# Table of Contents

1. About this Guide ................................................................. 4  
2. Loadbalancer.org Appliances Supported ............................ 4  
3. Loadbalancer.org Software Versions Supported .................. 4  
4. Cloudian HyperStore Software Versions Supported .......... 4  
5. Cloudian HyperStore .......................................................... 4  
6. Load Balancing Cloudian HyperStore ................................. 4  
   - Note on 'Direct to Node' GSLB Deployments .................. 5  
   - Persistence (aka Server Affinity) .................................. 5  
   - Virtual Service (VIP) Requirements ............................. 5  
   - Port Requirements ...................................................... 5  
   - Health Checks ......................................................... 5  
7. Performance and Sizing for a Virtual Load Balancer Deployment with Cloudian HyperStore ............................ 6  
8. Deployment Concept .......................................................... 6  
   - Layer 7 SNAT Mode (Default) ...................................... 6  
   - Layer 4 DR Mode / Direct Routing (Alternative for Read-Intensive Deployments) .................................................. 7  
   - 'Direct to Node' GSLB .................................................. 8  
   - Multi-Site GSLB and Location Affinity .......................... 9  
9. Loadbalancer.org Appliance — the Basics .......................... 10  
   - Virtual Appliance ....................................................... 10  
   - Initial Network Configuration ..................................... 11  
   - Accessing the WebUI ................................................... 11  
10. Appliance Configuration for Cloudian HyperStore — Using Layer 7 SNAT Mode ................................................. 13  
   - Enabling Multithreaded Load Balancing ....................... 13  
   - The Duplicate Service Function .................................. 13  
   - Configuring VIP 1 – Cloudian Management Console ....... 14  
   - Configuring VIP 2 – S3 Client Requests (HTTP) ............. 15  
   - Configuring VIP 3 – S3 Client Requests (HTTPS) ........... 17  
   - Configuring VIP 4 – Admin API Requests ...................... 18  
   - Configuring VIP 5 – Identity and Access Management Service (HTTP) ......................................................... 19  
   - Configuring VIP 6 – Identity and Access Management Service (HTTPS) ......................................................... 20  
   - Finalizing the Configuration ....................................... 21  
   - Testing the Configuration .......................................... 21  
   - Overview ................................................................. 22  
   - Appliance Configuration .............................................. 23  
   - Testing the Configuration .......................................... 26  
12. Appliance Configuration for Cloudian HyperStore — Using 'Direct to Node' GSLB ..................................................... 27  
   - Overview ................................................................. 27  
   - Multi-Site Variant ....................................................... 28  
   - Health Checks ......................................................... 29  
   - Handling Multiple Subdomains, Including Wildcard Subdomains ................................................................. 29  
   - Appliance Configuration .............................................. 30  
   - Testing the Configuration .......................................... 37  
13. Appliance Configuration for Cloudian HyperStore — Using Multi-Site GSLB and Location Affinity ........................................ 37  
   - Conceptual Overview .................................................. 37  
   - Appliance Configuration .............................................. 40  
14. Testing & Verification .......................................................... 49
1. About this Guide

This guide details the steps required to configure a load balanced Cloudian HyperStore environment utilizing Loadbalancer.org appliances. It covers the configuration of the load balancers and also any Cloudian HyperStore 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

Our hardware and virtual products, 10G models and above, can be used with Cloudian HyperStore. For full specifications of available models please refer to: https://www.loadbalancer.org/products

3. Loadbalancer.org Software Versions Supported

- V8.5.1 and later

4. Cloudian HyperStore Software Versions Supported

- Cloudian HyperStore – all versions

5. Cloudian HyperStore

Cloudian is a file and object storage company specialising in S3 (Simple Storage Service) API storage systems. The technology allows companies of all sizes realise the benefits of object storage in their own data centres. The Cloudian HyperStore Operating Environment software provides scalable enterprise object storage, with 100% native Amazon S3-API support.

Cloudian HyperStore architecture supports High Availability (HA) clustering by putting a load balancer in front of it. Load balancers monitor and perform health checks on a node to ensure traffic is routed correctly to healthy nodes. Without the use of a load balancer, an offline or failed node would still receive traffic, causing failures.

A variety of load balancing methods are currently supported by Cloudian HyperStore, dependent on customer infrastructure, including layer 4, layer 7, Geo GSLB/location affinity, and GSLB ‘direct to node’. The HyperStore services that should be load balanced are: S3, Cloudian Management Console (CMC), Admin-API, and Identity and Access Management service.

6. Load Balancing Cloudian HyperStore

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

Cloudian HyperStore can be load balanced in a variety of fundamentally different ways.

The remainder of this section describes general information about load balancing HyperStore, covering some of the commonalities between, and HyperStore services of interest to, the different load balancing methods.

Section 8, “Deployment Concept” describes each of the different specific load balancing methods.
Note on 'Direct to Node' GSLB Deployments

The 'Direct to Node' GSLB style of deployment is unique. It does not make use of load balancing in the same way as the other supported methods for load balancing Cloudian HyperStore. Virtual service, port, and health check information provided in the rest of this chapter is not applicable to this style of deployment.

For specific information relevant to this deployment type, refer to section 'Direct to Node' GSLB.

Persistence (aka Server Affinity)

The CMC service requires persistence to ensure that clients connect to the same HyperStore instance for the duration of their CMC session. This is a requirement for CMC to function correctly.

Client persistence is not required for HyperStore's other services (i.e. everything other than the CMC service) and should not be enabled.

Virtual Service (VIP) Requirements

To provide load balancing for Cloudian HyperStore, the following VIPs are required:

- **CMC**: for Cloudian Management Console requests
- **S3-HTTP**: handles requests from S3 client applications via HTTP
- **S3-HTTPS**: handles requests from S3 client applications via HTTPS
- **API**: handles API requests via HTTPS

The following VIPs are optional for HyperStore version 7.1.x and earlier but **mandatory** for version 7.2.x and above:

- **IAM-HTTP**: Identity and Access Management service traffic via HTTP
- **IAM-HTTPS**: Identity and Access Management service traffic via HTTPS

Port Requirements

The following table shows the ports that are load balanced:

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocols</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>TCP/HTTP</td>
<td>Requests from S3 client applications</td>
</tr>
<tr>
<td>443</td>
<td>TCP/HTTPS</td>
<td>Requests from S3 client applications</td>
</tr>
<tr>
<td>8443</td>
<td>TCP/HTTPS</td>
<td>Requests from clients</td>
</tr>
<tr>
<td>8888</td>
<td>TCP/HTTP</td>
<td>Requests from clients</td>
</tr>
<tr>
<td>16080</td>
<td>TCP/HTTP</td>
<td>IAM service traffic</td>
</tr>
<tr>
<td>16443</td>
<td>TCP/HTTPS</td>
<td>IAM service traffic</td>
</tr>
<tr>
<td>19443</td>
<td>TCP/HTTPS</td>
<td>Admin API requests</td>
</tr>
</tbody>
</table>

Health Checks

The S3-HTTP service uses the "Negotiate HTTP (HEAD)" health check, while the S3-HTTPS and API services both use the "Negotiate HTTPS (HEAD)" health check.
The CMC service uses the "Negotiate HTTPS (OPTIONS)" health check.

The health check for the API virtual service should be configured with the credentials for the sysadmin user so that it can authenticate against the API service in order to successfully check its health. This is described fully in section Configuring VIP 4 – Admin API Requests.

The IAM services use the default "Connect to port" health check.

The GSLB / location affinity based deployment types make use of an "Intelligent Site Health Check" to determine the health of a given site's HyperStore deployment. Configuring this health check is described as part of the instructions for configuring that deployment type.

7. Performance and Sizing for a Virtual Load Balancer Deployment with Cloudian HyperStore

The Loadbalancer.org appliance can be deployed as a virtual appliance.

To achieve the best level of performance and throughput when load balancing a Cloudian HyperStore deployment, the Loadbalancer.org appliance should be configured to actively use multiple CPU cores for the load balancing process. This must be considered when initially deploying and sizing virtual appliances.

A virtual host should be allocated a minimum of 4 vCPUs.

8. Deployment Concept

Cloudian HyperStore can be load balanced in a variety of different ways. The different deployment types are described below.

Deployment Types Overview and Quick Links:

- **Layer 7 SNAT Mode (Default)**
- **Layer 4 DR Mode / Direct Routing (Alternative for Read-Intensive Deployments)**
- 'Direct to Node' GSLB
- **Multi-Site GSLB and Location Affinity**

**Layer 7 SNAT Mode (Default)**

- The default, traditional, and recommended deployment type
- Flexible and simple: the load balancer acts as a reverse proxy
- Use this deployment method unless you have a specific reason not to
VIPs = Virtual IP Addresses

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

Note: This deployment type can also be used for multi-site style deployments. See the overview of Multi-Site GSLB and Location Affinity for information on extending this deployment type across multiple sites.

