PROXY header on connections from the load balancer. This is why the layer 4 SNAT variant of VIP 1, which does not send one, must not be used at the same time.

If you revert to the layer 4 variant of VIP 1, remove the two `SFTPGO_COMMONPROXY_*` lines and restart the service. Leave the two `SFTPGO_WEBDAVD*` lines in place – they are unaffected.

9.10.2. WebDAV – the X-Forwarded-For Header

WebDAV is served over HTTP, so the client address is carried in an `X-Forwarded-For` header rather than a PROXY header. The layer 7 HTTP Virtual Service inserts that header by default, and `CLIENT_IP_PROXY_HEADER` tells SFTPGo to read the client address from it.

📌 Note

This half has nothing to do with the PROXY protocol and nothing to do with how VIP 1 is configured. If WebDAV is the only service whose client addresses you need, apply just the two `SFTPGO_WEBDAVD__*` lines and leave VIP 1 in layer 4 SNAT mode.

10. Appliance Configuration for SFTPGo

10.1. Layer 7 – Advanced Configuration

SFTP sessions and large file transfers sit idle for long periods, and HAProxy's default timeouts would terminate them. The client and Real Server timeouts must therefore be increased before the layer 7 Virtual Services are brought into use.

1. Using the WebUI, navigate to *Cluster Configuration > Layer 7 – Advanced Configuration*.

Connection Timeout	4000	ms	?
Client Timeout	3600000	ms	?
Real Server Timeout	3600000	ms	?

2. Set *Client Timeout* to **3600000**, i.e. 1 hour. The units for this field are milliseconds.
3. Set *Real Server Timeout* to **3600000**, i.e. 1 hour.
4. Leave all other settings at their default values.
5. Click **Update**.

📌 Note

These are global settings and affect every layer 7 Virtual Service on the appliance. If that is not desirable, leave the global values alone and instead set the timeouts per VIP: modify the VIP,



scroll down to the *Other* section, click **[Advanced]**, enable (check) the *Timeout* checkbox and set both *Client Timeout* and *Server Timeout* to 1h.

10.2. VIP 1 – SFTPGo-SFTP

This is the recommended configuration for the SFTP/SCP Virtual Service. It uses layer 4 SNAT mode for maximum throughput. If the real client IP addresses must be visible to the SFTPGo nodes, use [VIP 1 \(Alternative\) – SFTPGo-SFTP Using the PROXY Protocol](#) instead – **not** both.

10.2.1. Virtual Service (VIP) Configuration

1. Using the WebUI, navigate to *Cluster Configuration > Layer 4 – Virtual Services* and click **Add a new Virtual Service**.

Virtual Service		
Label	SFTPGo-SFTP	?
IP Address	10.11.40.60	?
Ports	2022	?
Protocol		
Protocol	TCP	?
Forwarding		
Forwarding Method	SNAT	?
		Cancel Update

2. Enter an appropriate *Label* (name) for the Virtual Service, e.g. **SFTPGo-SFTP**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **10.11.40.60**.
4. Set *Ports* to **2022**.
5. Set the *Protocol* to **TCP**.
6. Set the *Forwarding Method* to **SNAT**.
7. Click **Update** to create the Virtual Service.
8. Now click **Modify** next to the newly created VIP. The remaining settings are only available on the Modify form.
9. In the *Connection Distribution Method* section, set *Balance Mode* to **Weighted Round Robin**.
10. Scroll down to the *Persistence* section.
 - Ensure that **Enable** is checked – this is the default for layer 4 Virtual Services.
 - Set the *Timeout* to **1200**, i.e. 1200 seconds (20 minutes). The units for this field are seconds.
11. Scroll down to the *Health Checks* section.

- Leave *Check Type* set to **Connect to port**.
- Leave *Check Port* blank – the VIP's only port, 2022, will be used.

12. Leave all other settings at their default values.

13. Click **Update**.

10.2.2. Define the Associated Real Servers (RIPs)

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

Label	SFTPGo-Node1	?
Real Server IP Address	10.11.40.65	?
Real Server Port	2022	?
Weight	100	?
Minimum Connections	0	?
Maximum Connections	0	?

Cancel
Update

2. Enter an appropriate *Label* (name) for the Real Server, e.g. **SFTPGo-Node1**.
3. Set the *Real Server IP Address* field to the required IP address, e.g. **10.11.40.65**.
4. Set the *Real Server Port* field to **2022**.
5. Leave all other settings at their default values.
6. Click **Update**.
7. Repeat these steps to add the remaining SFTPGo nodes, e.g. **SFTPGo-Node2** at **10.11.40.66**.

