LOADBALANCER

DEPLOYMENT GUIDE

Load Balancing Cloudian HyperStore

Version 2.3.2

Table of Contents

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/enterprise.

3. Software Versions Supported

3.1. Loadbalancer.org Appliance

- V8.9.1 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. Cloudian HyperStore

- All versions

4. 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 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.

5. Load Balancing Cloudian HyperStore

ℹ 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 7, "Deployment Concept" describes each of the different **specific** load balancing methods.

5.1. 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.

5.2. 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.

5.3. 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

5.4. Port Requirements

The following table shows the ports that are load balanced:

Port	Protocols	Use
80	TCP/HTTP	Requests from S3 client applications
443	TCP/HTTPS	Requests from S3 client applications
8443	TCP/HTTPS	Requests from clients
8888	TCP/HTTP	Requests from clients
16080	TCP/HTTP	IAM service traffic
16443	TCP/HTTPS	IAM service traffic
19443	TCP/HTTPS	Admin API requests

5.5. Health Checks

The S3-HTTP and IAM-HTTP services use the "Negotiate HTTP (HEAD)" health check, while the S3-HTTPS, IAM-HTTPS, and API services 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 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.

6. 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.

7. Deployment Concept

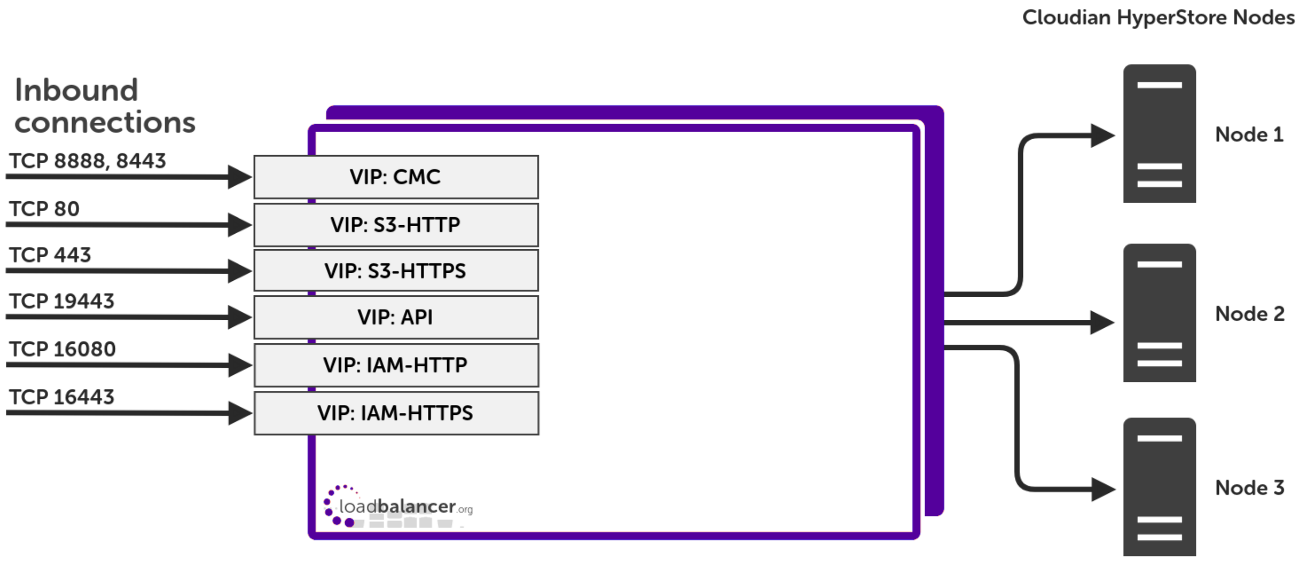
Cloudian HyperStore can be load balanced in a variety of different ways. The different deployment types are described below.

Deployment Types Overview and Quick Links:

- Layer 7 SNAT Mode (Default)
- 'Direct to Node' GSLB
- Multi-Site GSLB and Location Affinity

7.1. 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



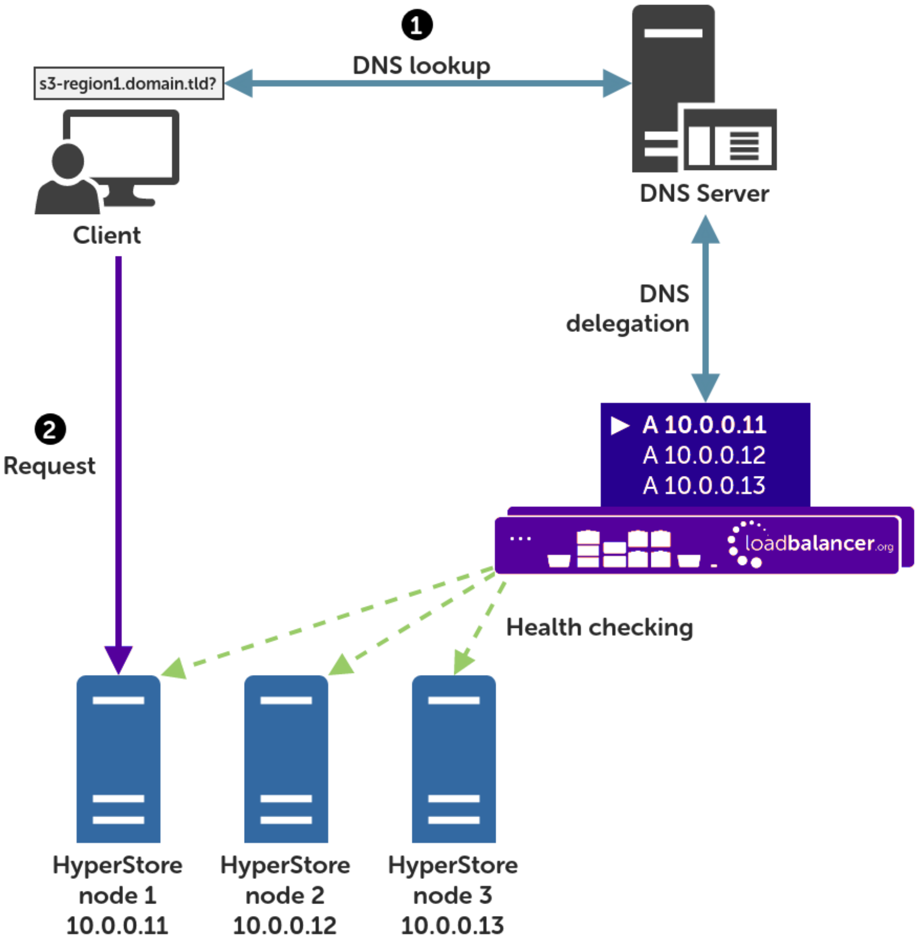
Note	The load balancer can be deployed as a single unit, although Loadbalancer.org recommends a clustered pair for resilience & high availability. Please refer to the section Configuring HA - Adding a Secondary Appliance in the appendix for more details on configuring a clustered pair.

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 9, “Appliance Configuration for Cloudian HyperStore – Using Layer 7 SNAT Mode”.

7.2. '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



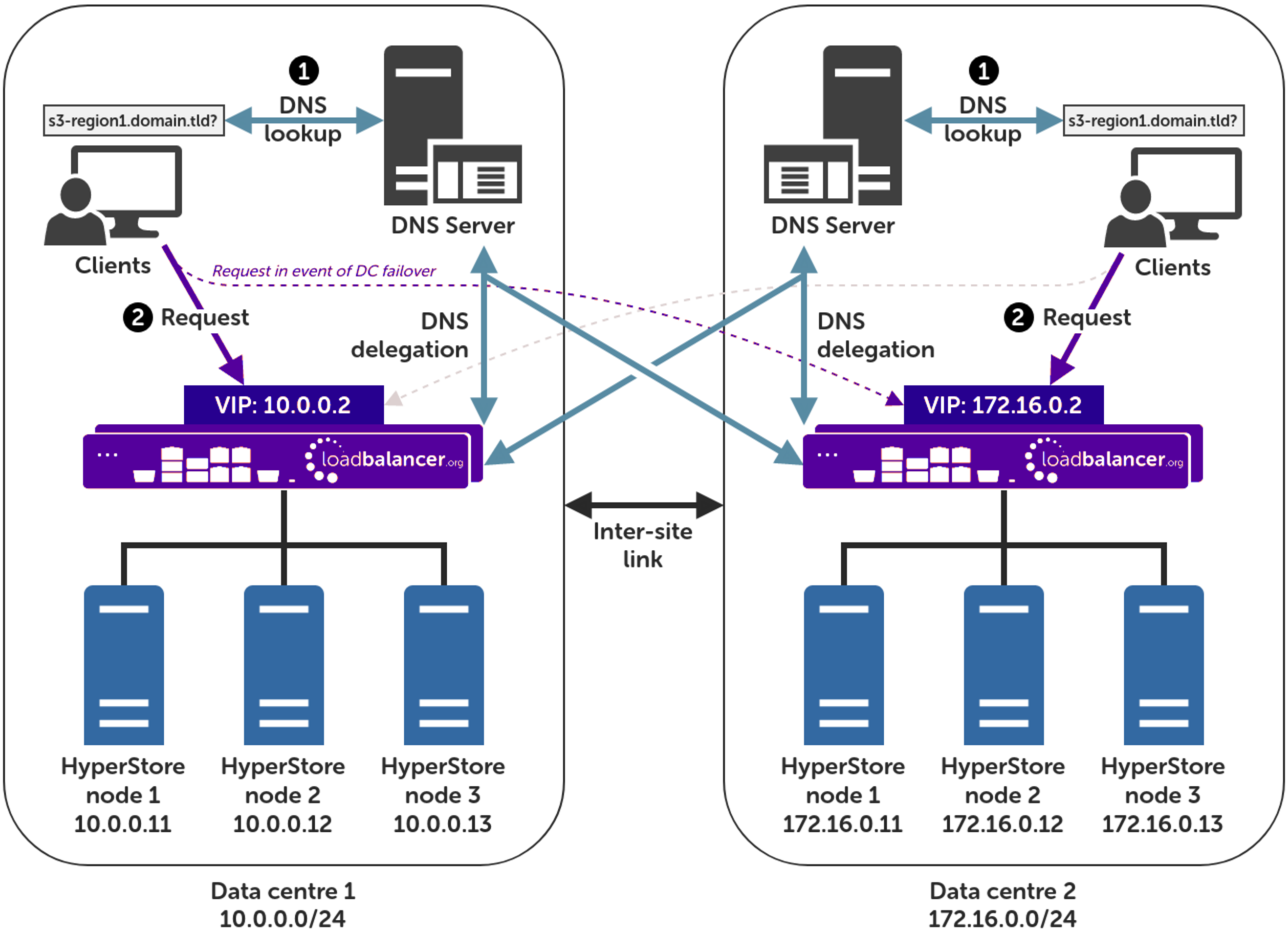
Note	The load balancer can be deployed as a single unit, although Loadbalancer.org recommends a clustered pair for resilience & high availability. Please refer to the section Configuring HA - Adding a Secondary Appliance in the appendix for more details on configuring a clustered pair.

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

A full explanation and instructions on setting up this type of deployment can be found in Section 10, “Appliance Configuration for Cloudian HyperStore – Using 'Direct to Node' GSLB”.

7.3. 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


1 DNS lookup
s3-region1.domain.tld?
DNS Server
Clients
Request in event of DC failover
2 Request
DNS delegation
VIP: 10.0.0.2
loadbalancer.org
HyperStore node 1 10.0.0.11
HyperStore node 2 10.0.0.12
HyperStore node 3 10.0.0.13
Data centre 1 10.0.0.0/24
Inter-site link
1 DNS lookup
s3-region1.domain.tld?
DNS Server
Clients
2 Request
DNS delegation
VIP: 172.16.0.2
loadbalancer.org
HyperStore node 1 172.16.0.11
HyperStore node 2 172.16.0.12
HyperStore node 3 172.16.0.13
Data centre 2 172.16.0.0/24

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 Secondary 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 Multi-Site GSLB and Location Affinity".

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.

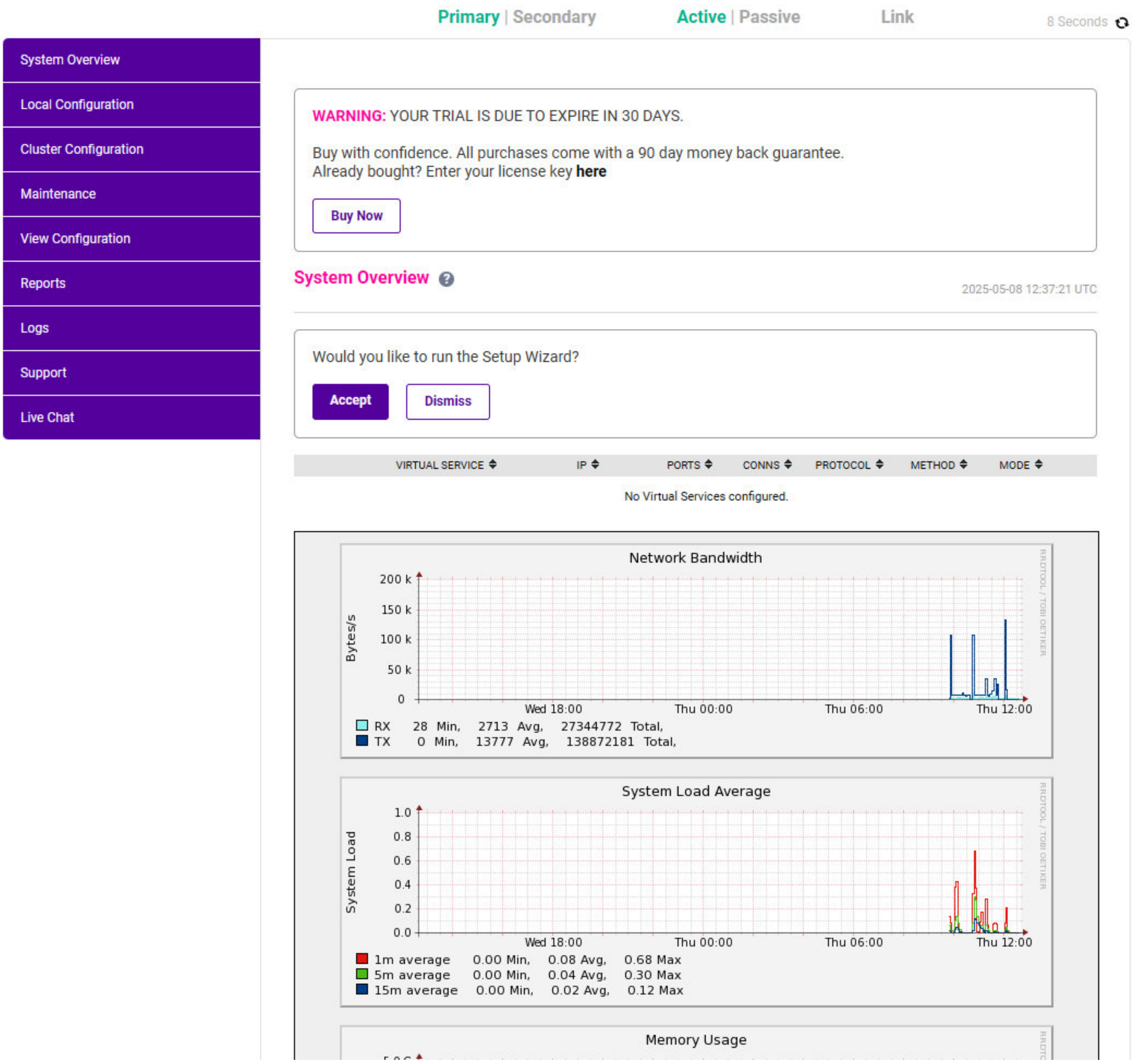
   Note If you need to change the port, IP address or protocol that the WebUI listens on, please refer to Service Socket Addresses.

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:



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**

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

Would you like to run the Setup Wizard?

VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE

No Virtual Services configured.

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

```
🟥 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
```

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

Live Chat - Start a live chat session with one of our Support Engineers

8.4. Appliance Software Update

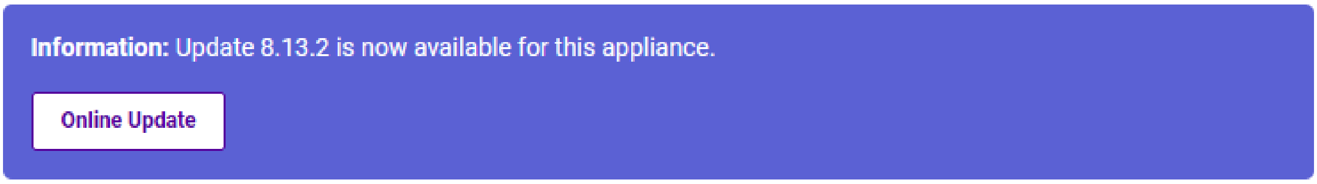
We recommend that the appliance is kept up to date to ensure that you benefit from the latest bug fixes, security updates and feature improvements. Both online and offline update are supported.

ℹ Note	For full details, please refer to Appliance Software Update in the Administration Manual.