Full instructions on setting up this type of deployment can be found in Section 10, “Appliance Configuration for Cloudian HyperStore – Using Layer 7 SNAT Mode”.

Layer 4 DR Mode / Direct Routing (Alternative for Read-Intensive Deployments)

- Useful load balancing method for deployments that are read-intensive, with a large 'reply traffic' to 'request traffic' ratio
- Reply traffic flows directly from the HyperStore nodes to the clients, removing the load balancer as a potential bottleneck
VIPs = Virtual IP Addresses

**Note**

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

A full explanation and instructions on setting up this type of deployment can be found in Section 11, “Appliance Configuration for Cloudian HyperStore – Using Layer 4 DR Mode / Direct Routing (Alternative for Read-Intensive Deployments”).

**'Direct to Node' GSLB**

- Round-robin DNS with health checking
- Client traffic flows directly to the Cloudian Nodes and directly back again – the load balancer is entirely removed from the path of HyperStore traffic
- Useful when network throughput is paramount while retaining the load balancer’s active health checking of HyperStore nodes
The load balancer can be deployed as a single unit, although Loadbalancer.org recommends a clustered pair for resilience & high availability. Please refer to the section Configuring HA - Adding a Slave Appliance in the appendix for more details on configuring a clustered pair.

This deployment type can also be used for multi-site style deployments. See the full explanation linked to below for details.

A full explanation and instructions on setting up this type of deployment can be found in Section 12, “Appliance Configuration for Cloudian HyperStore – Using ‘Direct to Node’ GSLB”.

Multi-Site GSLB and Location Affinity

- Uses DNS to provide high availability across multiple sites
- Assumes that each site’s own HyperStore cluster is being load balanced using Layer 7 SNAT Mode (Default)
- Clients at a site with a failed HyperStore service are automatically directed to a functioning site
- Provides optional location affinity (by default) to ensure clients connect to their local HyperStore service
9. Loadbalancer.org Appliance – the Basics

**Virtual Appliance**

A fully featured, fully supported 30 day trial is available if you are conducting a PoC (Proof of Concept) deployment. The VA is currently available for VMware, Virtual Box, Hyper-V, KVM 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.

---

VIPs = Virtual IP Addresses

**Note**

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

A full explanation and instructions on setting up this type of deployment can be found in Section 13, “Appliance Configuration for Cloudian HyperStore – Using Multi-Site GSLB and Location Affinity”.

---
Please refer to the Administration Manual and view the ReadMe.txt text file included in the VA download for more detailed information on deploying the VA using various Hypervisors.

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

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

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

Accessing the WebUI
The WebUI is accessed using a web browser. Appliance authentication is based on Apache .htaccess files. User admin tasks such as adding users and changing passwords can be performed using the WebUI menu option: Maintenance > Passwords.

A number of compatibility issues have been found with various versions of Internet Explorer. The WebUI has been tested and verified using both Chrome & Firefox.

If required, users can also be authenticated against LDAP, LDAPS, Active Directory or Radius. For more information please click here.

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


2. Log in to the WebUI:

   Username: loadbalancer

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

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

Once logged in, the WebUI will be displayed as shown below:
3. You’ll be asked if you want to run the Setup Wizard. If you click **Accept** the Layer 7 Virtual Service configuration wizard will start. If you want to configure the appliance manually, simple click **Dismiss**.

**Main Menu Options**

**System Overview** - Displays a graphical summary of all VIPs, RIPS and key appliance statistics

**Local Configuration** - Configure local host settings such as IP address, DNS, system time etc.

**Cluster Configuration** - Configure load balanced services such as VIPs & RIPS

**Maintenance** - Perform maintenance tasks such as service restarts and taking backups

**View Configuration** - Display the saved appliance configuration settings

**Reports** - View various appliance reports & graphs

---

**Note:** The WebUI for the VA is shown, the hardware and cloud appliances are very similar. The yellow licensing related message is platform & model dependant.
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 the section Configuring HA - Adding a Slave Appliance of the appendix.

10. Appliance Configuration for Cloudian HyperStore – Using Layer 7 SNAT Mode

Enabling Multithreaded Load Balancing

The Loadbalancer.org appliance should be configured to actively use multiple CPU cores for the load balancing process. This is required to achieve the high level of performance and throughput required when load balancing a Cloudian HyperStore deployment.

Note  
A virtual host should be allocated a minimum of 4 vCPUs.

A minimum of 4 threads should be defined. The number of threads can be set as high as the number of threads available to the system (setting the value even higher than that will not increase performance).

To enable multithreaded mode from the WebUI:

1. Navigate to Cluster Configuration > Layer 7 - Advanced Configuration
2. Check the Enable Multithreading checkbox
3. Set Number of Threads to a minimum of 4
4. Click Update to apply the changes

The Duplicate Service Function

The instructions throughout the remainder of this section make use of the Duplicate Service function. This allows an existing virtual service to be "duplicated", along with all real servers associated to that service. This can save a considerable amount of time when configuring the load balancer to work with a product like HyperStore, where multiple virtual services are required which all share the same pool of back end servers.

Care must be taken as the Duplicate Service function is a double-edged sword: configuration errors can easily propagate throughout an entire deployment. A misconfigured virtual service that is "duplicated" can spread misconfiguration throughout the whole setup.
Configuring VIP 1 – Cloudian Management Console

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. cmc.cloudian-hyperstore

3. Set the Virtual Service IP Address field to the required IP address, e.g. 192.168.87.67

4. Set the Ports field to 8888,8443

5. Set the Layer 7 Protocol to TCP Mode

6. Click Update to create the virtual service

7. Click Modify next to the newly created VIP

8. In the Persistence section click Advanced to expand the section

9. Set Persistence Mode to Source IP

10. Set Persistence Timeout to 30

11. Set Health Checks to Negotiate HTTPS (OPTIONS)

12. Set Request to send to /Cloudian/login.htm

13. Click the Advanced button to expand the Health Checks menu

14. Set Check Port to 8443
The Host Header field should be set if appropriate, such as with your S3 endpoint name, for example 's3-region1.domain'.

15. Click Update

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. cloudian-node1
3. Set the Real Server IP Address field to the required IP address, e.g. 10.10.10.11
4. Click Update
5. Repeat these steps to add additional HyperStore nodes as real servers as required

Configuring VIP 2 – S3 Client Requests (HTTP)

Configuring the Virtual Service (VIP)
1. Using the web user interface, navigate to Cluster Configuration > Layer 7 – Virtual Services and click Modify next to the previously created CMC VIP
2. Click Duplicate Service and confirm when prompted
3. Define the Label for the new virtual service as required, e.g. `s3.cloudian-hyperstore`

4. Set the Virtual Service IP Address field to the required IP address, e.g. `192.168.87.67`

5. Set the Ports field to `80`

6. Set Persistence Mode to None

   **Note** It is important to verify that the Persistence Mode has been correctly set to None. If this step is skipped, the configuration error will propagate throughout the rest of the configuration.

7. Set Health Checks to Negotiate HTTP (HEAD)

8. Set Request to send to `/healthCheck`

9. In the Health Checks section click Advanced to expand the menu

10. Clear the Check Port field to leave it empty

   **Note** The Host Header field should be set if appropriate, such as with your S3 endpoint name, for example ‘s3-region1.domain’.
11. Click **Update**

**Note**

If a HyperStore deployment *requires* the true source IP addresses of clients to be logged for S3 requests, for example so that S3 bucket policies or billing whitelisting can be used, then the PROXY protocol can be used to achieve this.

An explanation and instructions on setting up this optional feature can be found in the section *Using the PROXY Protocol to Retain Client IP Addresses* of the appendix.

### Configuring VIP 3 – S3 Client Requests (HTTPS)

#### Configuring the Virtual Service (VIP)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click **Modify** next to the previously created S3 VIP

2. Click **Duplicate Service** and confirm when prompted

3. Define the *Label* for the new virtual service as required, e.g. `https.s3.cloudian-hyperstore`

4. Set the *Virtual Service IP Address* field to the required IP address, e.g. `192.168.87.67`

5. Set the *Ports* field to **443**

6. Check that the *Persistence Mode* is pre-set to **None**

   (If this is not the case, delete this VIP and carefully retrace the instructions starting from section *Configuring VIP 2 – S3 Client Requests (HTTP)*, ensuring that *all* instructions, including the persistence mode settings, are followed)

7. Set *Health Checks* to **Negotiate HTTPS (HEAD)**
Clicking on **Advanced** reveals a **Host Header** field, which should be set if appropriate, such as with your S3 endpoint name, for example ‘s3-region1.domain’.