10.3. VIP 1 (Alternative) – SFTPGo-SFTP Using the PROXY Protocol

Configure this Virtual Service **instead of** [VIP 1 – SFTPGo-SFTP](#) if the real client IP addresses must be visible to the SFTPGo nodes. It requires [Step 10 – Enabling Client IP Visibility \(Optional\)](#) to have been completed on every node first.

ⓘ Important

If the layer 4 version of VIP 1 has already been created, delete it before creating this one. A layer 4 SNAT mode VIP and a layer 7 SNAT mode VIP must not share the same RIP:PORT combination because the required firewall rules conflict.

10.3.1. Virtual Service (VIP) Configuration

1. Using the WebUI, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click **Add a new Virtual Service**.



Virtual Service		[Advanced +]
Label	SFTPGo-SFTP	?
IP Address	10.11.40.60	?
Ports	2022	?
Protocol		[Advanced +]
Layer 7 Protocol	TCP Mode	?

Cancel Update

2. Enter an appropriate *Label* (name) for the Virtual Service, e.g. **SFTPGo-SFTP**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **10.11.40.60**.
4. Set *Ports* to **2022**.
5. Set the *Layer 7 Protocol* to **TCP Mode**.
6. Click **Update** to create the Virtual Service.
7. Now click **Modify** next to the newly created VIP. The remaining settings are only available on the Modify form.
8. In the *Connection Distribution Method* section, set *Balance Mode* to **Weighted Round Robin**.
9. In the *Persistence* section, confirm that *Persistence Mode* is set to **Source IP**. This is the default for TCP mode Virtual Services.
10. Still in the *Persistence* section, click **[Advanced]** and set *Persistence timeout* to **20**, i.e. 20 minutes (1200 seconds). The default units for this field are minutes.
11. It is also recommended that *Clear Stick on Drain* is enabled (checked). SSH sessions are long lived, and this ensures that clients are rebalanced normally after a Real Server has been drained.
12. Scroll down to the *Health Checks* section.
 - Leave *Check Type* set to **Connect to port**.
 - Leave *Check Port* blank – the VIP’s only port, 2022, will be used.
13. Scroll down to the *Other* section and click **[Advanced]**.

Other		[Advanced -]
Maximum Connections	40000	?
Timeout	☐	?
Force SSL Content Inspection	☐	?
Accept Proxy Protocol	☐	?
Send Proxy Protocol	Send Proxy V1	?
Set Source Address		?
Transparent Proxy	☐	?

- Set *Send Proxy Protocol* to **Send Proxy V1**.
- Leave all other settings at their default values.
- Click **Update**.

10.3.2. Define the Associated Real Servers (RIPs)

- Using the WebUI, navigate to *Cluster Configuration > Layer 7 – Real Servers* and click **Add a new Real Server** next to the newly created VIP.

Label	SFTPGo-Node1	?
Real Server IP Address	10.11.40.65	?
Real Server Port	2022	?
Re-Encrypt to Backend	☐	?
Weight	100	?

- Enter an appropriate *Label* (name) for the Real Server, e.g. **SFTPGo-Node1**.
- Set the *Real Server IP Address* field to the required IP address, e.g. **10.11.40.65**.
- Set the *Real Server Port* field to **2022**.
- Leave all other settings at their default values.
- Click **Update**.
- Repeat these steps to add the remaining SFTPGo nodes, e.g. **SFTPGo-Node2** at **10.11.40.66**.

10.4. VIP 2 – SFTPGo-FTP

This Virtual Service handles passive mode FTP. It must listen on both the control port and the same passive data port range that is configured on the SFTPGo nodes.

10.4.1. Virtual Service (VIP) Configuration

1. Using the WebUI, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click **Add a new Virtual Service**.

Virtual Service		[Advanced +]
Label	SFTPGO-FTP	?
IP Address	10.11.40.60	?
Ports	2121,50000-50100	?
Protocol		[Advanced +]
Layer 7 Protocol	TCP Mode	?
		Cancel Update

2. Enter an appropriate **Label** (name) for the Virtual Service, e.g. **SFTPGO-FTP**.
3. Set the **Virtual Service IP Address** field to the required IP address, e.g. **10.11.40.60**.
4. Set **Ports** to **2121,50000-50100**.