ℹ 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:

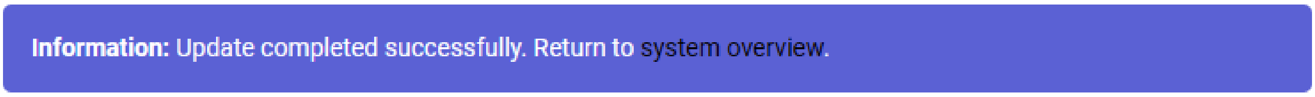


Information: Update 8.13.2 is now available for this appliance.

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.

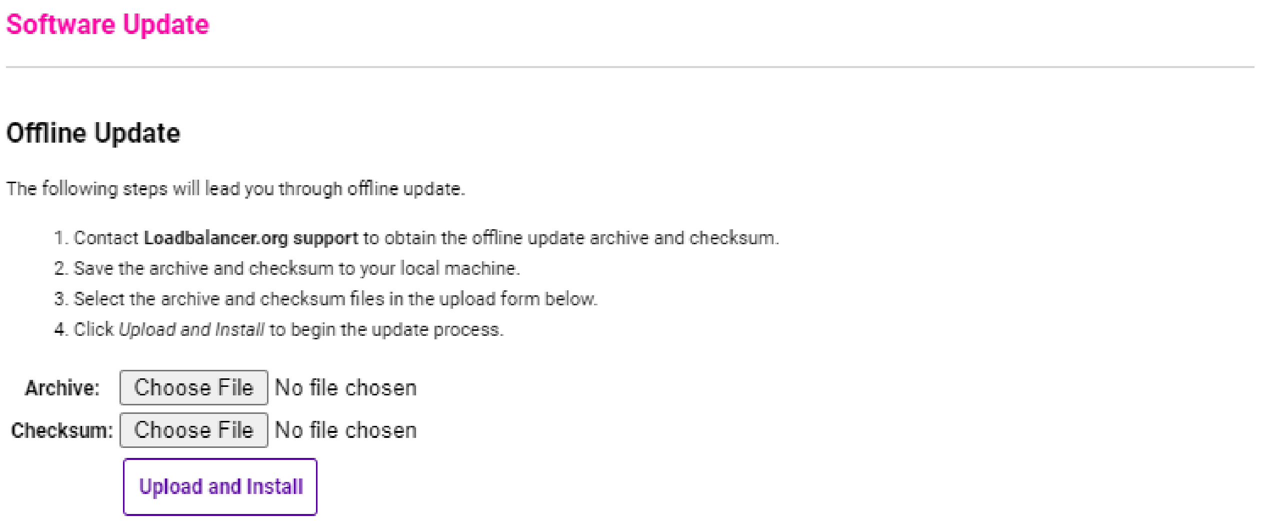
8.4.2. Offline Update

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: Choose File No file chosen

Checksum: Choose File 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 *	DNS / 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 (Centralized/Portal Management)
TCP	9080 *	WebUI - HTTP (disabled by default)
TCP	9081 *	Nginx fallback page
TCP	9443 *	WebUI - HTTPS
TCP	25565 *	Shuttle service (Centralized/Portal Management)

Note

The ports used for SSH, GSLB, SNMP, the WebUI, the fallback page, the gateway service and the shuttle service can be changed if required. For more information, please 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. Appliance Configuration for Cloudian HyperStore – Using Layer 7 SNAT Mode

9.1. Enabling Multithreaded Load Balancing

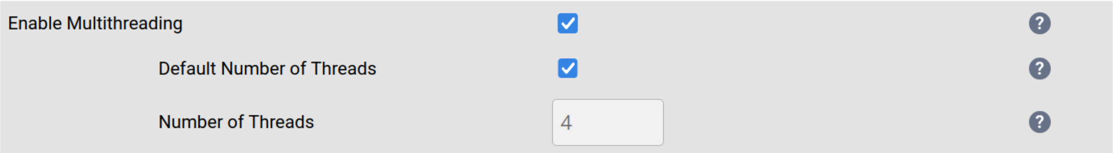
> **Note**
>
> Multithreading is enabled by default for ***new*** load balancers starting from version 8.5.1 and does not require changing.
>
> ***If upgrading an older appliance*** then ensure that the multithreading configuration is set correctly, as described below.

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.

To enable multithreaded mode from the WebUI:

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



Enable Multithreading	☑	?
Default Number of Threads	☑	?
Number of Threads	4	?

9.2. The Duplicate Service Function

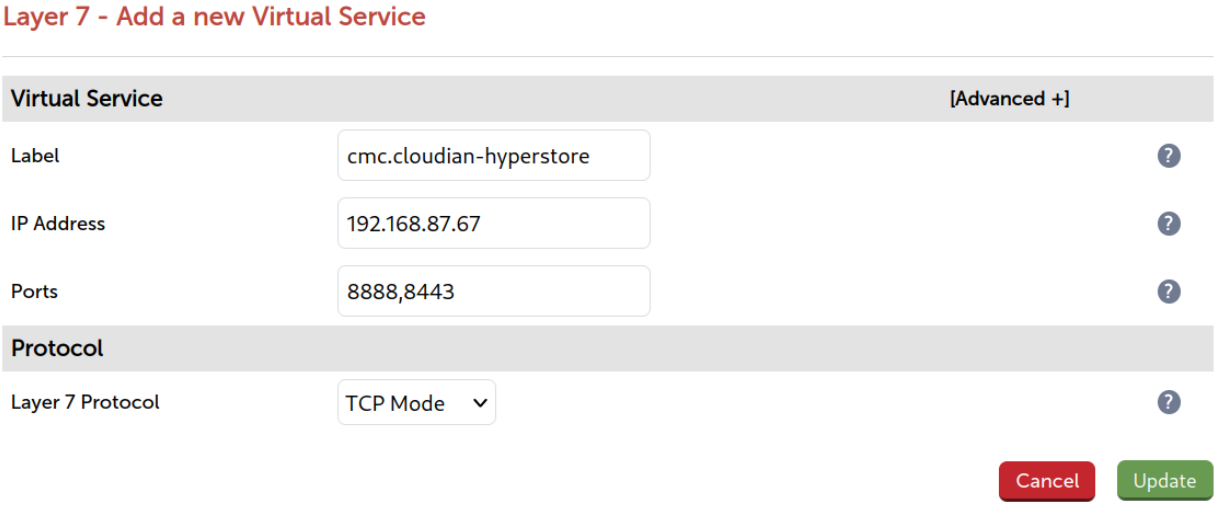
The instructions throughout the remainder of this section make use of the *Duplicate Service* function. This allows an existing virtual service to be "duplicated", along with all real servers associated to that service. This can save a considerable amount of time when configuring the load balancer to 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.

9.3. Configuring VIP 1 – Cloudian Management Console

9.3.1. 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.



Layer 7 - Add a new Virtual Service

Virtual Service [Advanced +]

Label: cmc.cloudian-hyperstore

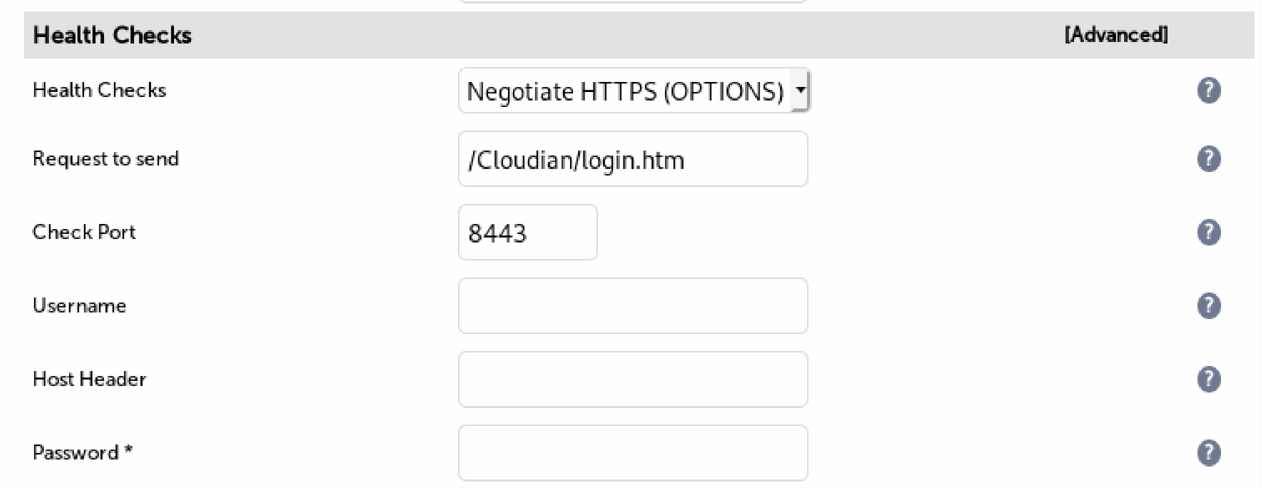
IP Address: 192.168.87.67

Ports: 8888,8443

Protocol

Layer 7 Protocol: TCP Mode

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**.



Health Checks [Advanced]

Health Checks	Negotiate HTTPS (OPTIONS)
Request to send	/Cloudian/login.htm
Check Port	8443
Username	
Host Header	
Password *	

> **Note** The *Host Header* field should be set if appropriate, such as with your S3 endpoint name, for example 's3-region1.domain'.

15. Click **Update**.

9.3.2. Defining the Real Servers (RIPs)

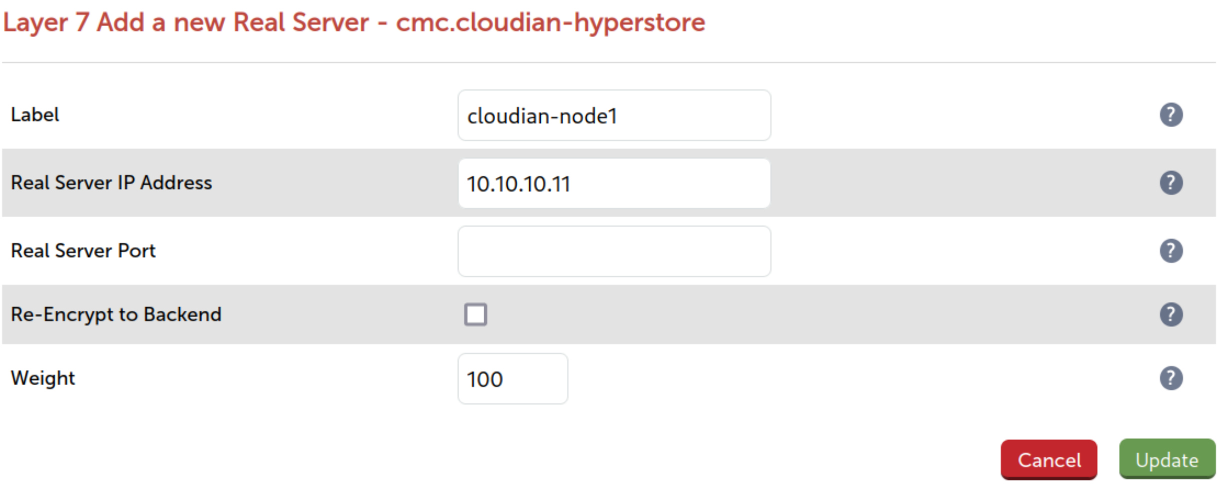
1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Real Servers* and click on **Add a new Real Server** next to the newly created VIP.

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.



Layer 7 Add a new Real Server - cmc.cloudian-hyperstore

Label	cloudian-node1
Real Server IP Address	10.10.10.11
Real Server Port	
Re-Encrypt to Backend	☐
Weight	100

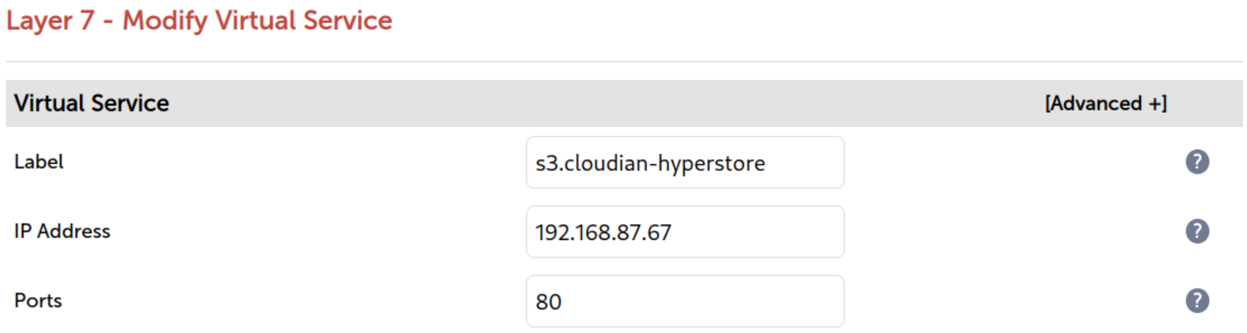
9.4. Configuring VIP 2 – S3 Client Requests (HTTP)

9.4.1. 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**.



Layer 7 - Modify Virtual Service

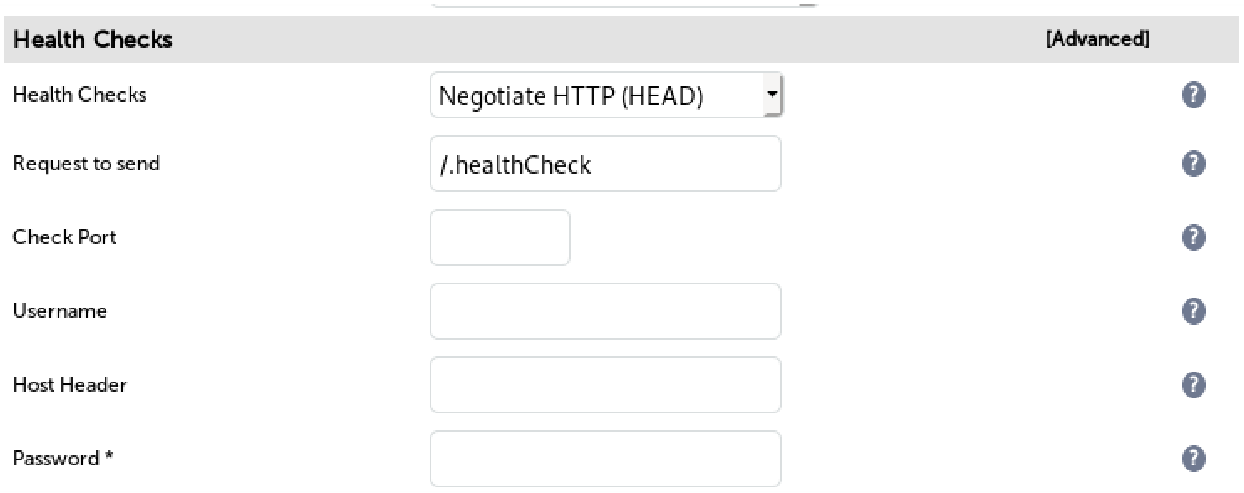
Virtual Service	[Advanced +]
Label	s3.cloudian-hyperstore
IP Address	192.168.87.67
Ports	80

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.



Health Checks	[Advanced]
Health Checks	Negotiate HTTP (HEAD)
Request to send	/.healthCheck
Check Port	
Username	
Host Header	
Password *	

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.

9.5. Configuring VIP 3 – S3 Client Requests (HTTPS)

9.5.1. Configuring the Virtual Service (VIP)

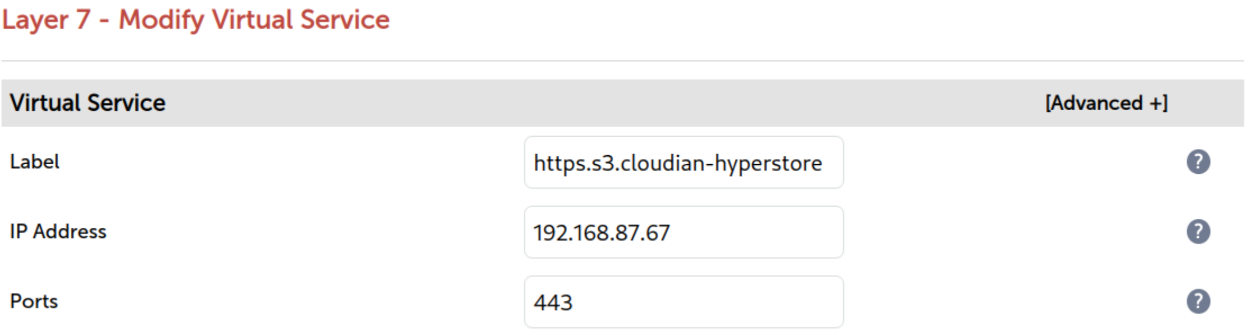
1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click **Modify** next to the previously created S3 VIP.

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**.