8. **Click Update**

### Configuring VIP 4 – Admin API Requests

#### Health Check Credentials

A valid username and password combination is required to health check the HyperStore admin API service. The specifics vary depending on the version of HyperStore in question, as explained below.

**HyperStore Versions Up to and Including 7.2.1**

The following default credentials should be used, unless they have been modified:

**Username**: sysadmin

**Password**: public

**HyperStore Version 7.2.2 and Later**

Starting with HyperStore version 7.2.2, the password for the **sysadmin** account is randomly generated. It can be found as follows:

1. Login to the HyperStore puppet master node as **root**, or **sa_admin** if root access is disabled
2. Execute the following command to retrieve the current **sysadmin** password:

   ```bash
   [root@hs1 7.2.4]# hsctl config get admin.auth
   {
   "base64": "c3lzYWRtaW46SWxpa2VmbGFtaW5nb3MuVGhleSdyZV9uZWF0PQo=",
   "password": "CQA4xFerdMUn8lvoZrbBC6HZ5[D=",
   "username": "sysadmin"
   }
   ```
3. Make a note of the password: it is used in the following section when configuring the virtual service

#### Configuring the Virtual Service (VIP)

1. Using the web user interface, navigate to **Cluster Configuration > Layer 7 – Virtual Services** and click **Modify** next to the previously created **S3 HTTPS VIP**
2. **Click Duplicate Service** and confirm when prompted
   
   ![Duplicate Service]

3. Define the **Label** for the new virtual service as required, e.g. `api.cloudian-hyperstore`
4. Set the **Virtual Service IP Address** field to the required IP address, e.g. **192.168.87.67**
5. Set the **Ports** field to **19443**
6. In the Health Checks section click Advanced to expand the menu

7. Set Username to sysadmin

8. Set the Password as appropriate (see the earlier section Health Check Credentials for details)

9. Click Update

### Configuring VIP 5 – Identity and Access Management Service (HTTP)

**Note** This VIP is optional for HyperStore version 7.1.x and earlier but mandatory for version 7.2.x and above.

### Configuring the Virtual Service (VIP)

1. Using the web user interface, navigate to Cluster Configuration > Layer 7 – Virtual Services and click Modify next to the previously created API VIP

2. Click Duplicate Service and confirm when prompted
3. Define the *Label* for the new virtual service as required, e.g. `iam.cloudian-hyperstore`.

4. Set the *Virtual Service IP Address* field to the required IP address, e.g. `192.168.87.67`.

5. Set the *Ports* field to `16080`.

### Layer 7 - Modify Virtual Service

<table>
<thead>
<tr>
<th>Manual Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Label</strong></td>
</tr>
</tbody>
</table>
| `iam.cloudian-hyperstore`  
| **IP Address**        |
| `192.168.87.67`  
| **Ports**             |
| `16080`  

6. Set *Health Checks* to *Connect to port*.

7. Click *Update*.

### Configuring VIP 6 – Identity and Access Management Service (HTTPS)

**Note** This VIP is *optional* for HyperStore version 7.1.x and earlier but *mandatory* for version 7.2.x and above.

### Configuring the Virtual Service (VIP)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click *Modify* next to the previously created IAM VIP.

2. Click *Duplicate Service* and confirm when prompted.

3. Define the *Label* for the new virtual service as required, e.g. `https.iam.cloudian-hyperstore`.

4. Set the *Virtual Service IP Address* field to the required IP address, e.g. `192.168.87.67`.

5. Set the *Ports* field to `16443`.
6. Click **Update**

**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 **Reload HAProxy**

**Testing the Configuration**

The best way to test the load balancer configuration is to pass traffic through the load balanced virtual services.

Ensure that the CMC, S3, API, and IAM services can all be accessed via the load balancer as expected.

**Using System Overview**

The System Overview can be viewed in the WebUI. It shows a graphical view of all VIPs & RIPS (i.e. the HyperStore Nodes) and shows the state/health of each server as well as the state of the each cluster as a whole. The example below shows that all HyperStore nodes are healthy and available to accept connections.

Overview

Direct routing, also known as direct server return or DSR, is a method of load balancing. With direct routing, reply traffic flows directly from the back end servers to the clients. In this way, the load balancer is completely bypassed on the return journey for a given connection, thus removing the load balancer as a potential bottleneck for traffic on the return path.

This alternative method of load balancing can benefit read-intensive deployments which feature a large reply traffic to request traffic ratio.

Consider a scenario where a typical client request is 10 kB in size while a typical reply is 10 GB in size (perhaps file retrieval or video streaming). Direct routing benefits such scenarios: the much larger volume of reply traffic bypasses the load balancer and is not limited by the load balancer’s network throughput. The reply traffic is instead limited by the total available network bandwidth between the servers and the clients, which is limited only by the underlying infrastructure.
Caveats

There are caveats for using the direct routing load balancing method which should be considered:

- The load balancers must be on the same network segment / switching fabric as the HyperStore nodes (due to the fact that this load balancing method works by rewriting MAC addresses, i.e. operates at layer 2 of the OSI model)
- Each HyperStore node must own the VIP address so that they can all accept and reply to the load balanced traffic. This address should be assigned to a loopback network adaptor
- Each HyperStore node must be configured to not reply to ARP requests for the VIP address or advertise that they own the address

The above conditions must be met for layer 4 DR mode to function:

For guidance on configuring the HyperStore nodes for direct routing, in the context of the caveats described above, please consult with Cloudian Sales Engineering or Support.

Appliance Configuration

Configuring VIP 1 – Cloudian Management Console

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. cmc.cloudian-hyperstore
3. Set the Virtual Service IP Address field to the required IP address, e.g. 192.168.87.67
4. Set the Ports field to 8888,8443
5. Leave the Protocol set to TCP
6. Leave the Forwarding Method set to Direct Routing
7. Click Update to create the virtual service
8. Click Modify next to the newly created VIP
9. Ensure that the Persistence Enable checkbox is checked and that the Timeout is set to 1800
10. Set the Health Checks Check Type to Connect to port
11. Set the Check Port to 8443
12. Click Update

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. cloudian-node1
3. Set the Real Server IP Address field to the required IP address, e.g. 10.10.10.11
4. Click Update
5. Repeat these steps to add additional HyperStore nodes as real servers as required

Configuring VIP 2 – S3 Client Requests (HTTP)

Configuring the Virtual Service (VIP)
1. Using the web user interface, navigate to Cluster Configuration > Layer 4 – Virtual Services and click Modify next to the previously created CMC VIP
2. Click Duplicate Service and confirm when prompted
3. Define the Label for the new virtual service as required, e.g. s3.cloudian-hyperstore
4. Set the Virtual Service IP Address field to the required IP address, e.g. 192.168.87.67
5. Set the Ports field to 80
6. Un-check the Persistence Enable checkbox

Note: It is important to verify that the Persistence Enable option has been correctly un-checked. If this step is skipped, the configuration error will propagate throughout the rest of the configuration.
7. Clear the Health Checks Check Port field leaving it empty
8. Click Update
Configuring VIP 3 – S3 Client Requests (HTTPS)

Configuring the Virtual Service (VIP)
1. Using the web user interface, navigate to Cluster Configuration > Layer 4 – Virtual Services and click Modify next to the previously created S3 VIP
2. Click Duplicate Service and confirm when prompted

3. Define the Label for the new virtual service as required, e.g. https.s3.cloudian-hyperstore
4. Set the Virtual Service IP Address field to the required IP address, e.g. 192.168.87.67
5. Set the Ports field to 443
6. Check that Persistence Enable is not checked
   (If this is not the case, delete this VIP and carefully retrace the instructions starting from section Configuring VIP 2 – S3 Client Requests (HTTP), ensuring that all instructions, including the persistence settings, are followed)
7. Click Update

Configuring VIP 4 – Admin API Requests

Configuring the Virtual Service (VIP)
1. Using the web user interface, navigate to Cluster Configuration > Layer 4 – Virtual Services and click Modify next to the previously created S3 HTTPS VIP
2. Click Duplicate Service and confirm when prompted

3. Define the Label for the new virtual service as required, e.g. api.cloudian-hyperstore
4. Set the Virtual Service IP Address field to the required IP address, e.g. 192.168.87.67
5. Set the Ports field to 19443
6. Click Update

Configuring VIP 5 – Identity and Access Management Service (HTTP)

Note: This VIP is optional for HyperStore version 7.1.x and earlier but mandatory for version 7.2.x and above.