Important

The passive port range must match `SFTPGO_FTPDPASSIVE_PORT_RANGESTART` and `SFTPGO_FTPDPASSIVE_PORT_RANGEEND` as configured in Step 8 – Enable the FTP Service.

5. Set the **Layer 7 Protocol** to **TCP Mode**.
6. Click **Update** to create the Virtual Service.
7. Now click **Modify** next to the newly created VIP. The remaining settings are only available on the Modify form.
8. In the **Connection Distribution Method** section, set **Balance Mode** to **Weighted Round Robin**.
9. In the **Persistence** section, confirm that **Persistence Mode** is set to **Source IP**. This is the default for TCP mode Virtual Services.

Important

Source IP persistence is mandatory here, not optional. It is what causes the passive data connection to land on the same SFTPGO node as the control connection that issued it.

10. Still in the **Persistence** section, click **[Advanced]** and set **Persistence timeout** to **20**, i.e. 20 minutes (1200 seconds). The default units for this field are minutes.
11. It is also recommended that **Clear Stick on Drain** is enabled (checked), for the same reason as for VIP 1.
12. Scroll down to the **Health Checks** section.
 - Leave **Check Type** set to **Connect to port**.
 - Leave **Check Port** blank. For a multi-port VIP the first port in the list is used, which is the FTP control port 2121.

(i) Important

Do not allow the health check to target the passive data range. Those ports are only open on a node while a transfer is in progress, so a check against them would mark healthy nodes as down.

13. Leave all other settings at their default values.
14. Click **Update**.

10.4.2. Define the Associated Real Servers (RIPs)

1. Using the WebUI, navigate to *Cluster Configuration > Layer 7 – Real Servers* and click **Add a new Real Server** next to the newly created VIP.

Label	SFTPGo-Node1	?
Real Server IP Address	10.11.40.65	?
Real Server Port		?
Re-Encrypt to Backend	☐	?
Weight	100	?

Cancel **Update**

2. Enter an appropriate *Label* (name) for the Real Server, e.g. **SFTPGo-Node1**.
3. Set the *Real Server IP Address* field to the required IP address, e.g. **10.11.40.65**.
4. Leave the *Real Server Port* field **blank**.

(i) Important

The port field must be left blank for this Virtual Service. This is a multi-port service, and leaving the port blank causes each connection to be forwarded to the same port on the Real Server that it arrived on – which is what passive mode requires.

5. Leave all other settings at their default values.
6. Click **Update**.
7. Repeat these steps to add the remaining SFTPGo nodes, e.g. **SFTPGo-Node2** at **10.11.40.66**.

10.5. VIP 3 – SFTPGo-WebDAV

10.5.1. Virtual Service (VIP) Configuration

1. Using the WebUI, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click **Add a new Virtual Service**.

Virtual Service		[Advanced +]
Label	SFTPGo-WebDAV	?
IP Address	10.11.40.60	?
Ports	10085	?
Protocol		[Advanced +]
Layer 7 Protocol	HTTP Mode	?

Cancel Update

2. Enter an appropriate *Label* (name) for the Virtual Service, e.g. **SFTPGo-WebDAV**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **10.11.40.60**.
4. Set *Ports* to **10085**.



Note

This must match `SFTPGO_WEBDAVDBINDINGS0__PORT` as configured in Step 9 – Enable the WebDAV Service.

5. Leave the *Layer 7 Protocol* set to **HTTP Mode**.
6. Click **Update** to create the Virtual Service.
7. Now click **Modify** next to the newly created VIP. The remaining settings are only available on the Modify form.
8. In the *Connection Distribution Method* section, set *Balance Mode* to **Weighted Round Robin**.
9. In the *Persistence* section, leave *Persistence Mode* set to **HTTP Cookie**. This is the default for HTTP mode Virtual Services.
10. Scroll down to the *Health Checks* section.
 - Set *Check Type* to **Negotiate HTTP (GET)**.
 - Set *Request to Send* to `/`.
 - Set *Response Expected* to **Authentication error**, leaving the comparison set to **Equals**.



Note

An unauthenticated GET of the WebDAV root returns HTTP 401 with a short body beginning **Authentication error:**. Because 401 is neither a 2xx nor a 3xx status, *Response Expected* cannot be left blank – if it were, every node would be marked down.