Layer 7 - Modify Virtual Service

Virtual Service [Advanced +]

Label: https.s3.cloudian-hyperstore

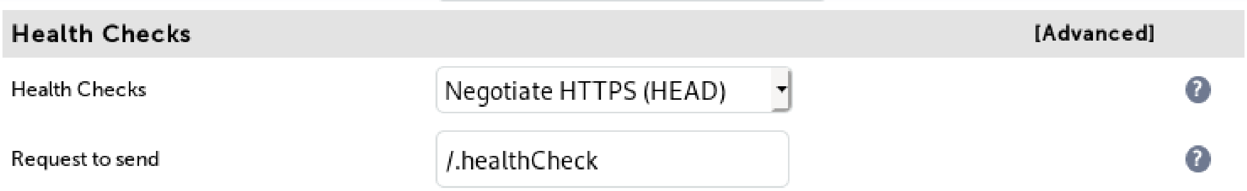
IP Address: 192.168.87.67

Ports: 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)**.



Health Checks [Advanced]

Health Checks: Negotiate HTTPS (HEAD)

Request to send: /.healthCheck

> **Note** 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**.

9.6. Configuring VIP 4 – Admin API Requests

9.6.1. 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:

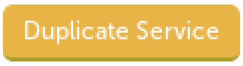
1. Login to the HyperStore puppet master node as `root`, or `sa_admin` if root access is disabled.
2. Execute the following command to retrieve the current `sysadmin` password:

```
[root@hs1 7.2.4]# hsctl config get admin.auth
{
    "base64": "c3lzYWRtaW46SWxpa2VmbGFtaW5nb3MuVGhleSdyZV9uZWF0PQo=",
    "password": "CQA4xFerdMUn8lvoZrbBC6HZ5[D=",
    "username": "sysadmin"
}
```

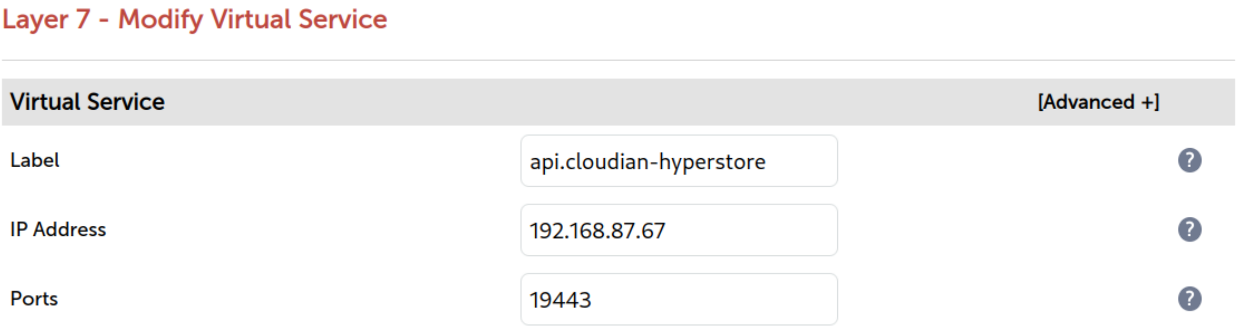
3. Make a note of the password: it is used in the following section when configuring the virtual service.

9.6.2. 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.



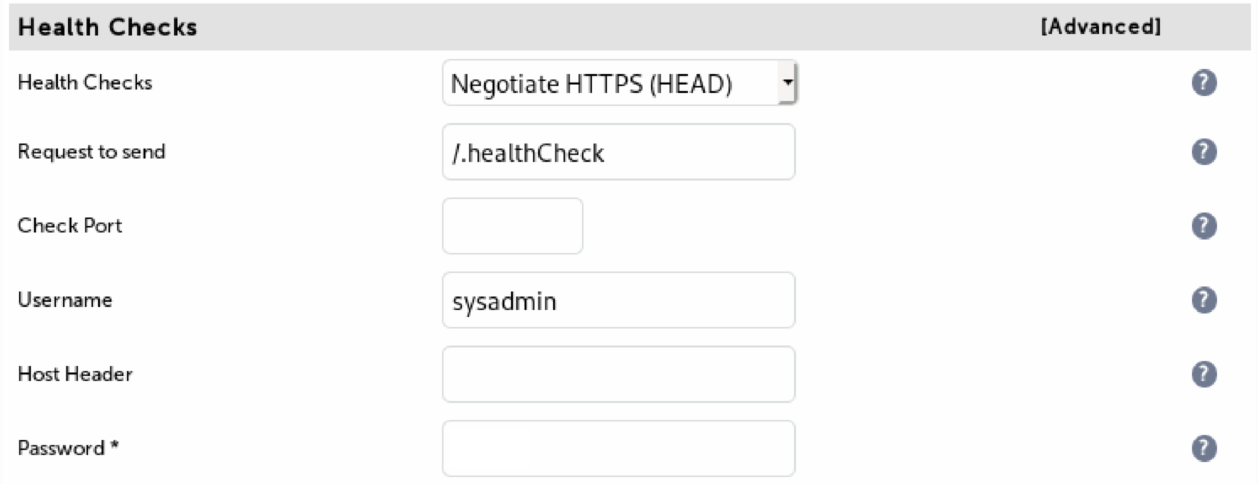
3. Define the *Label* for the new virtual service as required, e.g. **api.cloudian-hyperstore**.
4. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.87.67**.
5. Set the *Ports* field to **19443**.


Layer 7 - Modify Virtual Service

Virtual Service [Advanced +]

Label	api.cloudian-hyperstore
IP Address	192.168.87.67
Ports	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).


Health Checks [Advanced]

Health Checks	Negotiate HTTPS (HEAD)
Request to send	/.healthCheck
Check Port	
Username	sysadmin
Host Header	
Password *	

ℹ Note

The *Host Header* field should be set if appropriate, such as with your S3 endpoint name, for example 's3-region1.domain'.

9. Click **Update**.

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

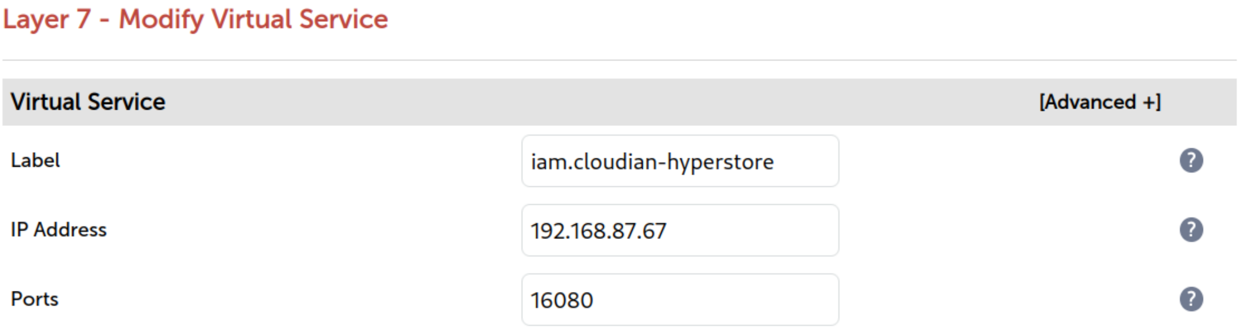
Note

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

9.7.1. 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.

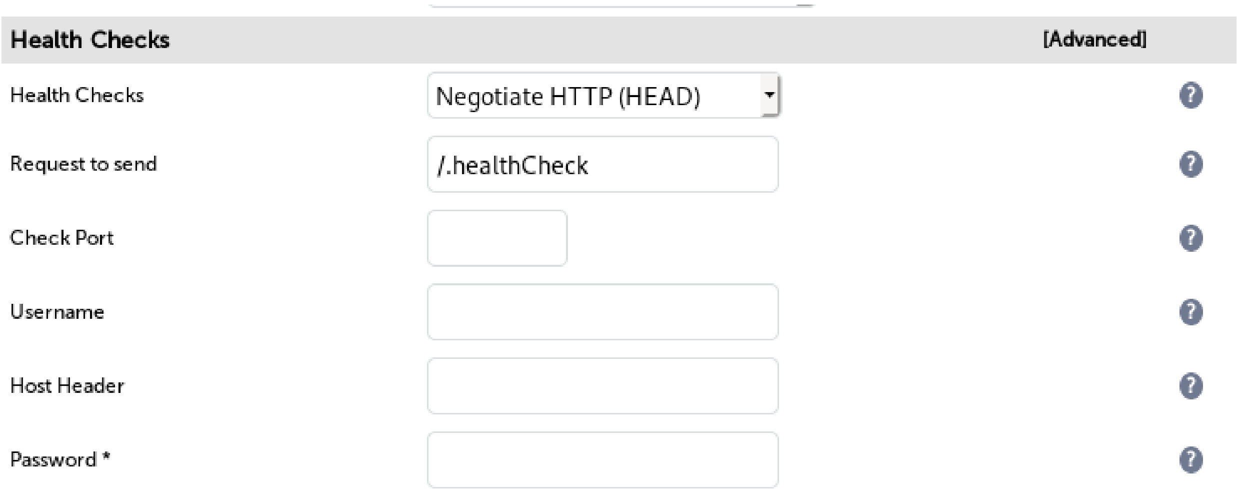
3. Define the *Label* for the new virtual service as required, e.g. **iam.cloudian-hyperstore**.
4. Set the *Virtual Service IP Address* field to the required IP address, e.g. **192.168.87.67**.
5. Set the *Ports* field to **16080**.



Layer 7 - Modify Virtual Service

Virtual Service	[Advanced +]
Label	iam.cloudian-hyperstore
IP Address	192.168.87.67
Ports	16080

6. Set *Health Checks* to **Negotiate HTTP (HEAD)**.
7. Set *Request to send* to **/.healthCheck**
8. In the *Health Checks* section click **Advanced** to expand the menu.
9. Clear the *Username* field to leave it empty.



Health Checks	[Advanced]
Health Checks	Negotiate HTTP (HEAD)
Request to send	/.healthCheck
Check Port	
Username	
Host Header	
Password *	

Note

The *Host Header* field should be set if appropriate, such as with your S3 endpoint name, for example 's3-region1.domain'.

10. Click **Update**.

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

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

9.8.1. Configuring the Virtual Service (VIP)

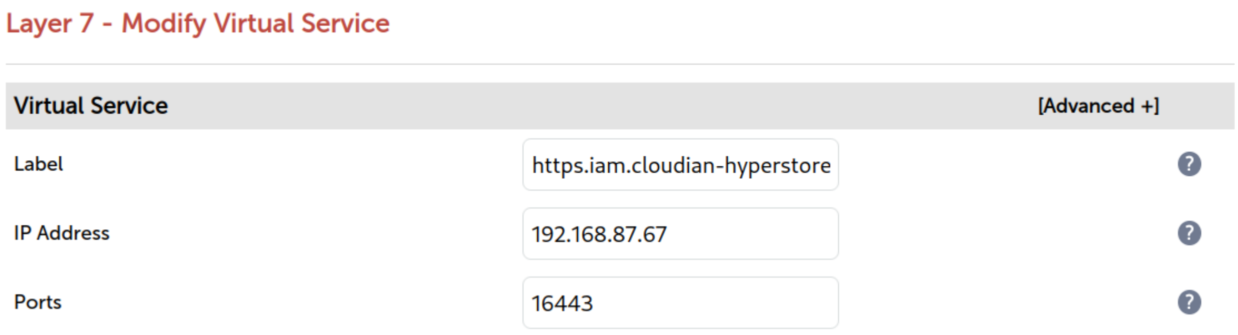
1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click **Modify** next to the previously created IAM VIP.

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**.

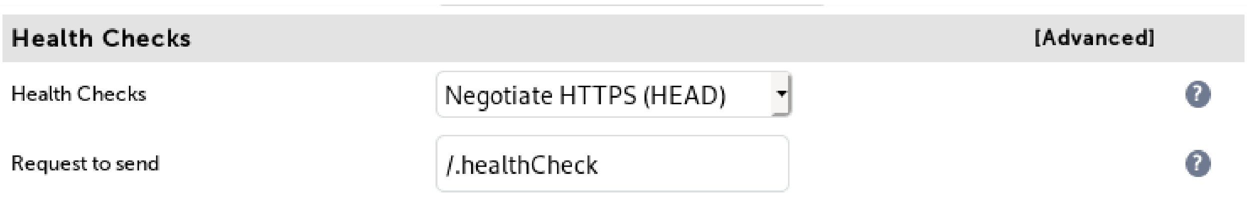


Layer 7 - Modify Virtual Service

Virtual Service [Advanced +]

Label	https.iam.cloudian-hyperstore
IP Address	192.168.87.67
Ports	16443

6. Set *Health Checks* to **Negotiate HTTPS (HEAD)**.



Health Checks [Advanced]

Health Checks	Negotiate HTTPS (HEAD)
Request to send	/.healthCheck

> ℹ Note
>
> 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'.

7. Click **Update**.

9.9. Finalizing the Configuration

To apply the new 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**.

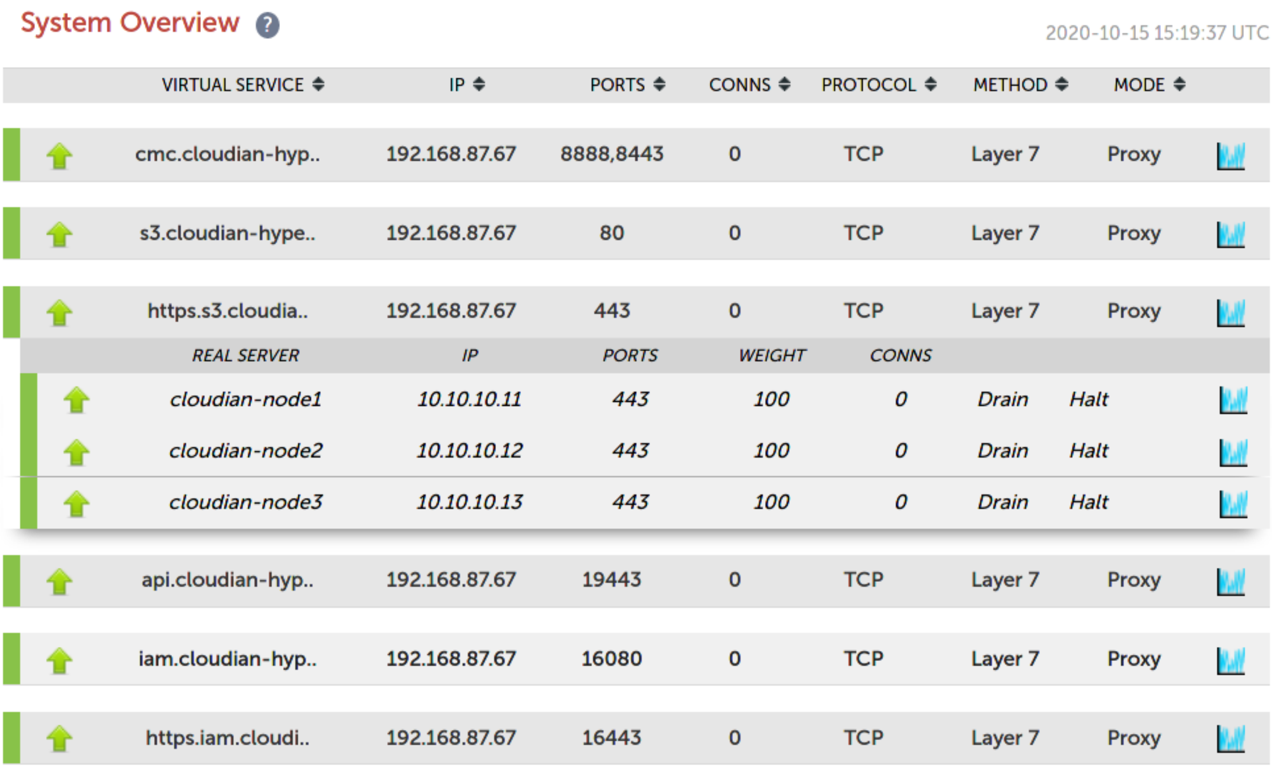
9.10. Testing the Configuration

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.

9.10.1. 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.



System Overview ?