Configuring the Virtual Service (VIP)
1. Using the web user interface, navigate to Cluster Configuration > Layer 4 – Virtual Services and click Modify next to the previously created API VIP
2. Click Duplicate Service and confirm when prompted
3. Define the **Label** for the new virtual service as required, e.g. `iam.cloudian-hyperstore`

4. Set the **Virtual Service IP Address** field to the required IP address, e.g. **192.168.87.67**

5. Set the **Ports** field to **16080**

6. Click **Update**

### Configuring VIP 6 – Identity and Access Management Service (HTTPS)

**Note**
This VIP is **optional** for HyperStore version 7.1.x and earlier but **mandatory** for version 7.2.x and above.

#### Configuring the Virtual Service (VIP)

1. Using the web user interface, navigate to **Cluster Configuration > Layer 4 – Virtual Services** and click **Modify** next to the previously created IAM VIP

2. Click **Duplicate Service** and confirm when prompted

3. Define the **Label** for the new virtual service as required, e.g. `https.iam.cloudian-hyperstore`

4. Set the **Virtual Service IP Address** field to the required IP address, e.g. **192.168.87.67**

5. Set the **Ports** field to **16443**

6. Click **Update**

### Testing the Configuration

The best way to test the load balancer configuration is to pass traffic through the load balanced virtual services.

Ensure that the CMC, S3, API, and IAM services can all be accessed via the load balancer as expected.

### Using System Overview

The System Overview can be viewed in the WebUI. It shows a graphical view of all VIPs & RIPs (i.e. the HyperStore Nodes) and shows the state/health of each server as well as the state of the each cluster as a whole. The example below shows that all HyperStore nodes are healthy and available to accept connections.
12. Appliance Configuration for Cloudian HyperStore – Using 'Direct to Node' GSLB

Overview
In the context of a 'GSLB only', 'direct to node' configuration, the function of the load balancer is to ensure that connections to a Cloudian HyperStore cluster are distributed across the HyperStore nodes. This is done to provide a highly available and scalable service. This is achieved by configuring the load balancers to actively health check the HyperStore nodes and serve up the IP address of a healthy node in response to a (delegated) DNS request for the HyperStore service’s domain.

Single Site Deployment Example:
**Explanation:**

- **Start:** A client tries to access the S3 service by using the service's fully qualified domain name, in this example `s3-region1.domain.tld`
- The client sends a DNS query for `s3-region1.domain.tld` to the DNS server
- The DNS server has a CNAME record for `s3-region1.domain.tld` which points to the domain `gslb.domain.tld`
- The DNS server has the domain `gslb.domain.tld` delegated to the load balancers
- The DNS server sends a delegated DNS query for `gslb.domain.tld` to one of the load balancers
- The load balancer that received the delegated DNS query replies to the DNS server by serving up the IP address of a healthy, online HyperStore node. In this example, `10.0.0.11` is the IP address returned by the load balancer
- The DNS server sends the delegated DNS answer to the client
- **Finish:** The client connects to the S3 service at `s3-region1.domain.tld` by using the IP address of the HyperStore node that it was served

**Multi-Site Variant**

The 'direct to node' type deployment can be extended to encompass multiple sites, as needed. The diagram below illustrates such a deployment.

The instructions throughout the remainder of this section clearly illustrate any modifications or additional steps required when setting up a *multi-site* deployment, as opposed to the default single site style of deployment.
Health Checks

The GSLB service, when configured as described in this section, polls each Cloudian HyperStore node at a regular interval to determine its health. This is achieved by sending an HTTPS request to a pre-defined health check location, which is in line with the traditional way of health checking and load balancing a HyperStore deployment.

Handling Multiple Subdomains, Including Wildcard Subdomains

Scenario

A Cloudian HyperStore deployment will typically use the following DNS subdomains (or something similar):

- `cmc.domain.tld`
- `s3-admin.domain.tld`
- `s3-region/location/.domain.tld` (e.g. `s3-region1.domain.tld`)
- `iam.domain.tld`

HyperStore also requires the use of wildcard DNS entries, for example to cover bucket specific subdomains like `app-instance-f57ac0.s3-region1.domain.tld`.

Solution

Configuring DNS delegation can be complex. As such, the supported solution is to:

- Delegate the CMC subdomain to the load balancer (the CMC must be handled separately)

Then, for everything else:

- Delegate a single subdomain to the load balancer, e.g. `gslb`.
- Use CNAME records to point everything else at the delegated subdomain
For example, the subdomain `gslb.domain.tld` would be delegated and everything else would point to it (apart from the CMC subdomain, which is handled separately). This would look like so:

<table>
<thead>
<tr>
<th>Subdomain</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmc.</td>
<td>Delegate to the load balancer</td>
</tr>
<tr>
<td>gslb.</td>
<td>Delegate to the load balancer</td>
</tr>
<tr>
<td>s3-admin.</td>
<td>CNAME to <code>gslb.domain.tld</code></td>
</tr>
<tr>
<td>s3-&lt;region&gt;.</td>
<td>CNAME to <code>gslb.domain.tld</code></td>
</tr>
<tr>
<td>*.s3-&lt;region&gt;.</td>
<td>CNAME to <code>gslb.domain.tld</code></td>
</tr>
<tr>
<td>iam.</td>
<td>CNAME to <code>gslb.domain.tld</code></td>
</tr>
</tbody>
</table>

This approach simplifies DNS entry configuration, particularly when wildcard entries are involved.

**Appliance Configuration**

The GSLB service should be configured on the *master* load balancer appliance (and should be configured at each site if a multi-site deployment is being configured.)

Configuration takes place in the WebUI under *Cluster Configuration > GSLB Configuration*:

**GSLB Configuration**

<table>
<thead>
<tr>
<th>Global Names</th>
<th>Members</th>
<th>Pools</th>
<th>Topologies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>No Data</td>
</tr>
</tbody>
</table>

**Step 1 – Configuring the Global Name**

1. Using the WebUI on the master appliance, navigate to *Cluster Configuration > GSLB Configuration*

2. Select the *Global Names* tab

3. Click the *New Global Name* button

4. Define a friendly *Name* for the new hostname, which can just be the subdomain itself, e.g. `gslb.domain.tld`

5. Define the *Hostname* of what will be the delegated subdomain, e.g. `gslb.domain.tld`

6. Click *Submit*
Step 2 – Configure the Members

Each member is a single HyperStore node.

1. Select the Members tab
2. Click the New Member button
3. Enter a friendly Name for the member, e.g. cloudian-node1
4. Specify an IP address for the member: in this context, this should be the IP address of the HyperStore node in question, e.g. 10.0.0.11
5. Ignore the example value in the Monitor IP field
6. Click Submit
7. Repeat these steps to add additional HyperStore nodes as members as required

(Note: for a multi-site deployment, all nodes from all sites should be added at this stage)
Step 3 – Configure the Pool

A pool must be created to link together a global name with the members that should serve traffic for that global name.

Continuing with the example presented in this section, a pool would be created linking the global name gslb.domain.tld with the members (HyperStore nodes), all of which should serve HyperStore traffic.

1. Select the Pools tab
2. Click the New Pool button
3. Enter a friendly Name for the pool, e.g. hyperstore-nodes
4. Set the Monitor to HTTP
5. Set Monitor Use SSL to Yes
6. Set Monitor Hostname to a hostname that should respond if the HyperStore service is online and healthy, e.g. s3-region1.domain.tld
7. Set Monitor URL Path to /healthCheck
8. Set Monitor Port to 443
9. Set Monitor Expected Codes to 200
10. Set LB Method to wrr

- **Multi-site deployments only:** the LB Method should instead be set to twrr, assuming location affinity is desired (i.e. clients should default to using their local HyperStore nodes)

11. From the Global Names list box, select the global name in question, e.g. gslb.domain.tld
12. In the Members section, drag the appropriate members (HyperStore nodes) from the Available Members box into the Members In Use box
13. Click Submit

Step 4 – Configure the Topology

This step is relevant to multi-site deployments only. For single site deployments, proceed directly to Step 5 – Configure the Separate CMC Service.

Topology configuration is used to map subnets to sites. This gives the solution its location awareness, allowing clients to be directed to a local HyperStore node instead of being bounced between every node at every site, for all nodes that have been defined.

Only skip this step (on a multi-site deployment) if it is not preferred for clients to connect to their local HyperStore nodes by default, i.e. if location affinity is not a requirement. It is assumed that location affinity will be desirable in
1. Select the Topologies tab
2. Click the New Topology button
3. Enter a friendly Name for the topology, e.g. DC1
4. In the IP/CIDR text box, define the subnet(s) that covers the site in question, e.g. 10.0.0.0/24.

   This can be a comma separated list of subnets and hosts, e.g. 10.0.0.0/24, 192.168.2.0/24, 192.168.17.57. The key is that the site's DNS server and the IP addresses of its HyperStore nodes fall within the union of all subnets and hosts defined for the site. This is what allows DNS queries originating from the site to be matched up with that site's local HyperStore nodes: the IP addresses of the local nodes are then served as DNS responses for clients at that site.