The appliance's negotiate checks match a string **within the response data**; they cannot assert an HTTP status code directly. The 401 is therefore detected indirectly, by matching text that only appears in SFTPGo's 401 response. If a future SFTPGo release changes that text, either update this string or fall back to a **Connect to port** check on port 10085.

11. Scroll down to the *Other* section and click **[Advanced]**.
12. Confirm that **Set X-Forwarded-For Header** is enabled. This is the default for new layer 7 Virtual Services, and it is what allows SFTPGo to log the real client address for WebDAV requests when **CLIENT_IP_PROXY_HEADER** has been configured.
13. Leave all other settings at their default values.
14. Click **Update**.

Note

HTTP cookie persistence only pins clients that store and return cookies – browsers do, but command line tools such as [curl](#) and the Windows WebDAV mini-redirector may not. This is not a problem for SFTPGo, because all nodes share the same data provider and the same file store, so a request served by any node returns the same result. If persistence is nevertheless required for all WebDAV clients, set the **Persistence Mode** to **Source IP** or to **HTTP Cookie and Source IP** instead.

10.5.2. Define the Associated Real Servers (RIPs)

1. Using the WebUI, navigate to *Cluster Configuration > Layer 7 – Real Servers* and click **Add a new Real Server** next to the newly created VIP.

Label	SFTPGo-Node1	
Real Server IP Address	10.11.40.65	
Real Server Port	10085	
Re-Encrypt to Backend	☐	
Weight	100	

2. Enter an appropriate **Label** (name) for the Real Server, e.g. **SFTPGo-Node1**.
3. Set the **Real Server IP Address** field to the required IP address, e.g. **10.11.40.65**.
4. Set the **Real Server Port** field to **10085**.
5. Leave all other settings at their default values.
6. Click **Update**.
7. Repeat these steps to add the remaining SFTPGo nodes, e.g. **SFTPGo-Node2** at **10.11.40.66**.

10.6. Finalizing the Configuration

To apply the new layer 7 settings, HAProxy must be reloaded. This can be done using the button in the "Commit changes" box at the top of the screen or by using the **Restart Services** menu option:

1. Using the WebUI, navigate to: **Maintenance > Restart Services**.

2. Click **Reload HAProxy**.

11. Testing & Verification

Note

For additional guidance on diagnosing and resolving any issues you may have, please also refer to [Diagnostics & Troubleshooting](#).

11.1. Verifying the SFTPGo Cluster

Complete these checks **before** testing through the load balancer. They confirm that the nodes really are interchangeable, and they isolate SFTPGo issues from load balancer issues.

Host key consistency

Run the following on each node and compare the output. All three fingerprints must match exactly across all nodes:

```
Get-Content "C:\ProgramData\SFTPGo\logs\sftpgo.log" | Select-String "Host key.*loaded" | Select-Object -Last 3
```

Shared data provider

1. Log in to the WebAdmin interface on each node in turn – e.g. <http://10.11.40.65:8080/web/admin> and <http://10.11.40.66:8080/web/admin> – using the same administrator credentials. If the shared data provider is working, the account created in [Step 4 – Create the Administrator Account](#) will be accepted by every node, with no initial setup screen.
2. On node 1's WebAdmin, create a test user, e.g. username **testuser**.
3. Refresh the **Users** page in node 2's WebAdmin. **testuser** must appear there without any further action.

Shared file storage

Upload a file directly to node 1, bypassing the load balancer, and confirm that it is visible when connecting directly to node 2.

11.2. Using System Overview

The System Overview can be viewed in the WebUI. It shows a graphical view of all Virtual Services & the associated Real Servers (i.e. the SFTPGo nodes) and shows the state/health of each server as well as the overall state of each cluster. The example below shows that all servers are healthy (green) and available to accept connections:

	VIRTUAL SERVICE ↕	IP ↕	PORTS ↕	CONNS ↕	PROTOCOL ↕	METHOD ↕	MODE ↕	
↑	SFTPGo-SFTP	10.11.40.60	2022	0	TCP	Layer 4	SNAT	
	REAL SERVER	IP	PORTS	WEIGHT	CONNS			
↑	SFTPGo-Node1	10.11.40.65	2022	100	0	Drain	Halt	
↑	SFTPGo-Node2	10.11.40.66	2022	100	0	Drain	Halt	
↑	SFTPGo-FTP	10.11.40.60	2121,5000..	0	TCP	Layer 7	Proxy	
	REAL SERVER	IP	PORTS	WEIGHT	CONNS			
↑	SFTPGo-Node1	10.11.40.65	2121,50000..	100	0	Drain	Halt	
↑	SFTPGo-Node2	10.11.40.66	2121,50000..	100	0	Drain	Halt	
↑	SFTPGo-WebDAV	10.11.40.60	10085	0	HTTP	Layer 7	Proxy	
	REAL SERVER	IP	PORTS	WEIGHT	CONNS			
↑	SFTPGo-Node1	10.11.40.65	10085	100	0	Drain	Halt	
↑	SFTPGo-Node2	10.11.40.66	10085	100	0	Drain	Halt	

11.3. Accessing SFTPGo via the Load Balancer

Verify that you're able to successfully access all load balanced applications and services via the Virtual Services on the load balancer.

Note

Make sure that DNS is updated so that any FQDNs used point to the VIPs rather than to individual SFTPGo nodes.

The following tests exercise each Virtual Service end to end, uploading a file, downloading it again and comparing checksums. They assume a Linux or macOS client; equivalent tools are available for Windows.

Prepare the test files on the client

```
echo "roundtrip $(date -Iseconds)" > /tmp/up.txt
head -c 1M /dev/urandom > /tmp/up.bin
md5sum /tmp/up.txt /tmp/up.bin
```

VIP 1 – SFTP

```
sftp -P 2022 testuser@10.11.40.60 <<'EOF'
pwd
ls -la
mkdir sftp_test
cd sftp_test
put /tmp/up.txt
put /tmp/up.bin
ls -la
get up.txt /tmp/sftp_down.txt
```

```
get up.bin /tmp/sftp_down.bin
bye
EOF

md5sum /tmp/up.txt /tmp/sftp_down.txt /tmp/up.bin /tmp/sftp_down.bin
```

The checksum of each uploaded file and its downloaded copy must match.

VIP 2 – FTP

```
FTP=ftp://10.11.40.60:2121
U=testuser:<test-user-password>

curl -s --ftp-pasv -u "$U" "$FTP/"
curl -s --ftp-pasv --ftp-create-dirs -u "$U" -T /tmp/up.bin "$FTP/ftp_test/up.bin"
curl -s --ftp-pasv -u "$U" "$FTP/ftp_test/up.bin" -o /tmp/ftp_down.bin
md5sum /tmp/up.bin /tmp/ftp_down.bin
```

Note

If the directory listing succeeds but the transfer stalls, the passive data connection is not reaching the same node as the control connection. Re-check that `FORCE_PASSIVE_IP` is set to the VIP address, that the VIP's port list includes the full passive range, that the Real Server port field is blank, and that source IP persistence is enabled.

VIP 3 – WebDAV

```
DAV=http://10.11.40.60:10085
U=testuser:<test-user-password>

curl -s -o /dev/null -w "PROPFIND %{http_code}\n" -u "$U" -X PROPFIND -H "Depth: 1" "$DAV/"
curl -s -o /dev/null -w "MKCOL %{http_code}\n" -u "$U" -X MKCOL "$DAV/webdav_test/"
curl -s -o /dev/null -w "PUT %{http_code}\n" -u "$U" -T /tmp/up.bin "$DAV/webdav_test/up.bin"
curl -s -u "$U" "$DAV/webdav_test/up.bin" -o /tmp/dav_down.bin
md5sum /tmp/up.bin /tmp/dav_down.bin
```

11.4. Verifying Client IP Visibility

To confirm which source address the SFTPGO nodes are recording, connect to a Virtual Service and then inspect the login entries on each node:

```
Get-Content "C:\ProgramData\SFTPGO\logs\sftpgo.log" -Tail 1000 | Select-String '"sender":"login"'
```

SFTPGO writes one of these for every successful login, and the address appears in the entry's `ip` field:

- With the layer 4 SNAT variant of VIP 1, `ip` will be the load balancer's address.
- With the layer 7 PROXY protocol variant of VIP 1 and `proxy.env` in place, `ip` will be the real client address.

Tip

Use the login entries rather than the transfer entries for this check. A login is written on every



connection, whereas the transfer entries ("**sender**" : "**Upload**" and "**sender**" : "**Download**", which carry the address as `remote_addr`) only exist while a transfer is actually running – so an empty result tells you nothing about whether client IP visibility is working.