2020-10-15 15:19:37 UTC

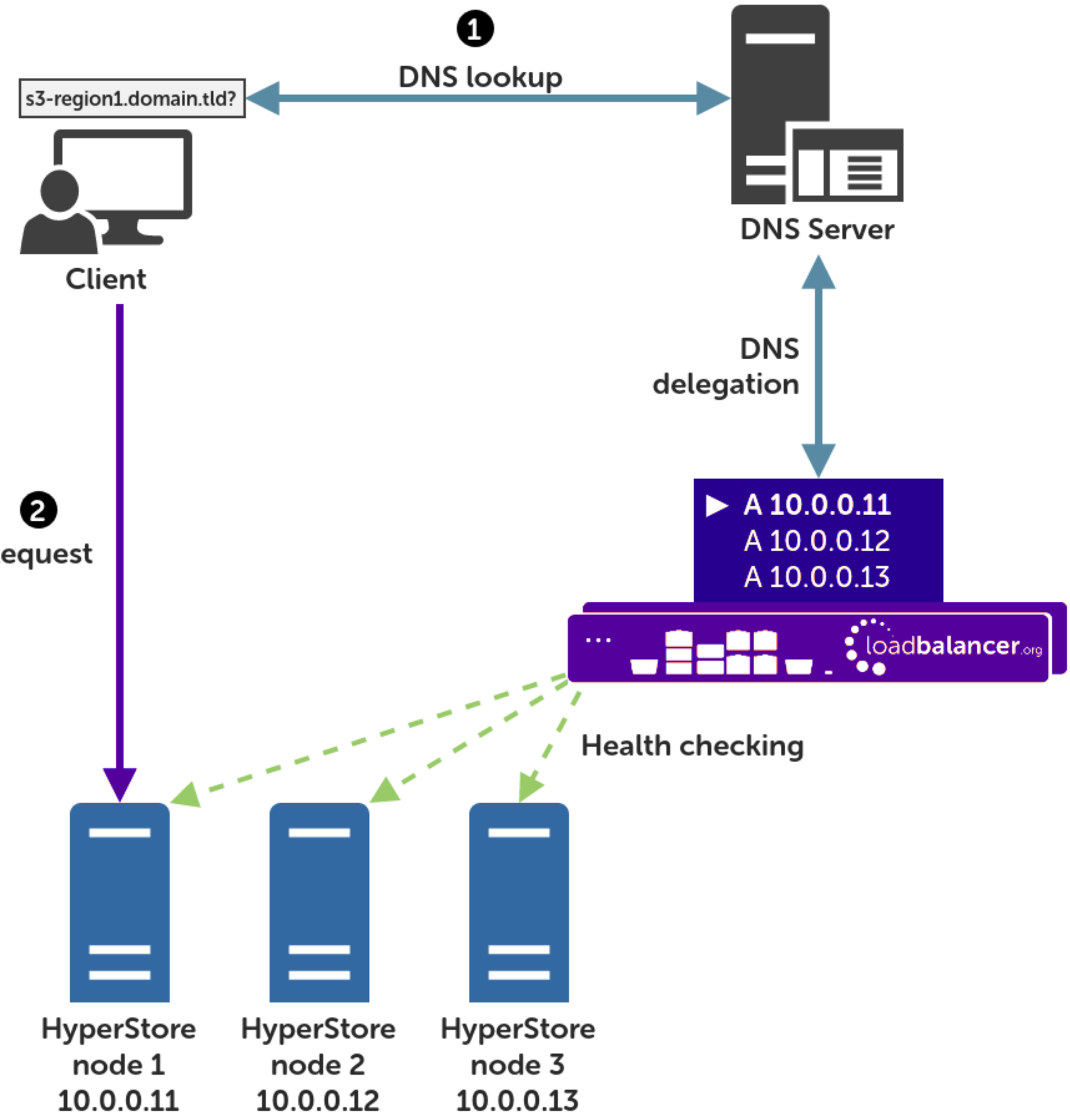
	VIRTUAL SERVICE	IP	PORTS	CONNS	PROTOCOL	METHOD	MODE
↑	cmc.cloudian-hyp..	192.168.87.67	8888,8443	0	TCP	Layer 7	Proxy
↑	s3.cloudian-hype..	192.168.87.67	80	0	TCP	Layer 7	Proxy
↑	https.s3.cloudia..	192.168.87.67	443	0	TCP	Layer 7	Proxy
	REAL SERVER	*IP*	*PORTS*	*WEIGHT*	*CONNS*		
↑	*cloudian-node1*	*10.10.10.11*	*443*	*100*	*0*	*Drain*	*Halt*
↑	*cloudian-node2*	*10.10.10.12*	*443*	*100*	*0*	*Drain*	*Halt*
↑	*cloudian-node3*	*10.10.10.13*	*443*	*100*	*0*	*Drain*	*Halt*
↑	api.cloudian-hyp..	192.168.87.67	19443	0	TCP	Layer 7	Proxy
↑	iam.cloudian-hyp..	192.168.87.67	16080	0	TCP	Layer 7	Proxy
↑	https.iam.cloudi..	192.168.87.67	16443	0	TCP	Layer 7	Proxy

10. Appliance Configuration for Cloudian HyperStore – Using 'Direct to Node' GSLB

10.1. 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:

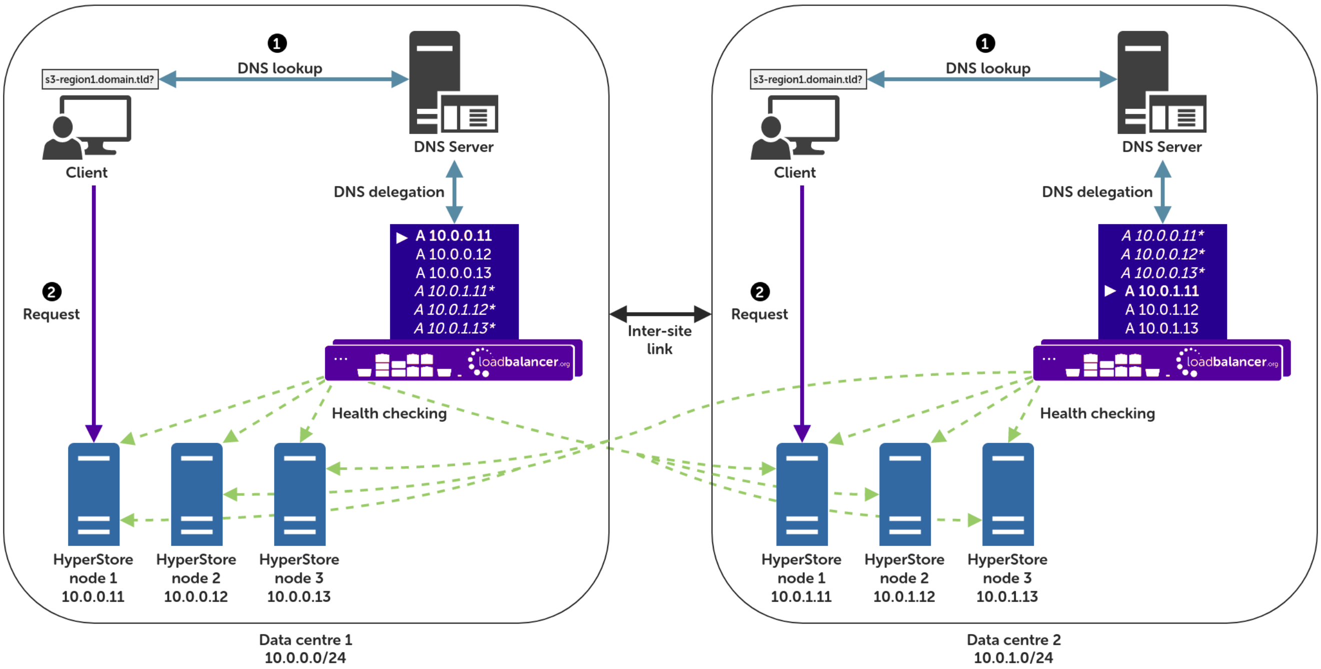
- **Start**: A client tries to access the S3 service by using the service's fully qualified domain name, in this example `s3-region1.domain.tld`
- The client sends a DNS query for `s3-region1.domain.tld` to the DNS server.
- The DNS server has a CNAME record for `s3-region1.domain.tld` which points to 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. The load balancer answers with 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.

10.2. 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.



* Address record of a HyperStore node not located at the local site.
Only served in the event that all local nodes are unavailable

10.3. 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.

10.4. Handling Multiple Subdomains, Including Wildcard Subdomains

10.4.1. 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.

10.4.2. 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:

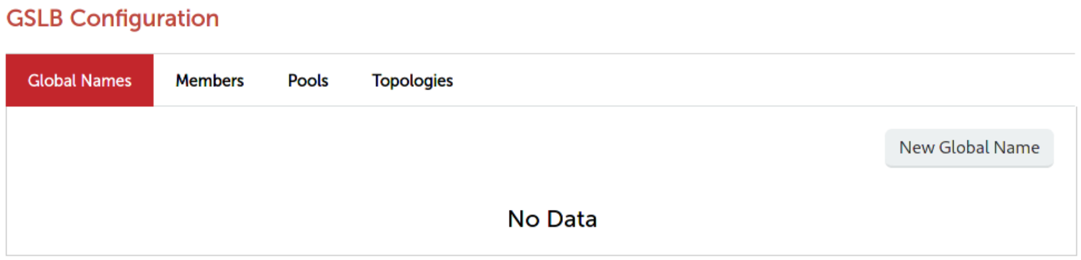
cmc.	Delegate to the load balancer
gslb.	Delegate to the load balancer
s3-admin.	CNAME to gslb.domain.tld
s3-<region>.	CNAME to gslb.domain.tld
*.s3-<region>.	CNAME to gslb.domain.tld
iam.	CNAME to gslb.domain.tld

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

10.5. Appliance Configuration

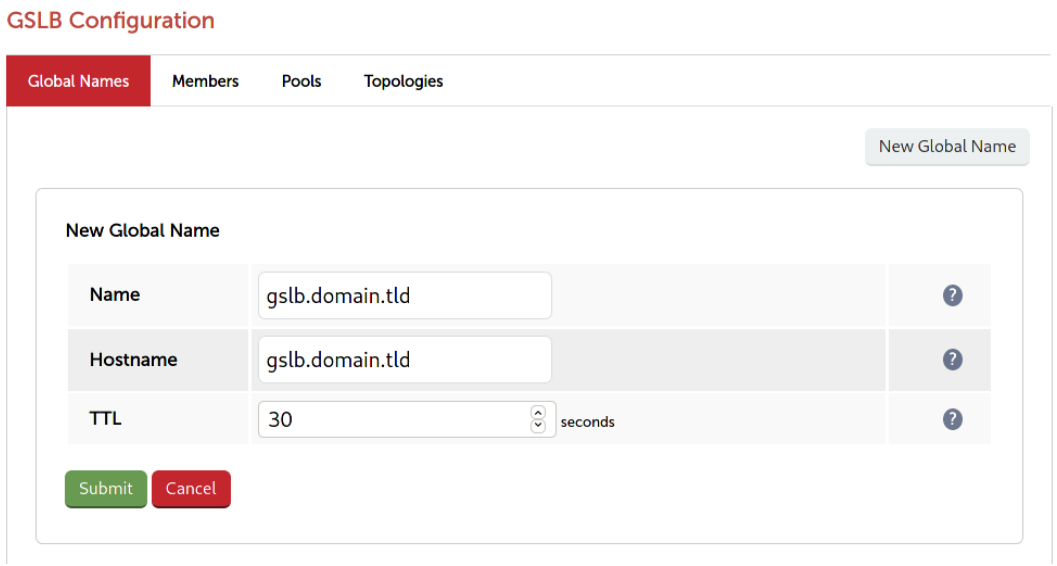
The GSLB service should be configured on the **primary** load balancer appliance and should be configured at each site if a multi-site deployment is being configured.

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



10.5.1. Step 1 – Configuring the Global Name

1. Using the WebUI on the primary 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**.



New Global Name

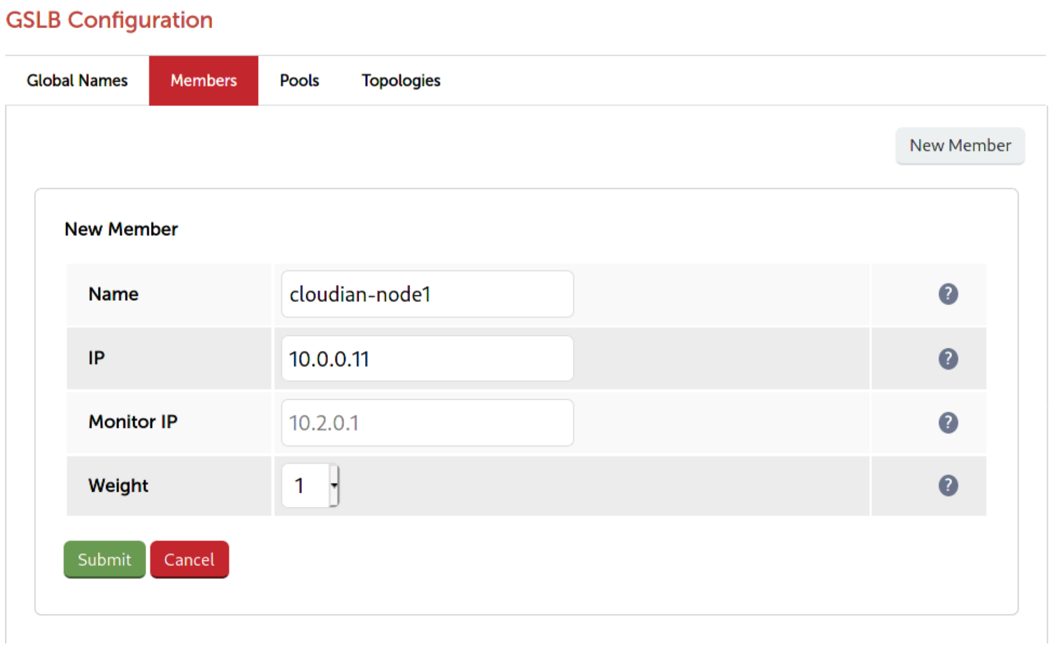
Name	gslb.domain.tld
Hostname	gslb.domain.tld
TTL	30 seconds

10.5.2. 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).



New Member

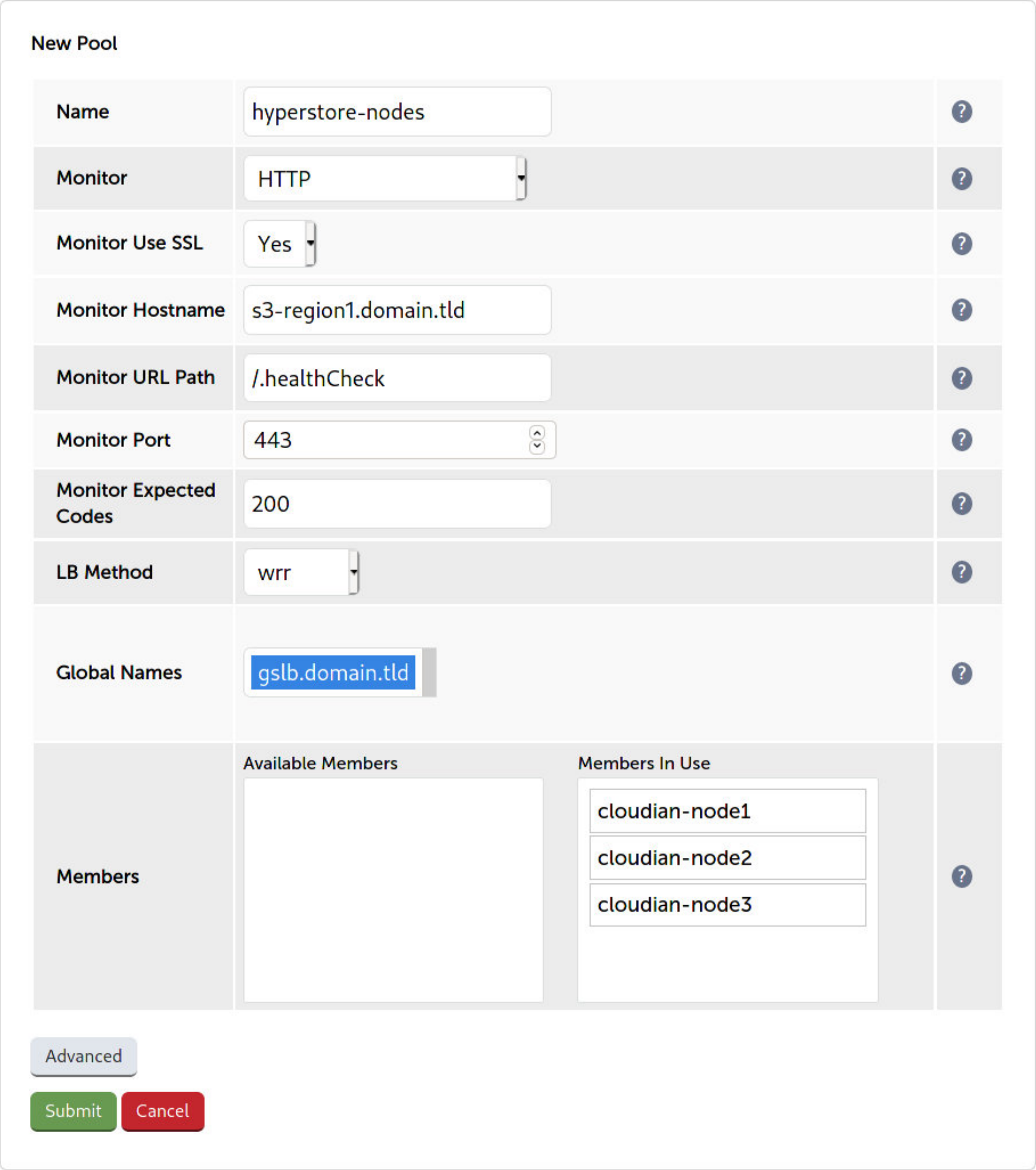
Name	cloudian-node1
IP	10.0.0.11
Monitor IP	10.2.0.1
Weight	1

10.5.3. 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**.



New Pool

Name	hyperstore-nodes
Monitor	HTTP
Monitor Use SSL	Yes
Monitor Hostname	s3-region1.domain.tld
Monitor URL Path	/.healthCheck
Monitor Port	443
Monitor Expected Codes	200
LB Method	wrr
Global Names	gslb.domain.tld
Members	Members In Use: cloudian-node1 cloudian-node2 cloudian-node3

10.5.4. 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 almost all situations.