5. Click Submit
6. Repeat these steps to add additional topology configurations as required

### Step 5 – Configure the Separate CMC Service

1. Using the WebUI on the master appliance for the first site, navigate to Cluster Configuration > GSLB Configuration
2. Select the Global Names tab
3. Click the New Global Name button
4. Define a friendly Name for the new hostname, which can just be the CMC subdomain itself, e.g. cmc.domain.tld
5. Define the Hostname of what will be the delegated CMC subdomain, e.g. cmc.domain.tld
6. Click Submit
7. Select the Pools tab
8. Click the New Pool button
9. Enter a friendly Name for the pool, e.g. `cmc_hyperstore-nodes`
10. Set the Monitor to HTTP
11. Set Monitor Use SSL to Yes
12. Set Monitor Hostname to a hostname that should respond if the HyperStore service is online and healthy, e.g. `s3-region1.domain.tld`
13. Set Monitor URL Path to `/healthCheck`
14. Set Monitor Port to `443`
15. Set Monitor Expected Codes to `200`
16. Set LB Method to `fogroup`
17. From the Global Names list box, select the global name in question, e.g. `cmc.domain.tld`
18. In the Members section, drag the appropriate members (HyperStore nodes) from the Available Members box into the Members In Use box

**Note**

The order of nodes is important. As this special CMC service uses the failover group load balancing method, HyperStore nodes will be prioritised to receive traffic in the order in which they are listed in the Members In Use box. The first node listed will always receive all CMC traffic. If the first node is offline then the second node in the list will receive all CMC traffic; if the first and second nodes are both offline then the third node will receive all CMC traffic, and so on for all nodes in the list.

19. Click Submit
Step 6 – Finalising the Configuration
To apply the new settings, the GSLB service must be restarted as follows:

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

DNS Server Configuration
Once the GSLB service has been configured on the master load balancer at every site, the DNS server at each site must then be configured for GSLB.

The DNS server at each site must be configured to delegate DNS requests for the subdomain in question to the load balancers; the load balancers' GSLB services will serve the appropriate IP addresses to the DNS servers. Using the example presented throughout this appendix, the DNS server at each site would be configured with a
delegation for the domain `s3-region1.domain.tld`. The domain would be delegated to every load balancer across every site, which provides multi-site redundancy.

Steps walking through creating a DNS delegation on a Microsoft DNS server in the context of setting up GSLB on our appliance can be found in the appendix, in the section **DNS Server Configuration for Cloudian HyperStore**.

### Testing the Configuration

The configuration can be tested to make sure it’s working as expected.

From the command line on a Microsoft Windows machine, the `nslookup` program can be used to send test DNS queries to the load balancer(s). The master load balancer is located at IP address 10.0.0.1 in the example presented here.

For the test, use the `-norecurse` option to instruct the load balancer **not** to attempt to query another server for the answer. A successful test would see the load balancer respond with the IP address of one of the online HyperStore nodes, like so:

```
C:\Users\me>nslookup -norecurse s3-region1.domain.tld 10.0.0.1
Server:  UnKnown
Address:  10.0.0.1
Name:  s3-region1.domain.tld
Address:  10.0.0.11
```

### 13. Appliance Configuration for Cloudian HyperStore – Using Multi-Site GSLB and Location Affinity

#### Conceptual Overview

For **multi-site HyperStore deployments**, it is possible to use the load balancer's GSLB functionality to provide both high availability and location affinity across multiple sites.

- **Clients across multiple sites** use the same fully qualified domain name to access HyperStore services
- **Under normal operation**: clients are directed to their local site’s HyperStore cluster
- **In the event of a local service failure**: clients are automatically directed to a functioning HyperStore cluster at another site. This would happen if the local site’s HyperStore cluster and/or load balancers were offline and unavailable

For the sake of simplicity, the diagram presented below shows a two site setup. The principle can be extended to encompass as many sites as desired.
### Explanation:

- **Start**: A client accesses the S3 service by using the service’s fully qualified domain name, in this example `s3-region1.domain.tld`.

- The client sends a DNS query for `s3-region1.domain.tld` to its local DNS server.

- The local site’s DNS server has a CNAME record for `s3-region1.domain.tld` which points to the domain `gslb.domain.tld`.

- The DNS server has the domain `gslb.domain.tld` delegated to the load balancers.

- The DNS server sends a delegated DNS query for `gslb.domain.tld` to one of the load balancers.

- The load balancer that received the delegated DNS query replies to the DNS server. The load balancer answers with the IP address of the VIP (HyperStore service) that is local to the DNS server making the query, and hence local to the original client.

  - An example: if the delegated query from the server originated from the 10.0.0.0/24 subnet then the VIP in that subnet is served up. Likewise, if the delegated query originated from the 172.16.0.0/24 subnet then the VIP in that subnet is served up. As such, clients are always directed to their local, on-site HyperStore instance, provided that the local instance is online and available.

- The DNS server sends the delegated DNS answer to the client.

- **Finish**: The client connects to the S3 service at `s3-region1.domain.tld` by using the local VIP address.
In the event that the HyperStore cluster and/or load balancers at one site should completely fail then local clients will be directed to the HyperStore cluster at the other site and the service will continue to be available.

This style of multi-site failover is possible because the load balancers' GSLB functionality continuously health checks the service at each site. When the service at a site is observed to be unavailable then that site’s IP address is no longer served when responding to DNS queries.

DNS Server Prerequisites

**Important**: location affinity (ensuring clients ‘stick’ to their local site) requires a unique DNS server at each site.

For this setup to work and provide location affinity, a unique DNS server is required at each site, like the example deployment shown at the beginning of this section.

If multiple sites share a common DNS server then clients cannot be directed to their local, on-site HyperStore instance.

Example: Consider a two data centre deployment with a shared, common DNS server located at DC 1. From the perspective of a load balancer in this scenario, every delegated DNS request would be seen to come from the single, shared DNS server at DC 1. Specifically, the requests would all come from the DNS server’s IP address, which would fall within DC 1’s subnet.

A load balancer would have no way to distinguish between delegated requests for DC 1’s clients and delegated requests for DC 2’s clients. All delegated requests would originate from within DC 1’s subnet, therefore all traffic would be directed to DC 1’s HyperStore instance.

To resolve such a situation, a DNS server would need to be deployed at DC 2. The load balancers could then easily tell which site a given delegated DNS query has come from and, therefore, which site the client should be directed to.

If having unique DNS servers per-site and splitting up sites using a topology configuration is not possible then clients will bounce between different VIPs (and hence bounce between sites sites) in a round-robin fashion. If this behaviour is acceptable then it can theoretically be used without significant issue, provided that a failover group is used for the CMC service (see the following section for more information.)

**Alternative Deployment Without Location Affinity**

The CMC service requires clients to stick to a single HyperStore instance (a single VIP in this scenario) for the entire duration of their session. If CMC connections bounce between different instances then the CMC service will break.

If deploying a multi-site setup without location affinity (not recommended) then a separate GSLB configuration must be written for the CMC service. This makes use of ‘failover groups’ to ensure that clients stick to the same HyperStore instance when using the CMC service.

To set up a deployment without location affinity, follow the standard instructions presented later in this section, taking note of the special instructions that are flagged as being necessary for this alternative deployment type.

**Handling Multiple Subdomains, Including Wildcard Subdomains**
Scenario
A Cloudian HyperStore deployment will typically use the following DNS subdomains (or something similar):

- cmc.domain.tld
- s3-admin.domain.tld
- s3-<region/location>.domain.tld (e.g. s3-region1.domain.tld)
- iam.domain.tld

HyperStore also requires the use of wildcard DNS entries, for example to cover bucket specific subdomains like app-instance-f57ac0.s3-region1.domain.tld.

Solution
Configuring DNS delegation can be complex. As such, the supported solution is to:

- Delegate the CMC subdomain to the load balancer (the CMC must be handled separately)

Then, for everything else:

- Delegate a single subdomain to the load balancer, e.g. gslb.
- Use CNAME records to point everything else at the delegated subdomain

For example, the subdomain gslb.domain.tld would be delegated and everything else would point to it (apart from the CMC subdomain, which is handled separately). This would look like so:

<table>
<thead>
<tr>
<th>Subdomain</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmc</td>
<td>Delegate to the load balancer</td>
</tr>
<tr>
<td>gslb</td>
<td>Delegate to the load balancer</td>
</tr>
<tr>
<td>s3-admin</td>
<td>CNAME to gslb.domain.tld</td>
</tr>
<tr>
<td>s3-&lt;region&gt;</td>
<td>CNAME to gslb.domain.tld</td>
</tr>
<tr>
<td>*.s3-&lt;region&gt;</td>
<td>CNAME to gslb.domain.tld</td>
</tr>
<tr>
<td>iam</td>
<td>CNAME to gslb.domain.tld</td>
</tr>
</tbody>
</table>

This approach simplifies DNS entry configuration, particularly when wildcard entries are involved.