11.5. Testing Failover

1. Confirm that all nodes show as **Up** in System Overview.
2. On the node currently serving traffic, stop the service:

```
Stop-Service sftpgo
```

3. Wait approximately 15 seconds for the health check to fail, and confirm in System Overview that the node is marked down.
4. Open a new SFTP session to the VIP. It must succeed, served by the remaining node.
5. Start the service again and confirm that the node returns to **Up**:

```
Start-Service sftpgo
```

6. Repeat the test with the other node.

Note

Existing sessions on a failed node are not migrated – the client must reconnect. This is inherent to file transfer protocols and is not specific to this deployment.

12. 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.

13. Further Documentation

For additional information, please refer to the [Administration Manual](#).

The following SFTPGo documentation was referenced in the preparation of this guide:

- [SFTPGo installation guide](#)
- [SFTPGo configuration file reference](#)
- [SFTPGo initial configuration](#)
- [SFTPGo data provider management](#)
- [SFTPGo REST API documentation](#)
- [SFTPGo metrics and telemetry](#)



- [SFTPGO CLI reference](#)
- [SFTPGO OpenAPI specification](#) – defines the `/healthz` endpoint discussed in [Health Checks](#)
- [SFTPGO discussion #665](#) – behaviour of the `/healthz` endpoint
- [SFTPGO issue #1977](#) – health checking from HAProxy when the PROXY protocol is enabled



14. Appendix

14.1. SFTPGO Environment File Reference

All of the following files live in `C:\ProgramData\SFTPGO\env.d\` on each SFTPGO node. SFTPGO reads every file in this directory at start up, so the service must be restarted after any change.

database.env

```
SFTPGO_DATA_PROVIDER__DRIVER=postgresql
SFTPGO_DATA_PROVIDER__NAME=sftpgo
SFTPGO_DATA_PROVIDER__HOST=10.11.40.5
SFTPGO_DATA_PROVIDER__PORT=5432
SFTPGO_DATA_PROVIDER__USERNAME=sftpgo
SFTPGO_DATA_PROVIDER__PASSWORD=<database-password>
```

defender.env

```
SFTPGO_COMMON__DEFENDER__ENABLED=true
SFTPGO_COMMON__DEFENDER__DRIVER=provider
SFTPGO_COMMON__DEFENDER__BAN_TIME=30
SFTPGO_COMMON__DEFENDER__THRESHOLD=15
```

storage.env

```
SFTPGO_COMMON__UPLOAD_MODE=1
SFTPGO_DATA_PROVIDER__USERS_BASE_DIR=\\10.11.40.5\SFTPGO-Data\users
```

ftpd.env

```
SFTPGO_FTPD__BINDINGS__0__PORT=2121
SFTPGO_FTPD__BINDINGS__0__FORCE_PASSIVE_IP=10.11.40.60
SFTPGO_FTPD__BINDINGS__0__TLS_MODE=0
SFTPGO_FTPD__PASSIVE_PORT_RANGE__START=50000
SFTPGO_FTPD__PASSIVE_PORT_RANGE__END=50100
```

webdavd.env

```
SFTPGO_WEBDAVD__BINDINGS__0__PORT=10085
SFTPGO_WEBDAVD__BINDINGS__0__ENABLE_HTTPS=false
```

proxy.env – optional, and made up of two independent halves (see [Step 10 – Enabling Client IP Visibility \(Optional\)](#))

```
# SFTP/SCP and FTP - only with the layer 7 PROXY protocol variant of VIP 1
SFTPGO_COMMON__PROXY_PROTOCOL=1
```



```
SFTPGO_COMMON__PROXY_ALLOWED=10.11.40.55,10.11.40.56

# WebDAV - independent of VIP 1, safe to use on its own
SFTPGO_WEBDAVD__BINDINGS__0__PROXY_ALLOWED=10.11.40.55,10.11.40.56
SFTPGO_WEBDAVD__BINDINGS__0__CLIENT_IP_PROXY_HEADER=X-Forwarded-For
```

14.2. Useful SFTPGo Service Commands

```
Get-Service sftpgo
Start-Service sftpgo
Stop-Service sftpgo
Restart-Service sftpgo
```

14.3. Additional Considerations

Adding further nodes – to add a third or subsequent node: install SFTPGo, copy the environment files from an existing node, set the SFTPGo service log-on account to the shared service account, copy the SSH host keys from node 1, add the node's address to [pg_hba.conf](#) on the PostgreSQL server, and finally add it as a Real Server to each Virtual Service.