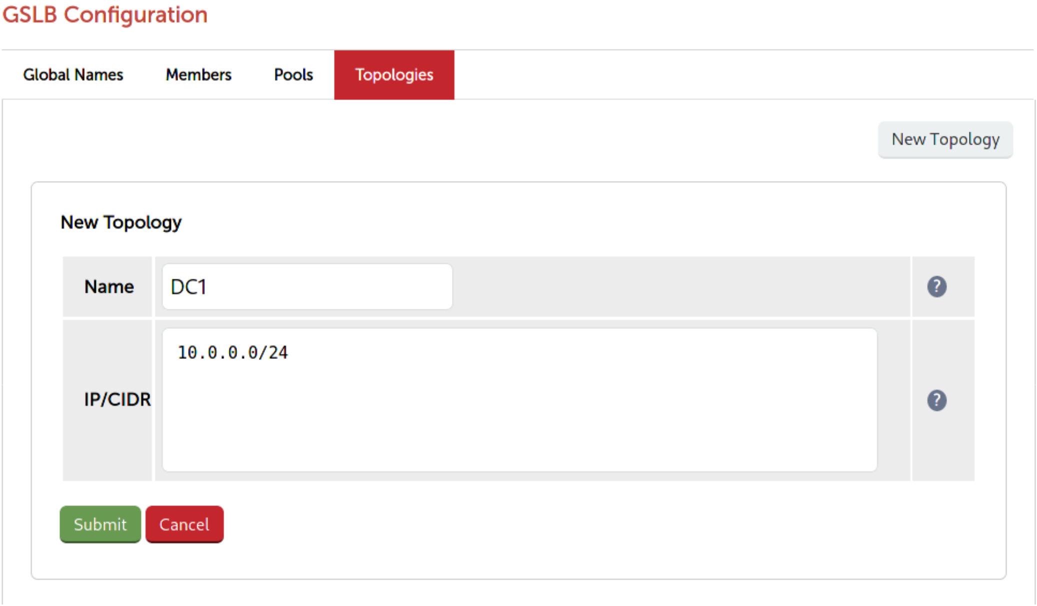
1. Select the **Topologies** tab.
2. Click the **New Topology** button.
3. Enter a friendly *Name* for the topology, e.g. **DC1**.

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.



Name: DC1

IP/CIDR: 10.0.0.0/24

10.5.5. Step 5 – Configure the Separate CMC Service

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

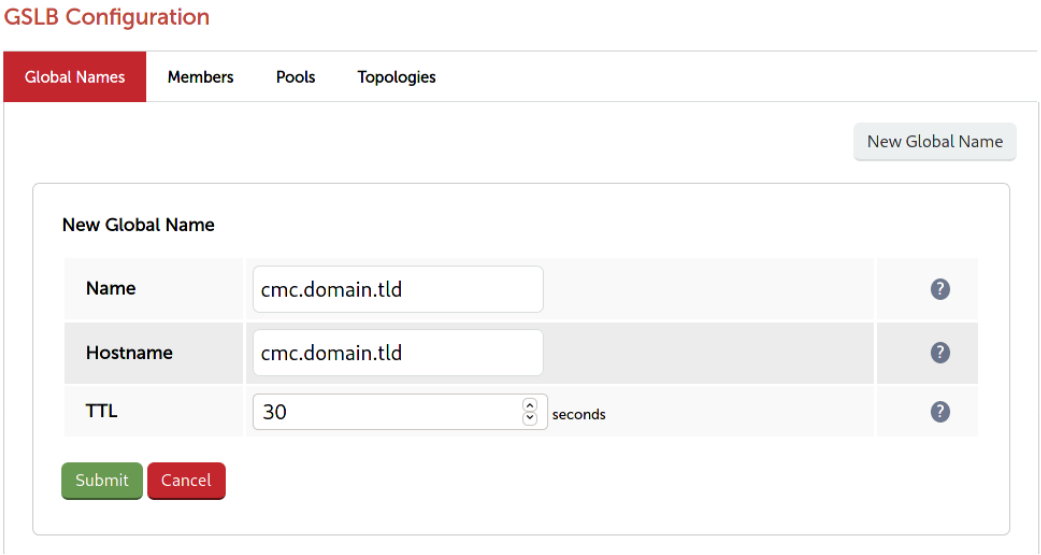
2. Select the **Global Names** tab.

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**.



New Global Name

Name	cmc.domain.tld	
Hostname	cmc.domain.tld	
TTL	30	seconds

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**.

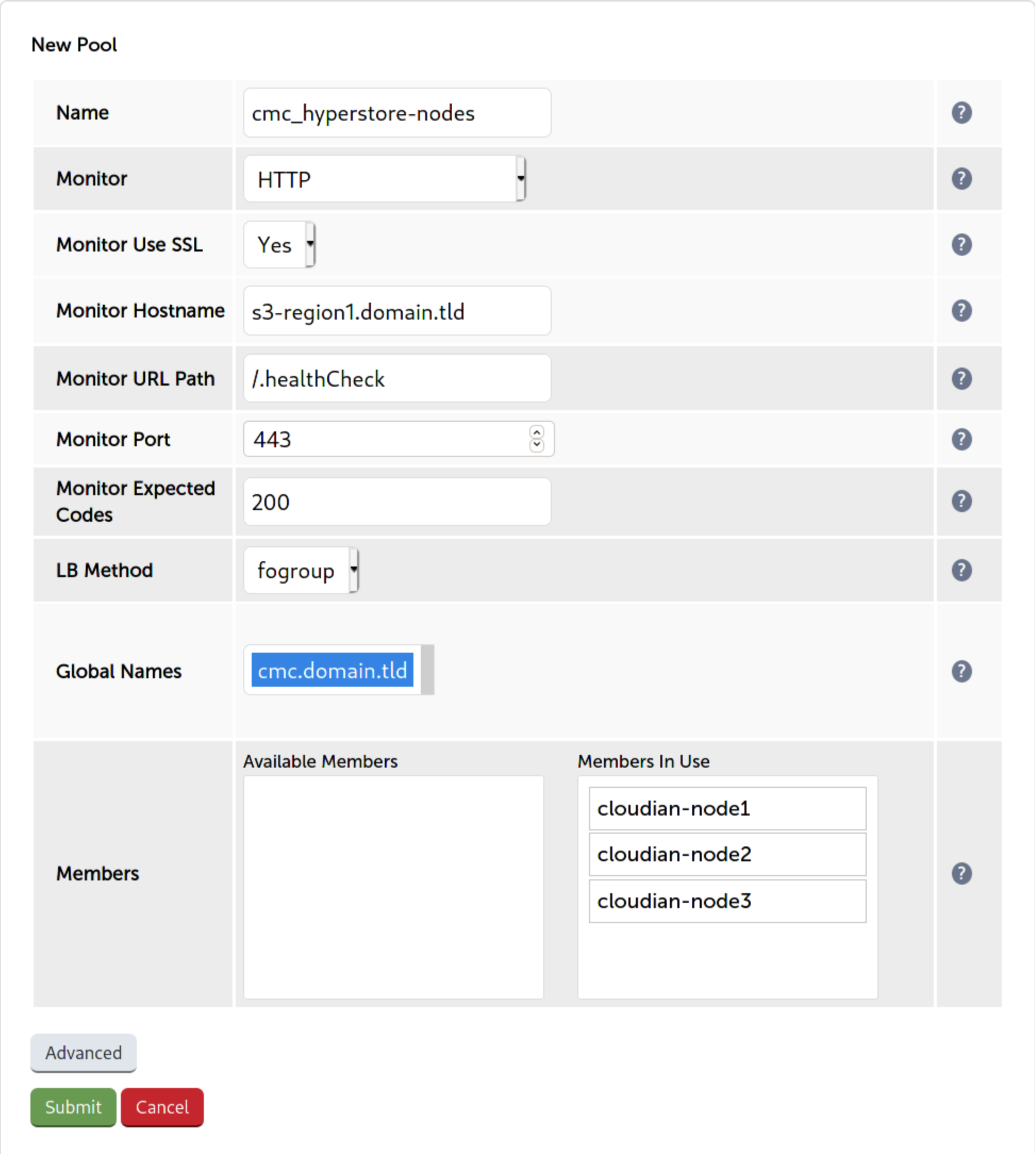
16. Set *LB Method* to **fogroup**.

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.

> ℹ 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**.



New Pool

Name	cmc_hyperstore-nodes
Monitor	HTTP
Monitor Use SSL	Yes
Monitor Hostname	s3-region1.domain.tld
Monitor URL Path	/.healthCheck
Monitor Port	443
Monitor Expected Codes	200
LB Method	fogroup
Global Names	cmc.domain.tld
Members	Available Members: Members In Use: cloudian-node1 cloudian-node2 cloudian-node3

10.5.6. 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**.

10.6. DNS Server Configuration

Once the GSLB service has been configured on the primary 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 section, the DNS server at each site would be configured with a delegation for the domain `gslb.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 Microsoft DNS Server Configuration.