Appliance Configuration
The GSLB service should be configured on the master load balancer appliance at each site.

Note that the GSLB configuration must be identical across all sites: inconsistent configurations will lead to unexpected behaviour.

Configuration takes place in the WebUI under Cluster Configuration > GSLB Configuration:
Step 1 – Configuring the Global Name

1. Using the WebUI on the master appliance for the first site, navigate to Cluster Configuration > GSLB Configuration

2. Select the Global Names tab

3. Click the New Global Name button

4. Define a friendly Name for the new hostname, which can just be the subdomain itself, e.g. gslb.domain.tld

5. Define the Hostname of what will be the delegated subdomain, e.g. gslb.domain.tld

6. Click Submit

Step 2 – Configure the Members

Each member can be thought of as a single site.

1. Select the Members tab

2. Click the New Member button

3. Enter a friendly Name for the member, e.g. DC1

4. Specify an IP address for the member: in this context, this should be the VIP address of the site’s HyperStore
service, e.g. 10.0.0.2

5. Ignore the example value in the Monitor IP field

6. Click Submit

7. Repeat these steps to add additional sites as members as required

---

Step 3 – Configure the Pool

A pool must be created to link together a global name with the members that should serve traffic for that global name.

Continuing with the example presented in this section, both sites have a functional HyperStore cluster ready for use. A pool would therefore be created linking the global name gslb.domain.tld with members (sites) DC1 and DC2, both of which should serve HyperStore traffic.

1. Select the Pools tab

2. Click the New Pool button

3. Enter a friendly Name for the pool, e.g. hyperstore-sites

4. Set the Monitor to HTTP

5. Set Monitor Use SSL to Yes

6. Set Monitor Hostname to a hostname that should respond if the HyperStore service is online and healthy, e.g. s3-region1.domain.tld

7. Set Monitor URL Path to /.site_health

8. Set Monitor Port to 50080

9. Set Monitor Expected Codes to 200
10. Set LB Method to twrr
   - Alternative Deployment Without Location Affinity **only**: the LB Method must instead be set to wrr

11. From the Global Names list box, select the global name in question, e.g. gslb.domain.tld

12. In the Members section, drag the appropriate members (sites) from the Available Members box into the Members In Use box

13. Click Submit

**Step 4 – Configure the Topology**

Topology configuration is used to map subnets to sites. This gives the solution its location awareness, allowing clients to be directed to their local HyperStore instance instead of being bounced between every site which has
been defined.

1. **Alternative Deployment Without Location Affinity only**: this step (Configure the Topology) must be skipped. Proceed to **Step 6 – Finalising the Configuration**

2. Select the **Topologies** tab

3. Click the **New Topology** button

4. Enter a friendly **Name** for the topology, e.g. **DC1**

5. In the **IP/CIDR** text box, define the subnet(s) that covers the site in question, e.g. **10.0.0.0/24**.

   This can be a comma separated list of subnets and hosts, e.g. **10.0.0.0/24, 192.168.2.0/24, 192.168.17.57**. The key is that the site's DNS server **and** its HyperStore VIP fall within the union of all subnets and hosts defined for the site. This is what allows DNS queries originating from the site to be matched up with that site’s local VIP: the local VIP is then served as a DNS response for clients at that site

6. Click **Submit**

7. Repeat these steps to add additional topology configurations as required

---

**GSLB Configuration**

<table>
<thead>
<tr>
<th>Global Names</th>
<th>Members</th>
<th>Pools</th>
<th>Topologies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### New Topology

<table>
<thead>
<tr>
<th>Name</th>
<th>DC1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP/CIDR</td>
<td>10.0.0.0/24</td>
</tr>
</tbody>
</table>

**Submit** | **Cancel**

---

**Step 5 – Configure the Separate CMC Service (Alternative Deployment Without Location Affinity only)**

This step **only applies** when using the **Alternative Deployment Without Location Affinity** and must be skipped if performing a standard installation.

1. Using the WebUI on the master appliance for the first site, navigate to **Cluster Configuration > GSLB Configuration**

2. Select the **Global Names** tab

3. Click the **New Global Name** button
4. Define a friendly Name for the new hostname, which can just be the CMC subdomain itself, e.g. cmc.domain.tld

5. Define the Hostname of what will be the delegated CMC subdomain, e.g. cmc.domain.tld

6. Click Submit

GSLB Configuration

New_global_name

<table>
<thead>
<tr>
<th>Name</th>
<th>cmc.domain.tld</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
<td>cmc.domain.tld</td>
</tr>
<tr>
<td>TTL</td>
<td>30 seconds</td>
</tr>
</tbody>
</table>

Submit Cancel

7. Select the Pools tab

8. Click the New Pool button

9. Enter a friendly Name for the pool, e.g. cmc_hyperstore-sites

10. Set the Monitor to HTTP

11. Set Monitor Use SSL to Yes

12. Set Monitor Hostname to a hostname that should respond if the HyperStore service is online and healthy, e.g. s3-region1.domain.tld

13. Set Monitor URL Path to /.site_health

14. Set Monitor Port to 50080

15. Set Monitor Expected Codes to 200

16. Set LB Method to fogroup

17. From the Global Names list box, select the global name in question, e.g. cmc.domain.tld

18. In the Members section, drag the appropriate members (sites) from the Available Members box into the Members In Use box
The order of sites is important. As this special CMC service uses the failover group load balancing method, sites will be prioritised to receive traffic in the order in which they are listed in the Members In Use box. The first site listed will always receive all CMC traffic. If the first site is offline then the second site in the list will receive all CMC traffic; if the first and second sites are both offline then the third site will receive all CMC traffic, and so on for all sites in the list.

19. Click Submit

Step 6 – Finalising the Configuration

To apply the new settings, the GSLB service must be restarted as follows:
1. Using the WebUI, navigate to: Maintenance > Restart Services and click Restart GSLB

Step 7 – Intelligent Site Health Check

On the master load balancer on each site, it is necessary to configure intelligent health checking. This allows an entire site to be marked as 'offline' if any single HyperStore service should become unavailable. For example, if the API service is no longer available at a given site but all other services are healthy then that site will be marked as offline, regardless: all services must be healthy and online to be able to guarantee consistent service for end users.

To configure the intelligent site health check, in the WebUI navigate to Cluster Configuration > Layer 7 – Manual Configuration. Clear out the default text and replace it with the following:

```plaintext
frontend gslb_site_health_check
  bind *:50080
  mode http
  acl site_dead nbsrv(cmc.cloudian-hyperstore) lt 3
  acl site_dead nbsrv(s3.cloudian-hyperstore) lt 3
  acl site_dead nbsrv(https.s3.cloudian-hyperstore) lt 3
  acl site_dead nbsrv(api.cloudian-hyperstore) lt 3
  monitor-uri /.site_health
  monitor fail if site_dead
```

The 3 in `lt 3` should be set as appropriate for the deployment in question. This number represents the minimum number of HyperStore nodes needed to provide full (read and write) access to all services. In this specific example, having three HyperStore nodes healthy and online is sufficient to provide users with full access to all services. If only two nodes are online then services become degraded or unavailable. As such, a site that falls to less than three online nodes is marked as 'down' and will no longer be served as a DNS answer to future client queries.

**Note**

The minimum viable node count may be linked to deployment specific settings, such as EC and replication settings. For guidance on what this number is for a specific deployment, please consult with Cloudian Sales Engineering or Support.

Note that if the Cloudian virtual services were defined using different names to the examples presented in this document then the names in use should be reflected in the intelligent health check configuration. For example, if the CMC virtual service was named "my-cmc-cloudian-service" then that should replace "cmc.cloudian-hyperstore" in the health check text.

The web interface will display an error if any spelling mistakes have been made in the names of the virtual services.

The final configuration should look like the following:

```
frontend gslb_site_health_check
  bind *:50080
  mode http
  acl site_dead nbsrv(cmc.cloudian-hyperstore) lt 3
  acl site_dead nbsrv(s3.cloudian-hyperstore) lt 3
  acl site_dead nbsrv(https.s3.cloudian-hyperstore) lt 3
  acl site_dead nbsrv(api.cloudian-hyperstore) lt 3
  monitor-uri /.site_health
  monitor fail if site_dead
```
Optional: Defining a Default Site for External Traffic (Handling DNS Requests from Unpredictable Source Addresses)

It is plausible that a HyperStore GSLB deployment may be required to answer DNS queries sourced from outside of the subnets defined in the topology configuration.