Health check log noise – the TCP connect checks against port 2022 cause SFTPGo to log a debug level [connection_failed](#) / [no_auth_tried](#) entry on every node at every check interval. This is expected and harmless. Setting the SFTPGo log level to [info](#) removes it.

Database single point of failure – a single PostgreSQL instance is acceptable for a lab, but in production it should itself be made highly available.

Workgroup versus domain – matching local service accounts are only required because the servers in this example are standalone. On domain joined servers, use a single domain service account for the SFTPGo service and for share access.

WebAdmin and the REST API – neither is load balanced in this deployment. If highly available access to either is required, add a further layer 7 HTTP Virtual Service for TCP 8080 following the pattern used for the WebDAV Virtual Service.



14.4. 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 is fully configured first, then the Secondary appliance can be added to create an HA pair. Once the HA pair is configured, load balanced services must be configured and modified on the Primary appliance. The Secondary appliance will be automatically kept in sync.

Note

For Enterprise Azure, the HA pair should be configured first. For more information, please refer to the Azure Quick Start/Configuration Guide available in the [documentation library](#)

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.

14.4.1. Non-Replicated Settings

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:

WebUI Main Menu Option	Sub Menu Option	Description
Local Configuration	Hostname & DNS	Hostname and DNS settings
Local Configuration	Network Interface Configuration	Interface IP addresses, bonding configuration and VLANs
Local Configuration	Routing	Default gateways and static routes
Local Configuration	System Date & time	Time and date related settings
Local Configuration	Physical – Advanced Configuration	Various appliance settings
Local Configuration	Portal Management	Portal management settings
Local Configuration	Security	Security settings
Local Configuration	SNMP Configuration	SNMP settings
Local Configuration	Graphing	Graphing settings
Local Configuration	License Key	Appliance licensing
Maintenance	Backup & Restore	Local XML backups
Maintenance	Software Updates	Appliance software updates
Maintenance	Firewall Script	Firewall (iptables) configuration
Maintenance	Firewall Lockdown Wizard	Appliance management lockdown settings

Important

Make sure that where any of the above have been configured on the Primary appliance, they're also configured on the Secondary.

14.4.2. Configuring the HA Clustered Pair

⚠ Important

During HA pairing, all WebUI users and passwords are synchronized from the Primary to the Secondary. After clustering completes (you will be logged out of the Secondary when this occurs), the Primary's credentials should be used to login to both nodes.

📌 Note

If you have already run the firewall lockdown wizard on either appliance, you'll need to ensure that it is temporarily disabled on both appliances whilst performing the pairing process.

1. Deploy a second appliance that will be the Secondary and configure initial network settings.
2. Using the WebUI on the Primary appliance, navigate to: **Cluster Configuration > High-Availability Configuration**.

Create a Clustered Pair

 **LOADBALANCER**

Local IP address

10.11.40.55

IP address of new peer

10.11.40.56

Password for *loadbalancer* user on peer

.....

Add new node

3. Specify the IP address and the *loadbalancer* user's password for the Secondary (peer) appliance as shown in the example above.
4. Click **Add new node**.
5. The pairing process now commences as shown below:

Create a Clustered Pair

 **LOADBALANCER** Primary

IP: 10.11.40.55

Attempting to pair..

 **LOADBALANCER** Secondary

IP: 10.11.40.56

Local IP address

10.11.40.55

IP address of new peer

10.11.40.56

Password for *loadbalancer* user on peer

.....

configuring

6. Once complete, the following will be displayed on the Primary appliance:

High Availability Configuration - primary

 **LOADBALANCER**

Primary

IP: 10.11.40.55

 **LOADBALANCER**

Secondary

IP: 10.11.40.56

Break Clustered Pair

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

Note

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 [Configuring High Availability](#).

Note

For details on testing and verifying HA, please refer to [Clustered Pair Diagnostics](#).

15. Document Revision History

Version	Date	Change	Reason for Change	Changed By
1.0.0	16 September 2026	Initial version		RJC





Visit us: www.loadbalancer.org

Phone us: +44 (0)330 380 1064

Phone us: +1 833 274 2566

Email us: info@loadbalancer.org

Follow us: [@loadbalancer.org](https://twitter.com/loadbalancer.org)

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.