10.7. 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 primary 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
```

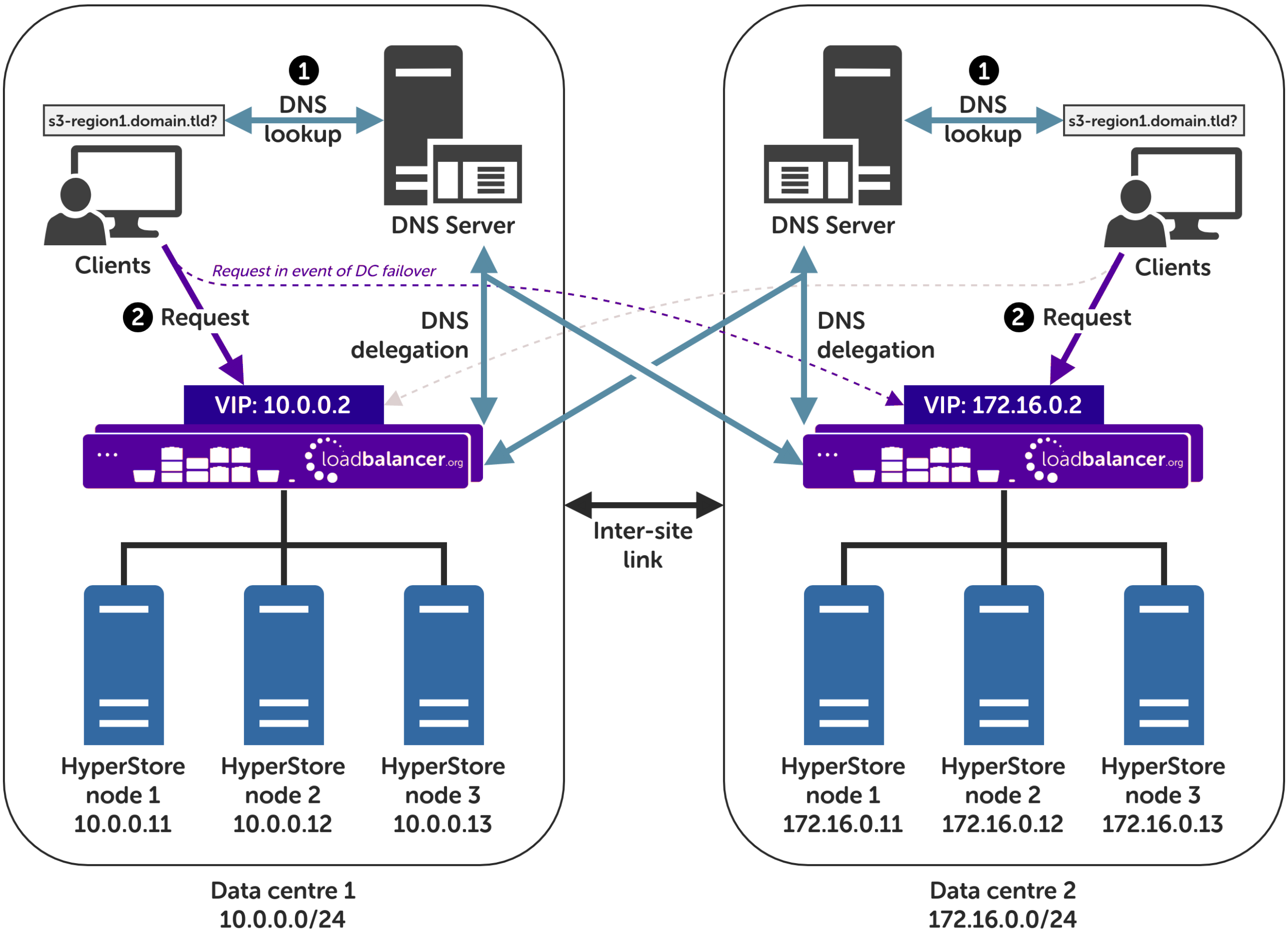
11. Appliance Configuration for Cloudian HyperStore – Using Multi-Site GSLB and Location Affinity

11.1. Conceptual Overview

For **multi-site HyperStore deployments**, it is possible to use the load balancer's global server load balancing (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.


s3-region1.domain.tld?
1 DNS lookup
DNS Server
Clients
Request in event of DC failover
2 Request
DNS delegation
VIP: 10.0.0.2
loadbalancer.org
HyperStore node 1 10.0.0.11
HyperStore node 2 10.0.0.12
HyperStore node 3 10.0.0.13
Data centre 1 10.0.0.0/24
Inter-site link
s3-region1.domain.tld?
1 DNS lookup
DNS Server
Clients
2 Request
DNS delegation
VIP: 172.16.0.2
loadbalancer.org
HyperStore node 1 172.16.0.11
HyperStore node 2 172.16.0.12
HyperStore node 3 172.16.0.13
Data centre 2 172.16.0.0/24

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 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 DNS 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.

Note

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.

11.2. DNS Server Prerequisites

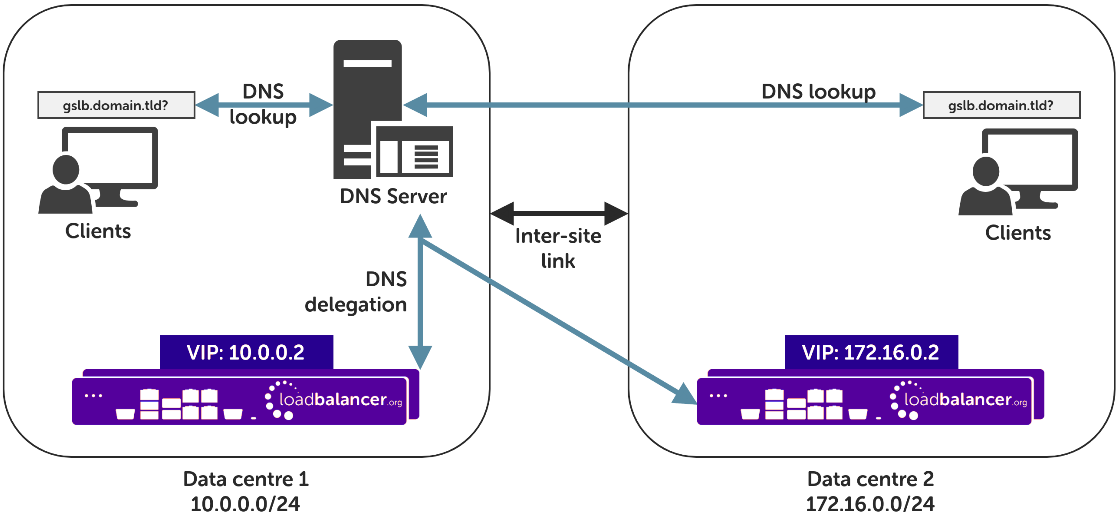
Important

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.

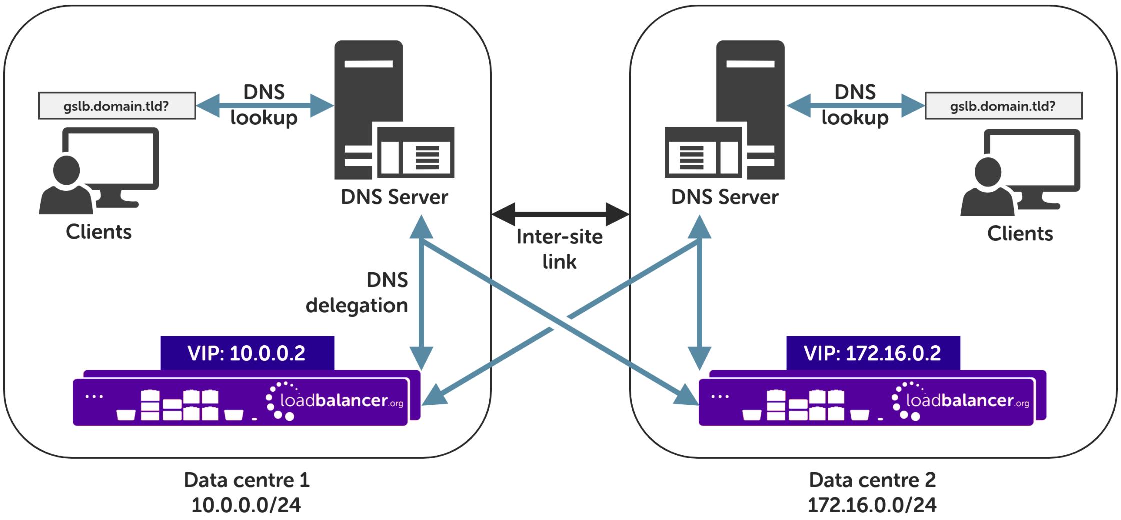
If multiple sites ***share*** a common DNS server then ***clients cannot be directed to their local, on-site HyperStore instance***.

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) 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.)

11.3. 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.

11.4. Handling Multiple Subdomains, Including Wildcard Subdomains

11.4.1. 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`.

11.4.2. 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:

`cmc.`	Delegate to the load balancer
`gslb.`	Delegate to the load balancer
`s3-admin.`	CNAME to `gslb.domain.tld`
`s3-<region>.`	CNAME to `gslb.domain.tld`
`*.s3-<region>.`	CNAME to `gslb.domain.tld`
`iam.`	CNAME to `gslb.domain.tld`

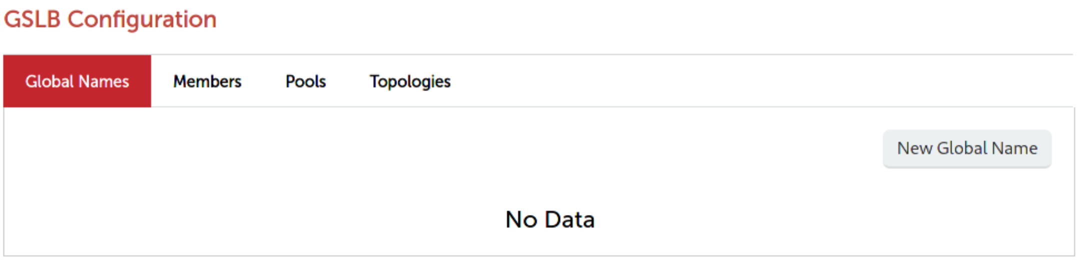
This approach simplifies DNS entry configuration, particularly when wildcard entries are involved.

11.5. Appliance Configuration

The GSLB service should be configured on the **primary** 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*:



No Data

11.5.1. Step 1 – Configuring the Global Name

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

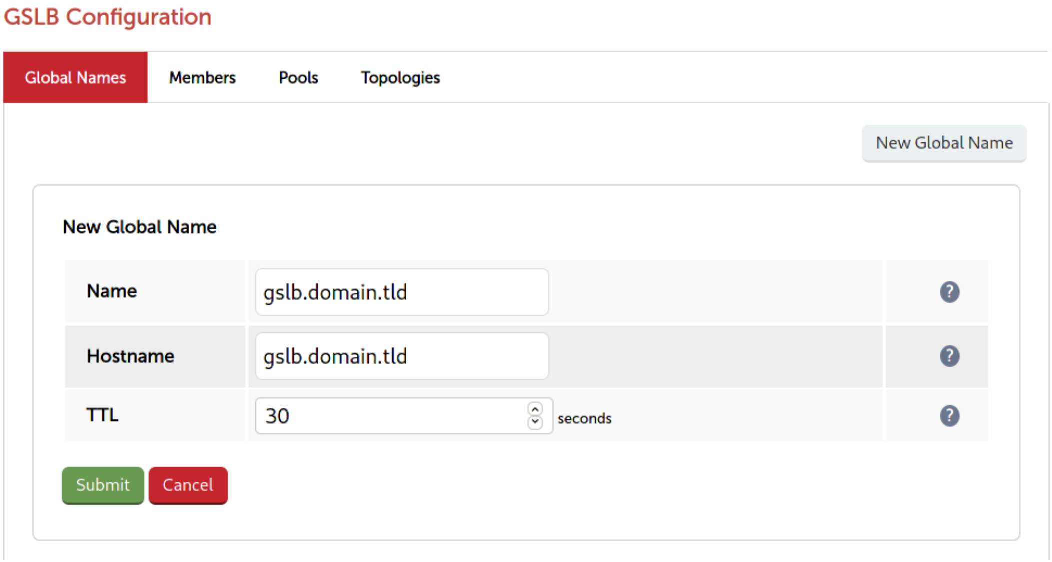
2. Select the **Global Names** tab.

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**.



Name	gslb.domain.tld	
Hostname	gslb.domain.tld	
TTL	30	seconds

11.5.2. 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.

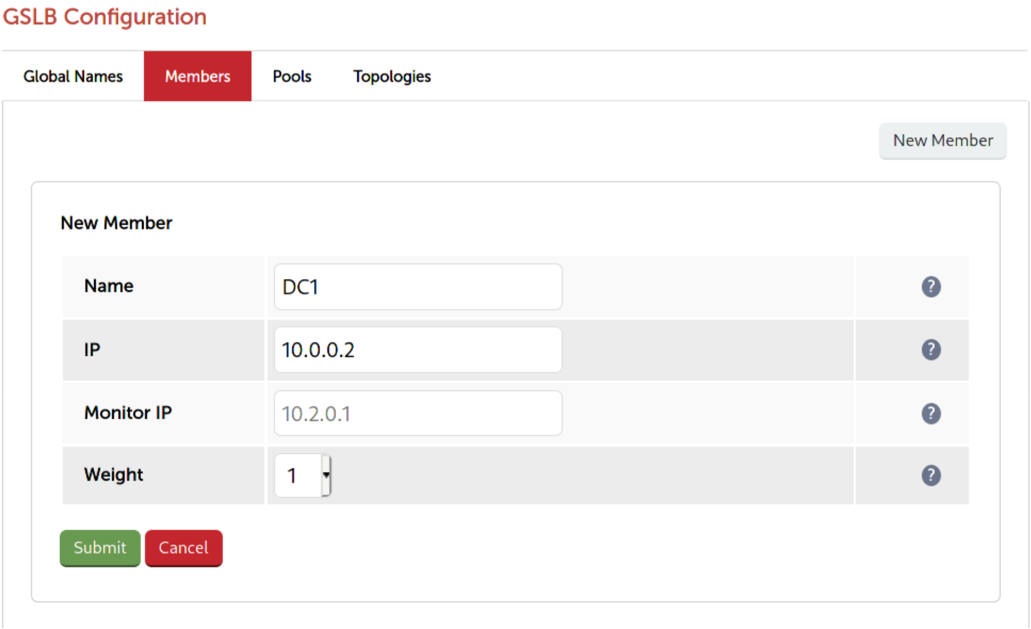
3. Enter a friendly *Name* for the member, e.g. **DC1**.

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.



New Member

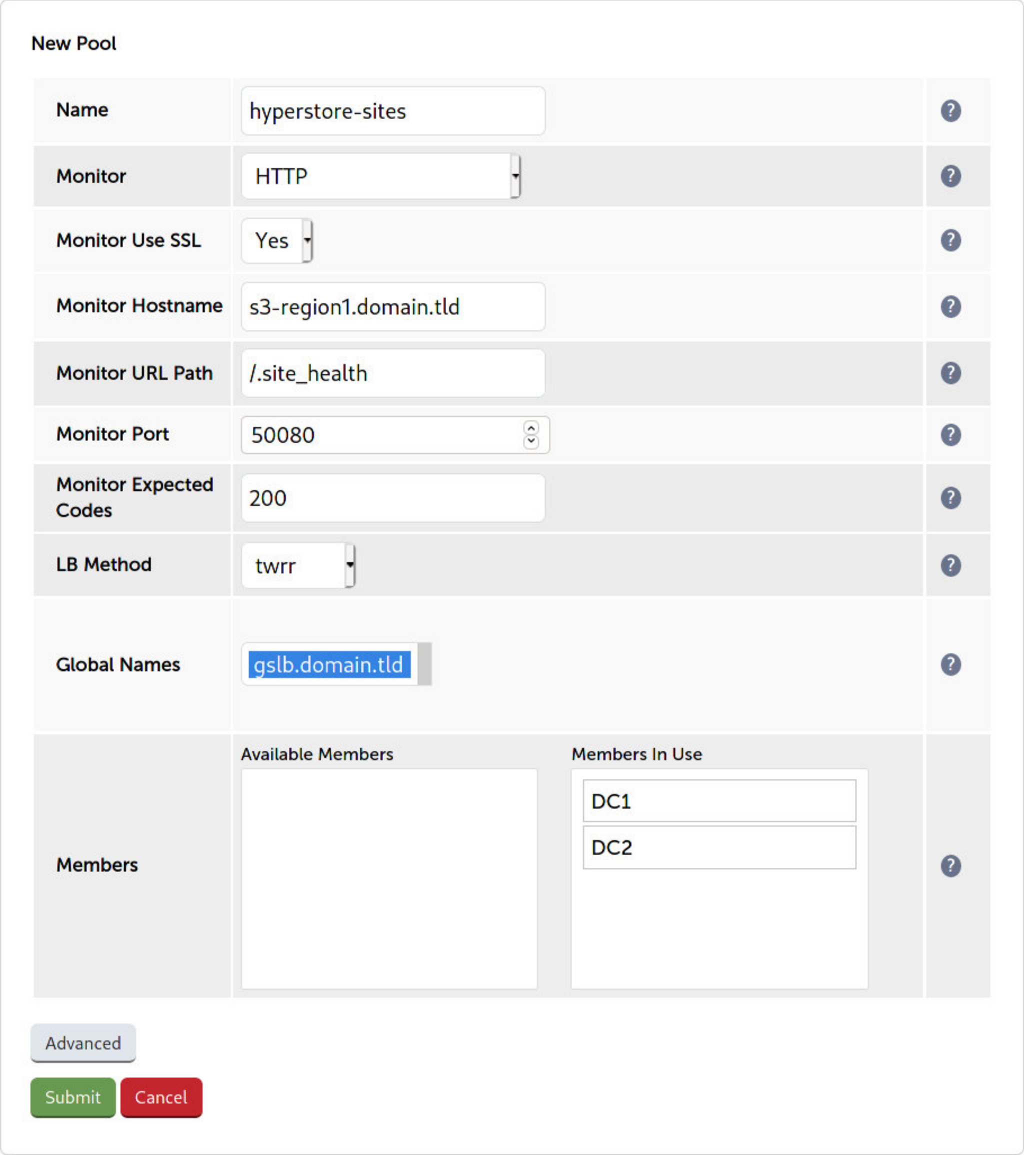
Name	DC1
IP	10.0.0.2
Monitor IP	10.2.0.1
Weight	1

11.5.3. 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**.



New Pool

Name	hyperstore-sites
Monitor	HTTP
Monitor Use SSL	Yes
Monitor Hostname	s3-region1.domain.tld
Monitor URL Path	/.site_health
Monitor Port	50080
Monitor Expected Codes	200
LB Method	twrr
Global Names	gslb.domain.tld
Members	Available Members: Members In Use: DC1 DC2

11.5.4. Step 4 – Configure the Topology

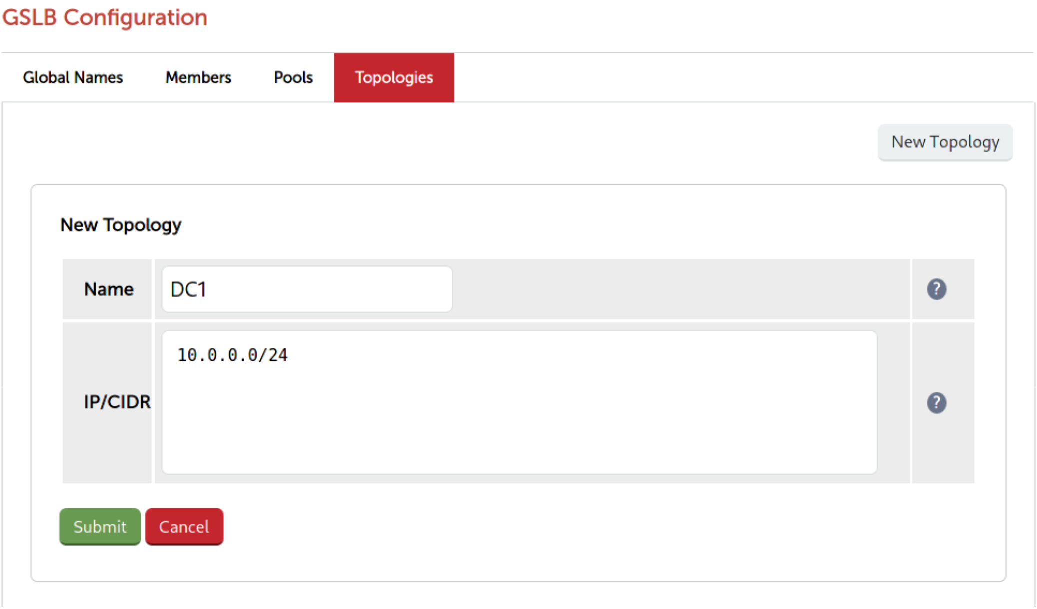
Topology configuration is used to map subnets to sites. This gives the solution its location awareness, allowing clients to be directed to their *local* HyperStore instance instead of being bounced between every site which has been defined.

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.



Name: DC1

IP/CIDR: 10.0.0.0/24

11.5.5. 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 primary appliance for the first site, navigate to *Cluster Configuration > GSLB Configuration*.

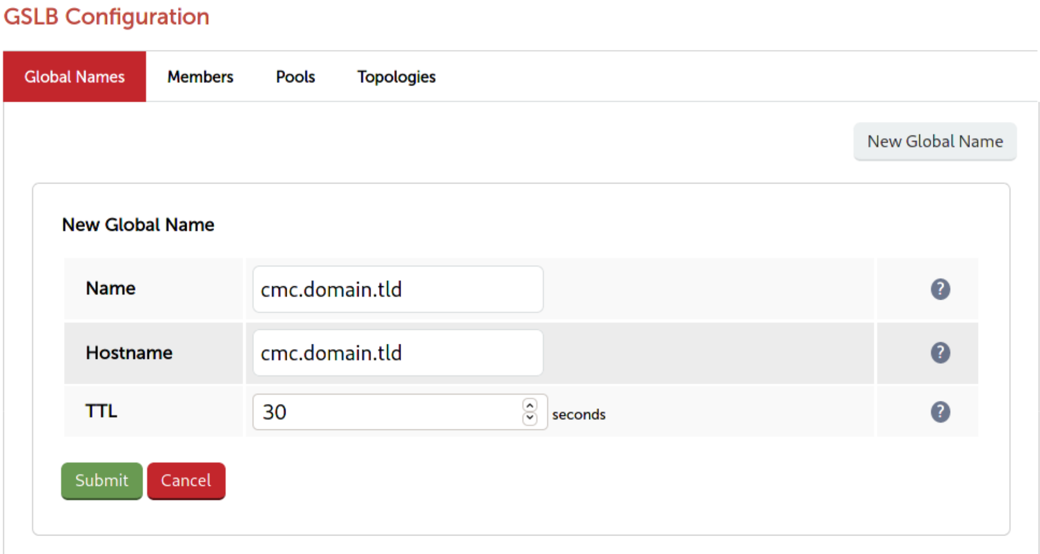
2. Select the **Global Names** tab.

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

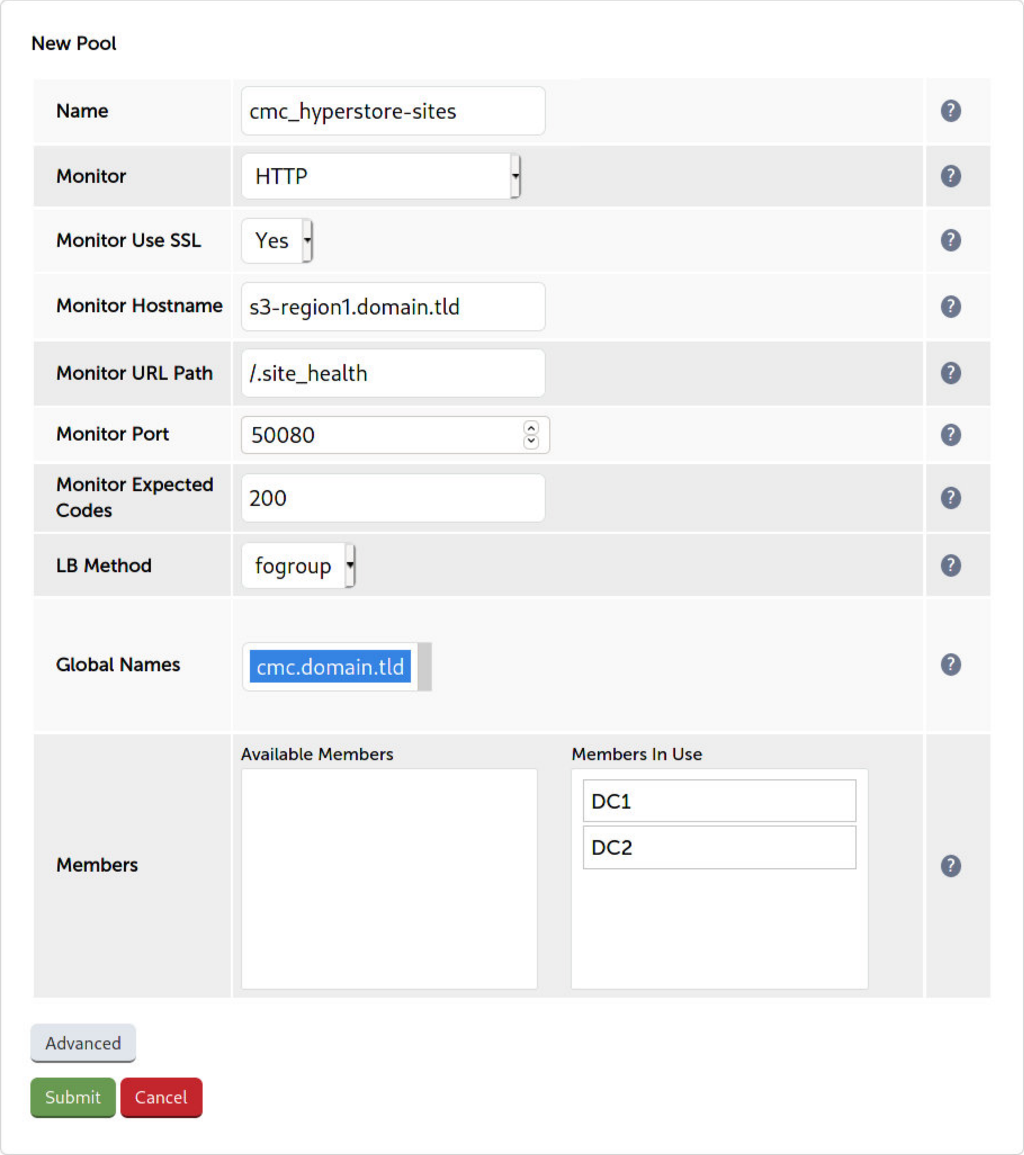
New Global Name

Name	cmc.domain.tld	?
Hostname	cmc.domain.tld	?
TTL	30 seconds	?

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.

> ℹ Note
>
> **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**.



New Pool

Name	cmc_hyperstore-sites
Monitor	HTTP
Monitor Use SSL	Yes
Monitor Hostname	s3-region1.domain.tld
Monitor URL Path	/.site_health
Monitor Port	50080
Monitor Expected Codes	200
LB Method	fogroup
Global Names	cmc.domain.tld
Members	Available Members: Members In Use: DC1 DC2

11.5.6. 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**.

11.5.7. Step 7 – Intelligent Site Health Check

On the primary 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:

```
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
```

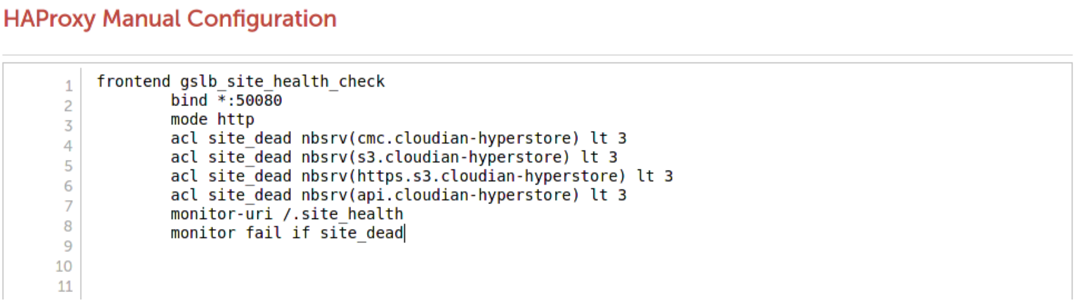
The `3` in `lt 3` should be set as appropriate for the deployment in question. This number represents the **minimum number of HyperStore nodes needed to provide full (read *and* write) access to all services**. In this specific example, having three HyperStore nodes healthy and online is sufficient to 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:



HAProxy Manual Configuration

```
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
```

11.5.8. 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.

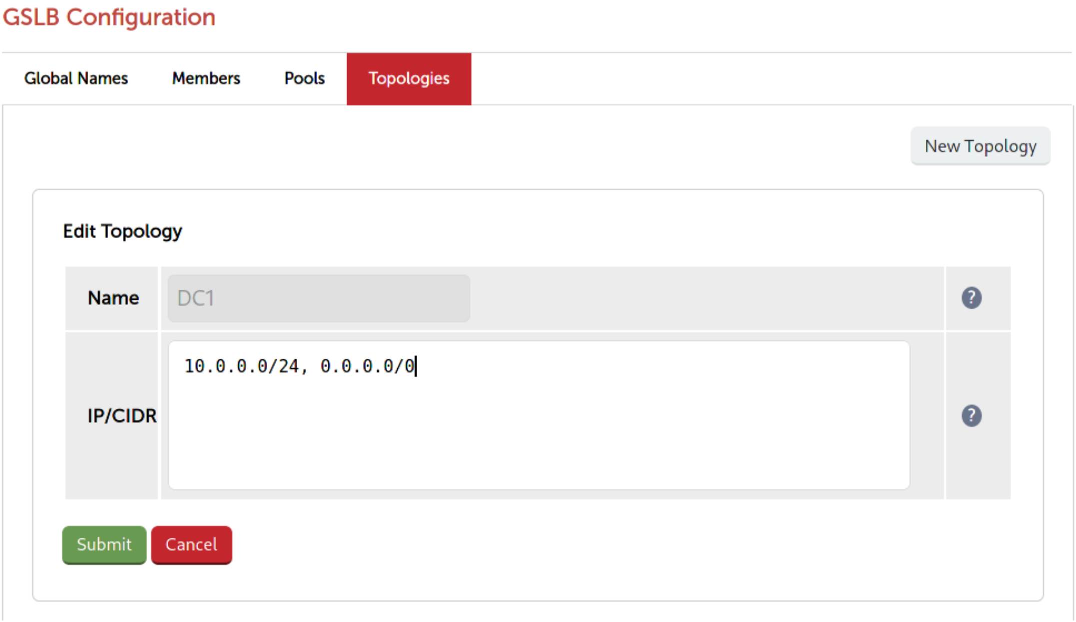
Consider a client on the public internet requesting a resource from the HyperStore cluster. The DNS query associated with the request may be sourced from a previously unseen, unpredictable public IP address. DNS queries from IP addresses that do not fall within the predefined network topology/subnets 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:



Name	DC1
IP/CIDR	10.0.0.0/24, 0.0.0.0/0

11.6. DNS Server Configuration

Once the GSLB service has been configured on the primary 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 section, the DNS server at each site would be configured with a delegation for the domain `gslb.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 Microsoft DNS Server Configuration.

12. Testing & Verification

> **Note** For additional guidance on diagnosing and resolving any issues you may have, please also refer to Diagnostics & Troubleshooting.

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.

13. 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.

14. Further Documentation

For additional information, please refer to the Administration Manual.

15. Appendix

15.1. Microsoft DNS Server Configuration

Once the GSLB service has been fully configured on the primary load balancer at every site, as described in the previous sections, the DNS server at each site must be configured for GSLB.

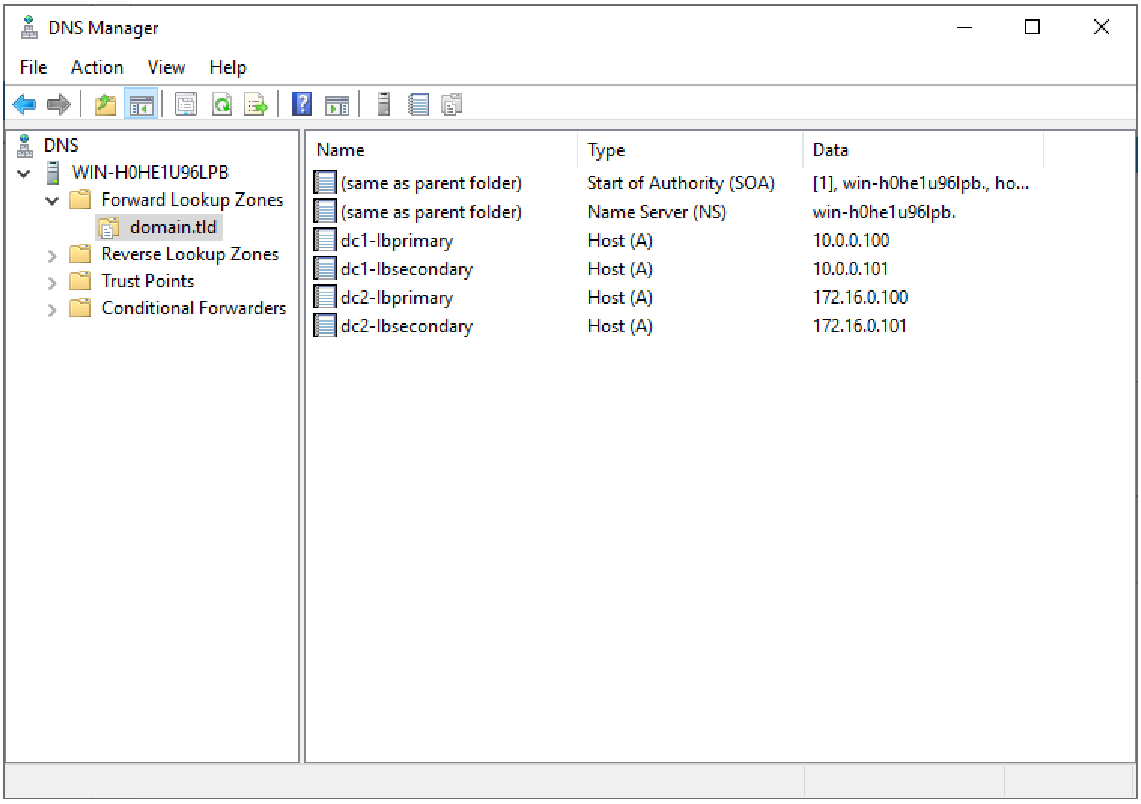
The DNS server at each site must be configured to delegate DNS requests for the subdomain in question to the load balancers; the load balancers' GSLB services will serve the appropriate IP addresses to the DNS servers. Using the example presented throughout this document, the DNS server at each site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

15.1.1. Microsoft DNS Server

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

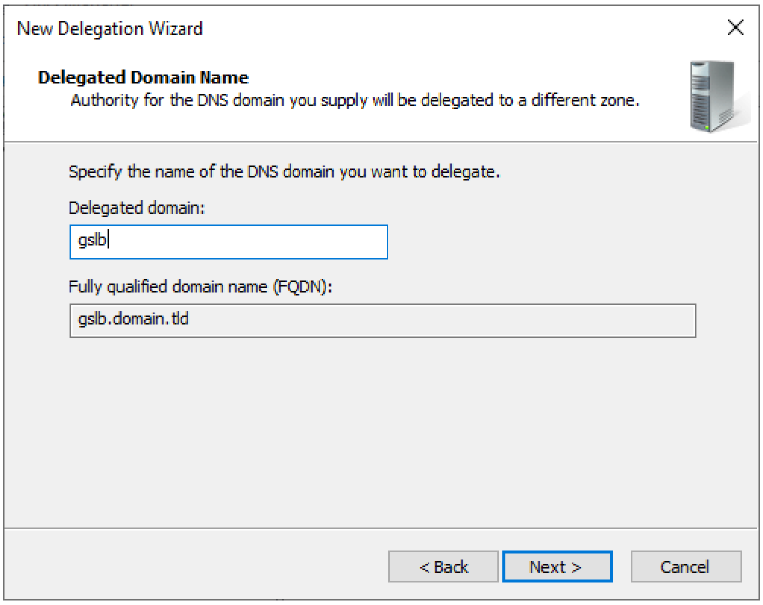
1. Open **DNS Manager** and create A records for every load balancer at every site, using ***Action > New Host*** (e.g. `dc1-lbprimary.domain.tld`, `dc1-lbsecondary.domain.tld`, `dc2-lbprimary.domain.tld`, and `dc2-lbsecondary`).



Name	Type	Data
(same as parent folder)	Start of Authority (SOA)	[1], win-h0he1u96lpb., ho...
(same as parent folder)	Name Server (NS)	win-h0he1u96lpb.
dc1-lbprimary	Host (A)	10.0.0.100
dc1-lbsecondary	Host (A)	10.0.0.101
dc2-lbprimary	Host (A)	172.16.0.100
dc2-lbsecondary	Host (A)	172.16.0.101

2. Provided that the load balancer part of the GSLB configuration has been completed and is working, 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*.



New Delegation Wizard

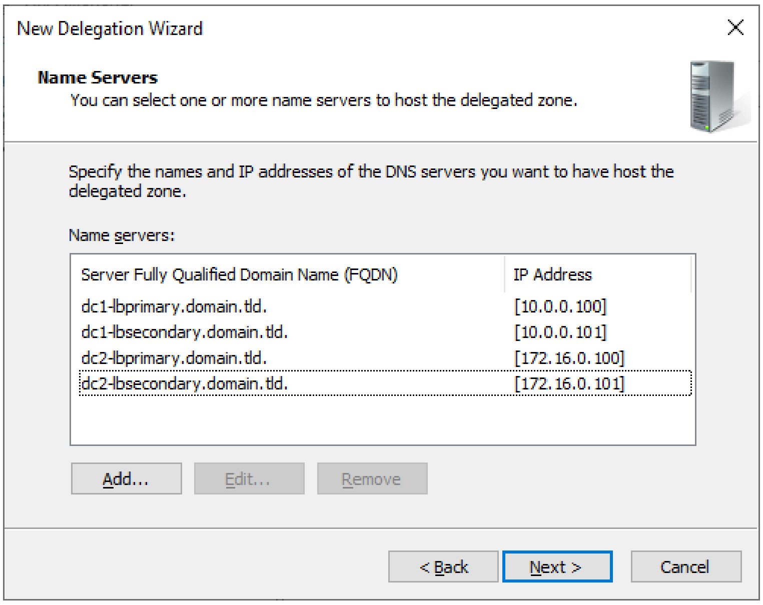
Delegated Domain Name

Authority for the DNS domain you supply will be delegated to a different zone.

Specify the name of the DNS domain you want to delegate.

Delegated domain: gslb

Fully qualified domain name (FQDN): gslb.domain.tld



New Delegation Wizard

Name Servers

You can select one or more name servers to host the delegated zone.

Specify the names and IP addresses of the DNS servers you want to have host the delegated zone.

Name servers:

Server Fully Qualified Domain Name (FQDN)	IP Address
dc1-lbprimary.domain.tld.	[10.0.0.100]
dc1-lbsecondary.domain.tld.	[10.0.0.101]
dc2-lbprimary.domain.tld.	[172.16.0.100]
dc2-lbsecondary.domain.tld.	[172.16.0.101]

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 gslb.domain.tld 10.0.0.50
Server: UnKnown
Address: 10.0.0.50

Name:   gslb.domain.tld
Served by:
```

```
- dc1-lbprimary.domain.tld
        10.0.0.100
        gslb.domain.tld
- dc1-lbsecondary.domain.tld
        10.0.0.101
        gslb.domain.tld
- dc2-lbprimary.domain.tld
        172.16.0.100
        gslb.domain.tld
- dc2-lbsecondary.domain.tld
        172.16.0.101
        gslb.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 sites/services, like so:

```
C:\Users\me>nslookup gslb.domain.tld 10.0.0.50
Server: UnKnown
Address: 10.0.0.50

Non-authoritative answer:
Name:   gslb.domain.tld
Address: 10.0.0.2
```

15.2. 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.

15.2.1. Enabling Proxy Protocol

This is a two step process. The first step is to enable Proxy Protocol for the Hyperstore nodes. The second step is to configure two additional VIPs by duplicating the S3 HTTP & S3 HTTPS VIPs, setting different ports and enabling Proxy Protocol.

Step 1 - 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**.

> ℹ Note
>
> 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'
```

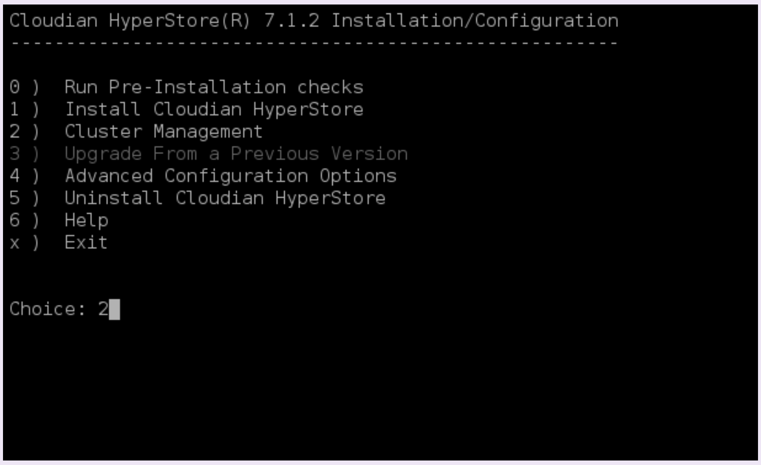
3. Push the configuration change to all HyperStore nodes in the cluster.