Consider a client on the public Internet requesting a resource from the HyperStore cluster. The DNS query associated with the request may be sourced from a previously unseen, unpredictable public IP address. DNS queries from IP addresses that do not fall within the predefined network topology/subnets will be answered with DNS records pointing to any of the defined sites in a round-robin fashion.

An alternative is to define a default site. All DNS queries from outside the predefined network topology will be answered with the same DNS record: a record pointing to the default site.

To configure this, add the widest possible subnet of 0.0.0.0/0 to the topology configuration of the site which is to be the 'default'. Any DNS query whose source IP address does not fall within one of the other, smaller subnets will be picked up by this new "catch all" subnet.

Following on from the previous example, setting data centre 1 to be the 'default' site would look like so:

**GSLB Configuration**

Once the GSLB service has been configured on the master load balancer at every site, the DNS server at each site must then be configured for GSLB.

The DNS server at each site must be configured to delegate DNS requests for the subdomain in question to the load balancers; the load balancers' GSLB services will serve the appropriate IP addresses to the DNS servers.

Using the example presented throughout this appendix, the DNS server at each site would be configured with a delegation for the domain s3-region1.domain.tld. The domain would be delegated to every load balancer across every site, which provides multi-site redundancy.
Steps walking through creating a DNS delegation on a Microsoft DNS server in the context of setting up GSLB on our appliance can be found in the appendix, in the section **DNS Server Configuration for Cloudian HyperStore**.

14. Testing & Verification
Appropriate steps for testing a load balanced Cloudian HyperStore deployment vary by deployment type. Refer to the end of the section dedicated to a specific deployment type for instructions on how to test and verify the configuration.

15. 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

16. Further Documentation
The Administration Manual contains much more information about configuring and deploying the appliance. It's available here: [https://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf](https://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf)

17. Conclusion
Loadbalancer.org appliances provide a very cost effective solution for highly available load balanced Cloudian HyperStore environments.
18. Appendix

DNS Server Configuration for Cloudian HyperStore

Once the GSLB service has been fully configured and applied on the load balancers, as described in the previous sections, the DNS server must be configured for GSLB.

The DNS server must be configured to delegate DNS requests for the subdomain in question to the load balancers; the load balancers' GSLB services will serve the appropriate IP addresses to the DNS server. Using the example presented throughout this document, the DNS server would be configured with a delegation for the domain s3-region1.domain.tld. The domain would be delegated to both load balancers, if using an HA pair, to provide redundancy.

The exact steps for creating a DNS delegation vary between different DNS servers. Presented below are steps that walk through creating a DNS delegation on a Microsoft DNS server in the context of setting up GSLB on our appliance.

Microsoft DNS Server

Delegating a subdomain in Microsoft DNS Manager is a short process.

1. Open DNS Manager and create A records for each load balancer, using Action > New Host (e.g. lbmaster.domain.tld and lbslave.domain.tld for an HA pair of load balancers)

2. Provided that the load balancer part of the GSLB configuration has been completed and is working, as described in Section 10, “Appliance Configuration for Cloudian HyperStore – Using Layer 7 SNAT Mode”, the New Delegation wizard should now be used to delegate the subdomain to the load balancers. The delegation will use the new FQDNs for the load balancers, as defined in the previous step. The delegation wizard is located at Action > New Delegation
3. Test the delegation to make sure it is working as expected.

From the Windows command line, the `nslookup` program can be used to send test DNS queries to the DNS server. The DNS server is located at IP address 10.0.0.50 in the example presented here.

For the first test, use the `-norecurse` option to instruct the DNS server not to query another server for the answer. A successful test would see the DNS server respond and indicate that the subdomain in question is served by another server(s), giving the other server’s details, like so:

```
C:\Users\me>nslookup -norecurse s3-region1.domain.tld 10.0.0.50
Server: UnKnown
Address: 10.0.0.50

Name:   s3-region1.domain.tld
Served by:
 - lbmaster.domain.tld
   10.0.0.1
   s3-region1.domain.tld
 - lbslave.domain.tld
   10.0.0.2
   s3-region1.domain.tld
```

For the second test, execute the same command without the `-norecurse` option. This should see the DNS server fetch the answer from the load balancer and then serve up the ‘fetched’ answer in its response. A successful test would see the server reply with the IP address of one of the online HyperStore nodes, like so:

```
C:\Users\me>nslookup s3-region1.domain.tld 10.0.0.50
Server: UnKnown
Address: 10.0.0.50

Name:   s3-region1.domain.tld
Served by:
 - lbmaster.domain.tld
   10.0.0.1
   s3-region1.domain.tld
 - lbslave.domain.tld
   10.0.0.2
   s3-region1.domain.tld
```
Using the PROXY Protocol to Retain Client IP Addresses

The PROXY protocol is an HTTP-like protocol which allows a reverse proxy, such as a layer 7 load balancer, to pass along the original client's source IP address. The PROXY protocol is not HTTP compliant and cannot be read by applications that are expecting to receive standard HTTP requests.

HyperStore 7.0 and later supports reading traffic that uses the PROXY protocol. When correctly configured, this makes the HyperStore nodes aware of the original client's source IP address for S3 requests. This allows for the use of S3 bucket policies and the HyperStore billing whitelist feature, both of which require clients' IP addresses to function. This also means that clients' real IP addresses are recorded in the S3 request log, which can assist with troubleshooting and monitoring.

Configuring the HyperStore Nodes to Accept the PROXY Protocol

1. Log on to the Puppet master node (the HyperStore node on which the installation script was run)
2. In common.csv (the main HyperStore configuration file), set the value of s3_proxy_protocol_enabled to true

Step 2 can be carried out using the vi text editor. The common.csv file to be edited is located at:
/etc/cloudian-<version>-puppet/manifests/extdata/common.csv

The line
s3_proxy_protocol_enabled,false
should be amended to read
s3_proxy_protocol_enabled,true
and the file should then be saved.

Alternatively, executing the following series of commands from the console will make the necessary change:

```
find /etc -iname "common.csv" | xargs -n1 sed -i.bkup "s/s3_proxy_protocol_enabled,false/s3_proxy_protocol_enabled,true/g"
```

3. Push the configuration change to all HyperStore nodes in the cluster

The easiest way to carry out step 3 is to change into the installation staging directory at the command line and then run the HyperStore installation/configuration script. To run the script, execute:

```
./cloudianInstall.sh
```

Using the script, enter 2 for Cluster Management:
Enter b for *Push Configuration Settings to Cluster*:

```
Cluster Management

a ) Review Cluster Configuration  
b ) Push Configuration Settings to Cluster [OK]  
c ) Manage Services  
d ) Run Validation Tests  
x ) Return to Main Menu

Choice: b
```

Press **Enter** at the prompt to select all nodes, and then wait for a success message to be displayed:

```
Run Puppet to configure agent nodes
message reads: contains the following hosts: cloudian1, cloudian2, cloudian3.
Receive a comma-separated list of hosts in order to execute agents on (empty for all) !!!!
Sending to /var/run/puppet/puppetserver-service
Restarting to /var/run/puppet/puppetserver-service

Configuring agent node cloudian1.
Ready to run Puppet agent on host cloudian1. This could take some time.
Configuring agent node cloudian2.
Configuring agent node cloudian3.
Ready to run Puppet agent on host cloudian2. This could take some time.
Ready to run Puppet agent on host cloudian3. This could take some time.
All Puppet agents run completed successfully in office region.
Puppet agent daemon is now running on cloudian1.
Puppet agent daemon is now running on cloudian2.
Puppet agent daemon is now running on cloudian3.
Puppet agent run ended for office.
Press any key to continue ... []
```

4. Restart the S3 service to apply the new configuration across all nodes
From the *Cluster Management* menu of the installation/configuration script, enter `c` for *Manage Services*, enter `5` for S3 service, enter "restart" to trigger a cluster-wide restart of the service, and then wait for each success message to be displayed:

```plaintext
<table>
<thead>
<tr>
<th>Service Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ) All services</td>
</tr>
<tr>
<td>1 ) Redis Credentials</td>
</tr>
<tr>
<td>2 ) Redis 00S</td>
</tr>
<tr>
<td>3 ) Cassandra</td>
</tr>
<tr>
<td>4 ) HyperStore service</td>
</tr>
<tr>
<td>5 ) S3 service</td>
</tr>
<tr>
<td>6 ) Redis Monitor</td>
</tr>
<tr>
<td>7 ) Cloudian Agent</td>
</tr>
<tr>
<td>8 ) Consul</td>
</tr>
<tr>
<td>9 ) Cloudian Management Console (CMC)</td>
</tr>
<tr>
<td>X ) Quit</td>
</tr>
</tbody>
</table>
```