Note

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:

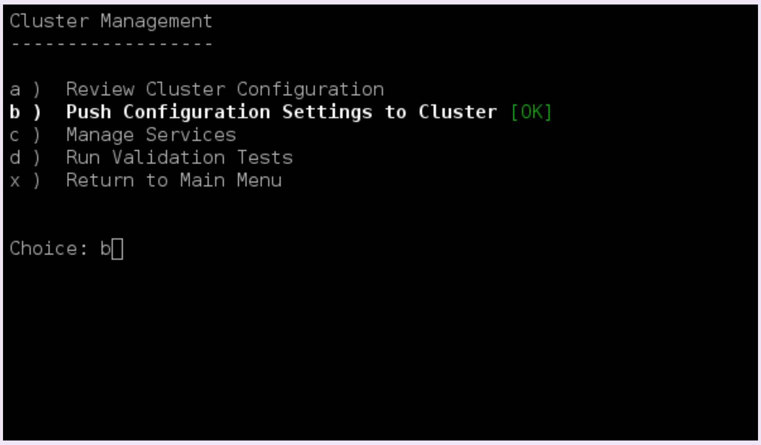
```
./cloudianInstall.sh
```

Using the script, enter **2** for *Cluster Management*:



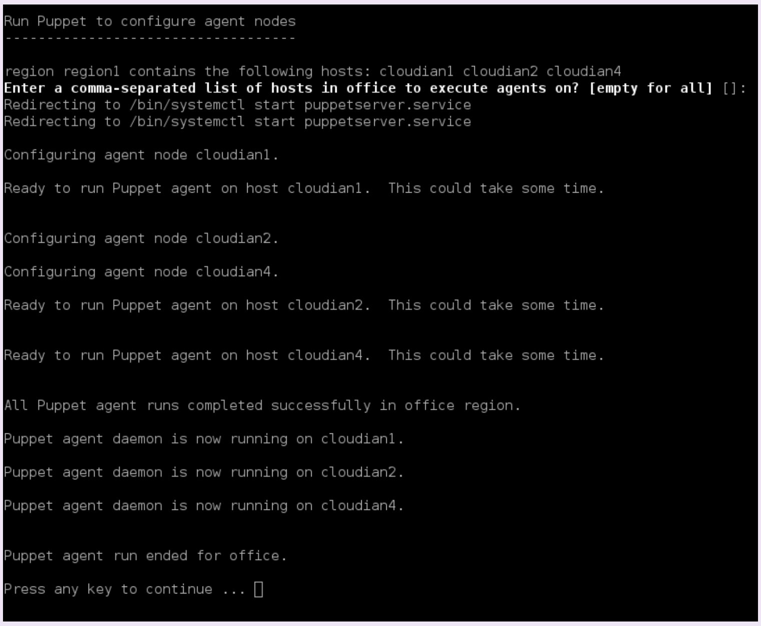
```
Cloudian HyperStore(R) 7.1.2 Installation/Configuration
-------------------------------------------------------

0 )  Run Pre-Installation checks
1 )  Install Cloudian HyperStore
2 )  Cluster Management
3 )  Upgrade From a Previous Version
4 )  Advanced Configuration Options
5 )  Uninstall Cloudian HyperStore
6 )  Help
x )  Exit


Choice: 2
```

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:

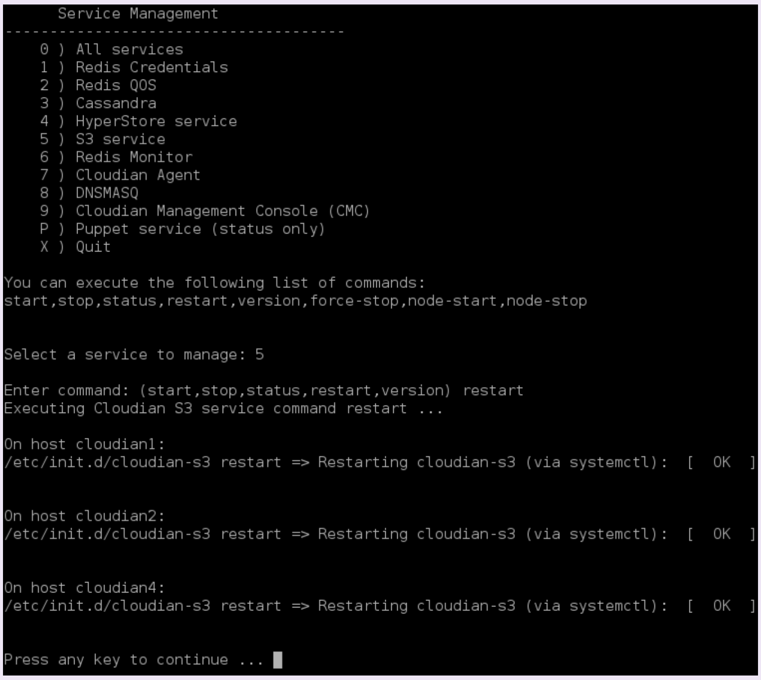


```
Run Puppet to configure agent nodes
-----------------------------------

region region1 contains the following hosts: cloudian1 cloudian2 cloudian4
Enter a comma-separated list of hosts in office to execute agents on? [empty for all] []:
Redirecting to /bin/systemctl start puppetserver.service
Redirecting to /bin/systemctl start 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 cloudian4.

Ready to run Puppet agent on host cloudian2.  This could take some time.


Ready to run Puppet agent on host cloudian4.  This could take some time.


All Puppet agent runs 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 cloudian4.


Puppet agent run ended for office.

Press any key to continue ... 
```

4. Restart the S3 service to apply the new configuration across all nodes.

Note

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:



```
      Service Management
--------------------------------------
    0 ) All services
    1 ) Redis Credentials
    2 ) Redis QOS
    3 ) Cassandra
    4 ) HyperStore service
    5 ) S3 service
    6 ) Redis Monitor
    7 ) Cloudian Agent
    8 ) DNSMASQ
    9 ) Cloudian Management Console (CMC)
    P ) Puppet service (status only)
    X ) Quit

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 cloudian4:
/etc/init.d/cloudian-s3 restart => Restarting cloudian-s3 (via systemctl):  [  OK  ]


Press any key to continue ... 
```

Step 2 - Appliance Configuration

Configuring Additional VIP 2 – 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 3 – 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 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**.

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

15.2.2. Filtering the Cloudian Request Log

The following command can be used on the Cloudian nodes to view the source IP of the requests:

```
tail -f /var/log/cloudian/cloudian-request-info.log | cut -d "|" -f 2
```

This is useful when checking that Proxy Protocol has been enabled correctly.

15.3. 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.

> ℹ 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.

15.3.1. 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:

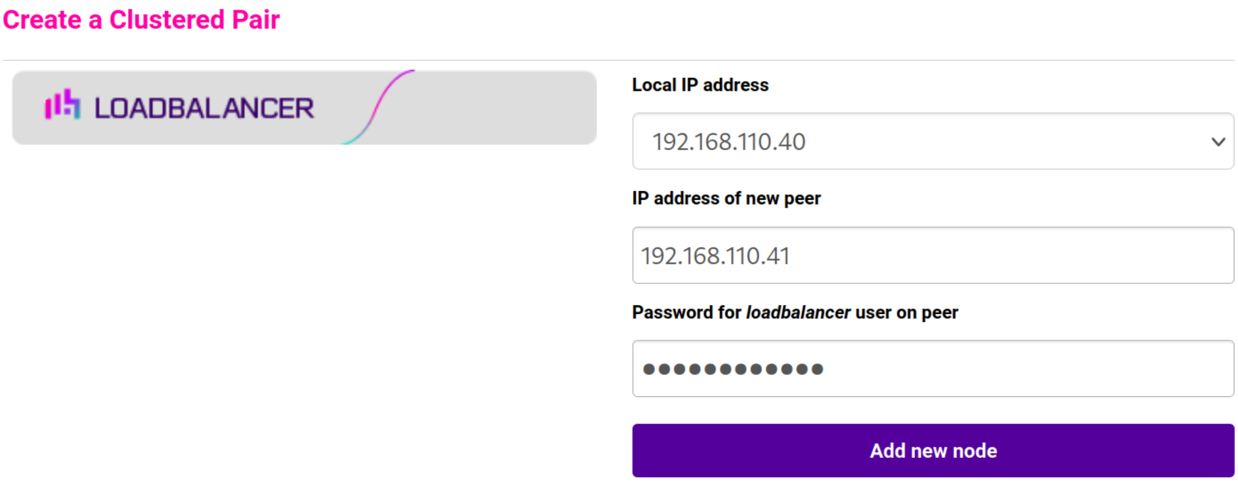
WebUI Main Menu Option	Sub Menu Option	Description
Local Configuration	Hostname & DNS	Hostname and DNS settings
Local Configuration	Network Interface Configuration	Interface IP addresses, bonding configuration and VLANs
Local Configuration	Routing	Default gateways and static routes
Local Configuration	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.

15.3.2. Configuring the HA Clustered Pair

> 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*.



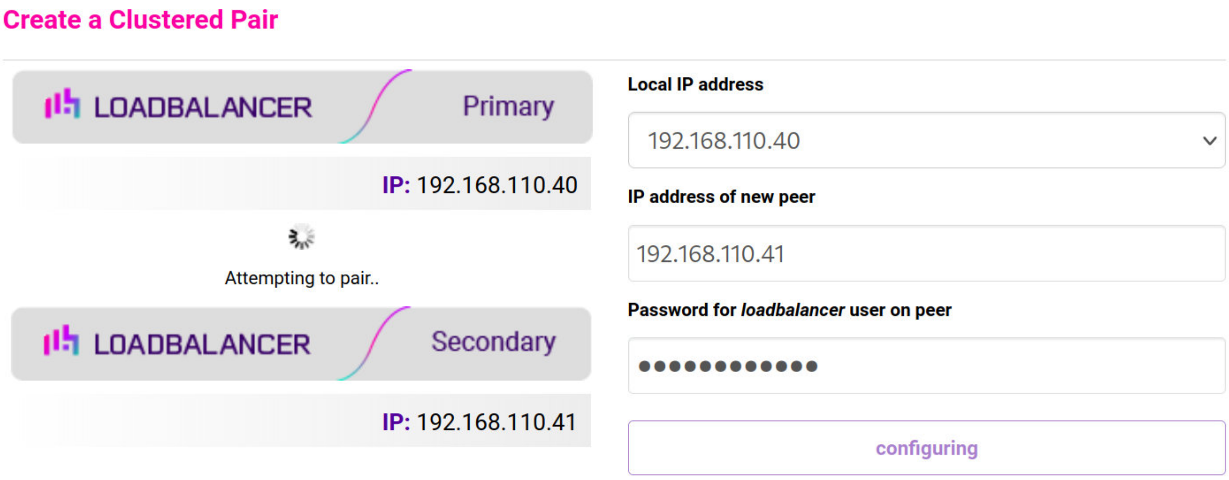
Create a Clustered Pair

Local IP address: 192.168.110.40

IP address of new peer: 192.168.110.41

Password for *loadbalancer* user on peer: ●●●●●●●●●●●●

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

Primary IP: 192.168.110.40

Attempting to pair..

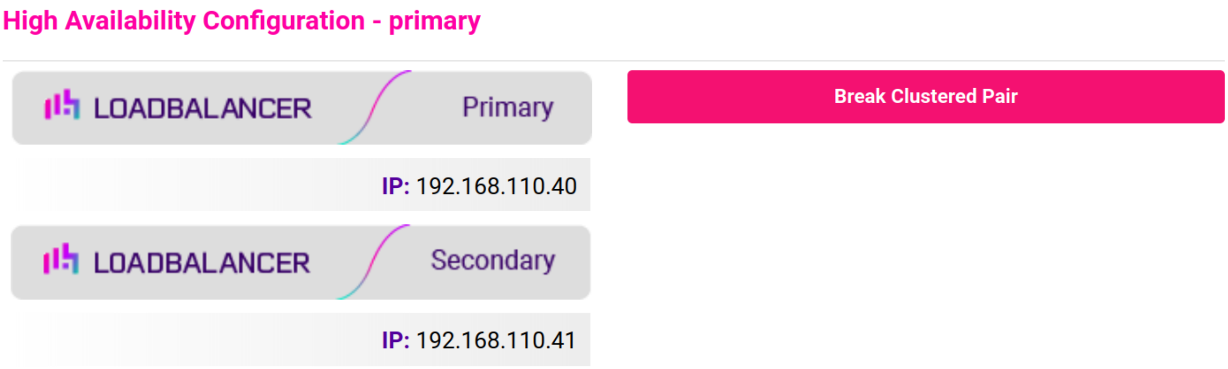
Secondary IP: 192.168.110.41

Local IP address: 192.168.110.40

IP address of new peer: 192.168.110.41

Password for *loadbalancer* user on peer: ●●●●●●●●●●●●

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


High Availability Configuration - primary

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.

> ℹ **Note** Clicking the **Restart Heartbeat** button on the Primary appliance will also automatically restart heartbeat on the Secondary appliance.

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

> ℹ **Note** For details on testing and verifying HA, please refer to Clustered Pair Diagnostics.

16. Document Revision History

Version	Date	Change	Reason for Change	Changed By
1.0.0	4 April 2018	Initial version		AH
1.1.0	7 November 2018	Changed the "Health Checks" section to talk about the new built-in health checks and the rewritten custom health check 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 Changed the VIP configuration instructions to refer to the new built-in health checks	Required updates	AH
1.1.1	6 December 2018	Added the new "Company Contact Information" page	Required updates	AH
1.1.2	18 December 2018	Modified the 'Loadbalancer.org Appliances Supported' section to state that only the Enterprise 10G, Enterprise 40G, and Enterprise VA MAX models are supported	Required updates	AH
1.1.3	31 January 2019	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	Required updates	AH

Version	Date	Change	Reason for Change	Changed By
1.2	3 April 2019	Changed Loadbalancer.org software versions supported to V8.3.6 and later, to account for the new authenticated health check in the WebUI Replaced the explanatory note regarding persistence on the CMC VIP with an explicit instruction to set the persistence option to HTTP cookie 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 Added credentials for the health check for the API VIP Changed the "Health Checks" section to reflect the changes to health checks Removed the section referencing the custom health check Removed the appendix containing the custom health check Added a section on "Performance and Sizing" explaining the need for virtual appliances to be assigned a minimum of 4 vCPUs Added instructions for configuring and enabling multithreaded HAProxy	Required updates	AH
1.2.1	4 April 2019	Corrected the persistence mode for the CMC VIP to source IP, as it is a 'TCP mode' VIP Removed the note for the CMC VIP that explained the rationale for checking against the TLS port, for simplicity	Required updates	AH
1.2.2	25 April 2019	Added additional screenshots regarding the health checks, for clarity Added notes regarding setting the health check host header if needed	Required updates	AH

Version	Date	Change	Reason for Change	Changed By
1.2.3	4 June 2019	Changed the health check for the Cloudian Management Console VIP, based on updated documentation from Cloudian	Required updates	AH
1.2.4	5 July 2019	Changed the health check 'Request to send' fields from .healthCheck to /.healthCheck, and updated screen shots accordingly	Required updates	AH
1.3.0	27 August 2019	Styling and layout Changed the health check for the Cloudian Management Console VIP to use an OPTIONS check	General styling updates Required updates	AH

Version	Date	Change	Reason for Change	Changed By
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

Version	Date	Change	Reason for Change	Changed By
1.5.0	No public release	Minor visual updates to multi-site GSLB diagram Important content changes and additions regarding how GSLB is configured and deployed New title page Updated Canadian contact details	Required technical changes Branding update Change to Canadian contact details	AH
2.0.0	15 January 2021	Complete overhaul of the document Document restructured with several sections added and some removed 'Direct to node' deployment type added from a previously separate internal document New advice added regarding the minimum viable node count for the 'Intelligent Site Health Check'	Required technical changes	AH
2.0.1	29 January 2021	Corrected screenshot for IAMS virtual service	Required correction	AH
2.0.2	11 February 2021	Added admin API credential information Added warnings to prevent 'duplicate service' related configuration errors from propagating throughout an entire deployment	Required updates requested by Cloudian	AH
2.1.0	29 April 2021	Converted the document to AsciiDoc	Move to new documentation system	AH
2.1.1	5 July 2021	Amended the ‘Handling Multiple Subdomains’ sections and added an additional example	Technical content improvement and clarification	AH
2.1.2	21 March 2022	Added new multithreading advice	Product change means multithreading is now enabled by default	AH

Version	Date	Change	Reason for Change	Changed By
2.1.3	6 April 2022	Updated DNS server configuration instructions Amended GSLB set up instructions	Changed to use new, consistent common component GSLB updates across all documentation	AH
2.1.4	28 September 2022	Updated layer 7 VIP and RIP creation screenshots	Reflect changes in the web user interface	AH
2.1.5	5 January 2023	Combined software version information into one section Added one level of section numbering Added software update instructions Added table of ports used by the appliance Reworded 'Further Documentation' section Removed references to the colour of certain UI elements	Housekeeping across all documentation	AH
2.2.0	10 January 2023	Completely removed layer 4 DR mode as a deployment option Changed IAM services health checks to copy the S3 services health checks	Changes requested by Cloudian	AH
2.2.1	2 February 2023	Updated screenshots	Branding update	AH
2.2.2	7 March 2023	Removed conclusion section	Updates across all documentation	AH
2.3.0	24 March 2023	New document theme Modified diagram colours	Branding update	AH
2.3.1	29 June 2023	Updated multithreading advice	New default option in the web user interface	AH

Version	Date	Change	Reason for Change	Changed By
2.3.2	26 March 2024	Updated the 'Using the PROXY Protocol to Retain Client IP Addresses' section to make the required steps clearer Added command for filtering the Cloudian Request log	Technical content improvement	RJC

LOADBALANCER

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

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.