You can execute the following list of commands:
```
start,stop,status,restart,version,force-stop,node-start,node-stop
```

Select a service to manage: 5
Enter command: (start,stop,status,restart,version) restart
Executing Cloudian S3 service command restart ...

```
On host Cloudian1:
  /etc/init.d/cloudian-s3 restart => Restarting cloudian-s3 (via systemctl): [ OK ]

On host Cloudian2:
  /etc/init.d/cloudian-s3 restart => Restarting cloudian-s3 (via systemctl): [ OK ]

On host Cloudian3:
  /etc/init.d/cloudian-s3 restart => Restarting cloudian-s3 (via systemctl): [ OK ]

Press any key to continue ...
```

### Appliance Configuration

#### Configuring Additional VIP 1 – S3 Client Requests Using the PROXY Protocol (HTTP)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click **Modify** next to the existing S3 VIP, e.g. `s3.cloudian-hyperstore`
2. Click **Duplicate Service** and confirm when prompted
3. Define the **Label** for the new virtual service as required, e.g. `pp_s3.cloudian-hyperstore`
4. Set the **Ports** field to `81`
5. In the **Other** section click **Advanced** to expand the menu
6. Set Send Proxy Protocol to **Send Proxy V1**
7. Click **Update**

#### Configuring Additional VIP 2 – S3 Client Requests Using the PROXY Protocol (HTTPS)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click **Modify** next to the existing S3 HTTPS VIP, e.g. `https.s3.cloudian-hyperstore`
2. Click **Duplicate Service** and confirm when prompted
3. Define the Label for the new virtual service as required, e.g. pp_https.s3.cloudian-hyperstore
4. Set the Ports field to 4431
5. In the Other section click Advanced to expand the menu
6. Set Send Proxy Protocol to Send Proxy V1
7. Click Update

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 Reload HAProxy

The two new virtual services will now be ready for use and will send traffic to the HyperStore nodes using the PROXY protocol.

Configuring HA - Adding a Slave 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 master appliance is fully configured first, then the slave should be added. Once the master and slave are paired, all load balanced services configured on the master are automatically replicated to the slave over the network using SSH/SCP.

Note
For Enterprise Azure, the HA pair should be configured first. In Azure, when creating a VIP using an HA pair, 2 private IPs must be specified – one for the VIP when it’s active on the Master and one for the VIP when it’s active on the Slave. Configuring the HA pair first, enables both IPs to be specified when the VIP is created.

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

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 by WebUI menu option in the table below:

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

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

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

![Create a Clustered Pair](image)

3. Specify the IP address and the `loadbalancer` user’s password for the slave (peer) appliance as shown above
4. Click Add new node
5. The pairing process now commences as shown below:
6. Once complete, the following will be displayed:

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

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

Note: For more details on configuring HA with 2 appliances, please refer to Chapter 9 in the **Administration Manual**.
## 19. 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>4 April 2018</td>
<td>Initial version</td>
<td></td>
<td>AH</td>
</tr>
<tr>
<td>1.1.0</td>
<td>7 November 2018</td>
<td>Changed the &quot;Health Checks&quot; section to talk about the new built-in health checks and the rewritten custom health check</td>
<td>Required updates</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removed the two custom health checks from the appendix and replaced them with the updated new single script, along with a link to download it online</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changed the VIP configuration instructions to refer to the new built-in health checks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.1</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>1.1.2</td>
<td>18 December 2018</td>
<td>Modified the 'Loadbalancer.org Appliances Supported' section to state that only the Enterprise 10G, Enterprise 40G, and Enterprise VA MAX models are supported</td>
<td>Required updates</td>
<td>AH</td>
</tr>
<tr>
<td>1.1.3</td>
<td>31 January 2019</td>
<td>Added a clarifying note to the section on creating the CMC virtual service, explicitly stating that it requires the default setting of source IP persistence to be left enabled</td>
<td>Required updates</td>
<td>AH</td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Change</td>
<td>Reason for Change</td>
<td>Changed By</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>1.2</td>
<td>3 April 2019</td>
<td>Changed Loadbalancer.org software versions supported to V8.3.6 and later, to account for the new authenticated health check in the WebUI</td>
<td>Required updates</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Replaced the explanatory note regarding persistence on the CMC VIP with an explicit instruction to set the persistence option to HTTP cookie</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added instructions for setting up a negotiate health check for the CMC VIP, and added a new explanatory note regarding the choice of port used</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added credentials for the health check for the API VIP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changed the &quot;Health Checks&quot; section to reflect the changes to health checks</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removed the section referencing the custom health check</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removed the appendix containing the custom health check</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added a section on &quot;Performance and Sizing&quot; explaining the need for virtual appliances to be assigned a minimum of 4 vCPUs</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added instructions for configuring and enabling multithreaded HAProxy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2.1</td>
<td>4 April 2019</td>
<td>Corrected the persistence mode for the CMC VIP to source IP, as it is a 'TCP mode' VIP</td>
<td>Required updates</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removed the note for the CMC VIP that explained the rationale for checking against the TLS port, for simplicity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Change</td>
<td>Reason for Change</td>
<td>Changed By</td>
</tr>
<tr>
<td>---------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>1.2.2</td>
<td>25 April 2019</td>
<td>Added additional screenshots regarding the health checks, for clarity</td>
<td>Required updates</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added notes regarding setting the health check host header if needed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2.3</td>
<td>4 June 2019</td>
<td>Changed the health check for the Cloudian Management Console VIP, based on updated documentation from Cloudian</td>
<td>Required updates</td>
<td>AH</td>
</tr>
<tr>
<td>1.2.4</td>
<td>5 July 2019</td>
<td>Changed the health check 'Request to send' fields from .healthCheck to /.healthCheck, and updated screen shots accordingly</td>
<td>Required updates</td>
<td>AH</td>
</tr>
<tr>
<td>1.3.0</td>
<td>27 August 2019</td>
<td>Styling and layout</td>
<td>General styling updates</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changed the health check for the Cloudian Management Console VIP to use an OPTIONS check</td>
<td>Required updates</td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Change</td>
<td>Reason for Change</td>
<td>Changed By</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| 1.4.0   | 31 October 2019 | Rewrote the instructions for setting up the virtual services to make use of the new ‘Duplicate Service’ function

Moved the health check specific screenshots and note blocks above the ‘Click Update’ instructions for clarity

Added paragraph ‘GSLB / Location Affinity’ and associated appendix with configuration instructions

Added paragraph ‘Alternative Load Balancing Method for Read-Intensive Deployments (Direct Routing)’ and associated appendix with configuration instructions

Added a note at the end of the S3 VIP setup instructions pointing to a new appendix, ‘Using the PROXY Protocol to Retain Client IP Addresses’, containing configuration instructions

Changed the Loadbalancer.org software version supported to 8.4.1 due to now needing health checks using the PROXY protocol

Updated the advice for configuring multithreading in HAProxy to recommend a minimum of 4 threads and a maximum of the total number of available threads                                                                 | Required updates Added support for the PROXY protocol at the request of Cloudian Added additional deployment options |           |
| 1.4.1   | 15 October 2020 | Added instructions for setting up the new IAM VIPs

Updated diagrams to reflect additional IAM VIPs                                                                 | Required updates requested by Cloudian | AH         |
<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>1.5.0</td>
<td>No public release</td>
<td>Minor visual updates to multi-site GSLB diagram</td>
<td>Required technical changes</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Important content changes and additions regarding how GSLB is configured and deployed</td>
<td>Branding update</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>New title page</td>
<td>Change to Canadian contact details</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Canadian contact details</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0.0</td>
<td>15 January 2021</td>
<td>Complete overhaul of the document</td>
<td>Required technical changes</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Document restructured with several sections added and some removed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Direct to node' deployment type added from a previously separate internal document</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>New advice added regarding the minimum viable node count for the 'Intelligent Site Health Check'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0.1</td>
<td>29 January 2021</td>
<td>Corrected screenshot for IAMS virtual service</td>
<td>Required correction</td>
<td>AH</td>
</tr>
<tr>
<td>2.0.2</td>
<td>11 February 2021</td>
<td>Added admin API credential information</td>
<td>Required updates requested by Cloudian</td>
<td>AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added warnings to prevent 'duplicate service' related configuration errors from propagating throughout an entire deployment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.0</td>
<td>29 April 2021</td>
<td>Converted the document to AsciiDoc</td>
<td>Move to new documentation system</td>
<td>AH</td>
</tr>
<tr>
<td>2.1.1</td>
<td>5 July 2021</td>
<td>Amended the ‘Handling Multiple Subdomains’ sections and added an additional example</td>
<td>Technical content improvement and clarification</td>
<td>AH</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.