EDGE NEXUS

EdgeGSLB

Administrator User Guide

Document Properties

Document Number: 2.0.10.21.22.14.10

Document Creation Date: 5 August 2021

Document Last Edited: 21 October 2022

Document Author: Jay Savoor

Document Last Edited by:

Document Disclaimer

This manual's screenshots and graphics may differ slightly from your product due to differences in product release. Edgenexus ensures that they make every reasonable effort to ensure that the information in this document is complete and accurate. Edgenexus makes changes and corrections to the information in this document in future releases when the need arises. Edgenexus assumes no liability for any errors.

Copyrights

Trademarks

Edgenexus logo, Edgenexus, EdgeADC, EdgeWAF, EdgeGSLB, EdgeDNS are all Edgenexus Limited's trademarks. All other trademarks are the properties of their respective owners and are acknowledged.

Edgenexus Support

If you have any technical questions regarding this product, please raise a support ticket at: support@edgenexus.io

Table of Contents

GSLB Explained

GSLB, or its full name Global Server Load Balancing, allows organizations to improve their IT resilience using multiple data centres, on-ground or in-cloud. EdgeGSLB works by using server health and DNS to direct the traffic across geographical sets based on logic defined by the networking administrator.

The great thing about EdgeGSLB is that its resilience is no-touch, meaning that failover, failback, or redirection of traffic is seamless should any resources within a data centre fail.

EdgeGSLB can also drive traffic based on geo-proximity, so users can obtain the lowest possible latency when accessing the applications within the data centre. Geo-proximity means that Melbourne users are sent to an Australian data centre, while users in Frankfurt can be sent to their nearest location, London. If the Australian data centre fails, its users will automatically be diverted to the London data centre until fixes are made.

The advantage of using the EdgeGSLB is that it is totally under your control and does not rely on external DNS or other contractual services.

EdgeGLSB – The mechanics

There are some use cases for EdgeGSLB, and in this document, we will work to explain these to you.

Geographical Active-Active

This illustration is the classic way in which most organizations use EdgeGSLB. It operates by forming an Active-Active high availability framework of data centre server farms. Users can be located across geographies, and their requests are sent to the nearest and available data centre to have the best user experience.
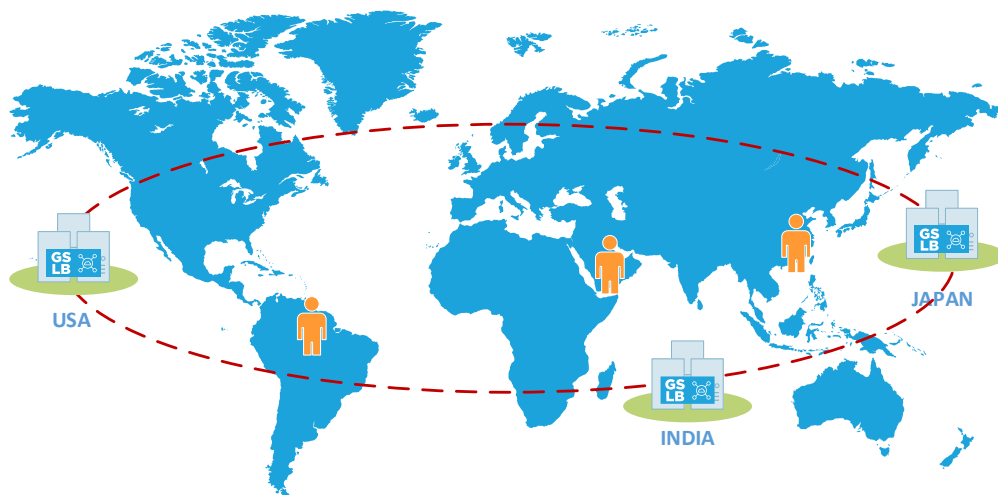
End users from across the globe will access their application, with the EdgeGSLB determining where best to route the requests. The 'where best' is decided by the EdgeGSLB based on the system administrator's specified geo-proximity configuration.

The health of each server within each data centre is continually monitored using various monitoring methods available. If an anomaly is detected within any data centre, the EdgeGSLB in that location will automatically route the end user to one of the other available and working server farms. We call this an Active-Active configuration because all the EdgeGSLB modules, servers, and data centres are accessible and operational.

Effective Disaster Recovery (DR)

The Edgenexus GSLB solution can be deployed to form a highly effective disaster recovery solution.

In standard DR scenarios, failure of the primary site would mean that the DNS entries would need to be manually changed to point to the secondary DR location and then back again when failback is initiated.

In the case of a data centre or services failure in the primary location, the EdgeGSLB will initiate a failover to the secondary. Using Edgenexus GSLB, this is no longer required. The image below shows two data centre locations, with an ADC+EdgeGSLB in each and a continual health check on their respective server farms.
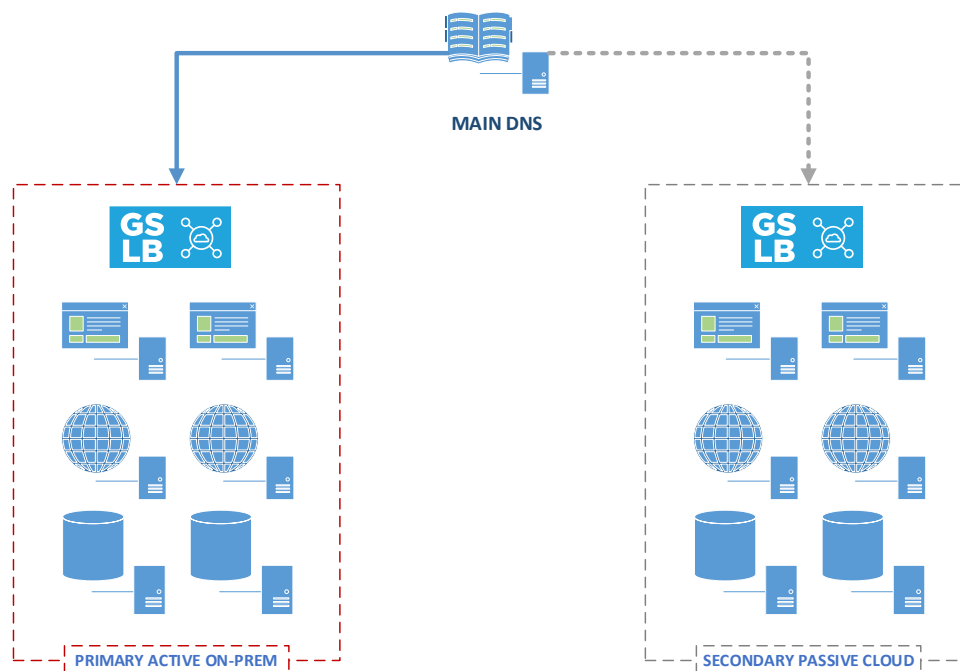
Should the EdgeGSLB instances determine that the health of the primary data centre or any of its applications poses a problem, the failover to the secondary will be initiated.

Zero-Touch Scalability

Another good use case of the EdgeGSLB is 'zero-touch scalability.

Let us look at a scenario with a primary data centre on-premise. We currently have several application servers in a cloud data centre, and we use GSLB for high availability. We then experience a sudden upsurge in usage and risk our servers being stressed.

The EdgeGSLB can monitor the usage in the primary on-premise servers and redirect users to the secondary cloud-located servers when needed. Since cloud data centres can utilize elastic computing, this method proves to be highly cost-effective to use.
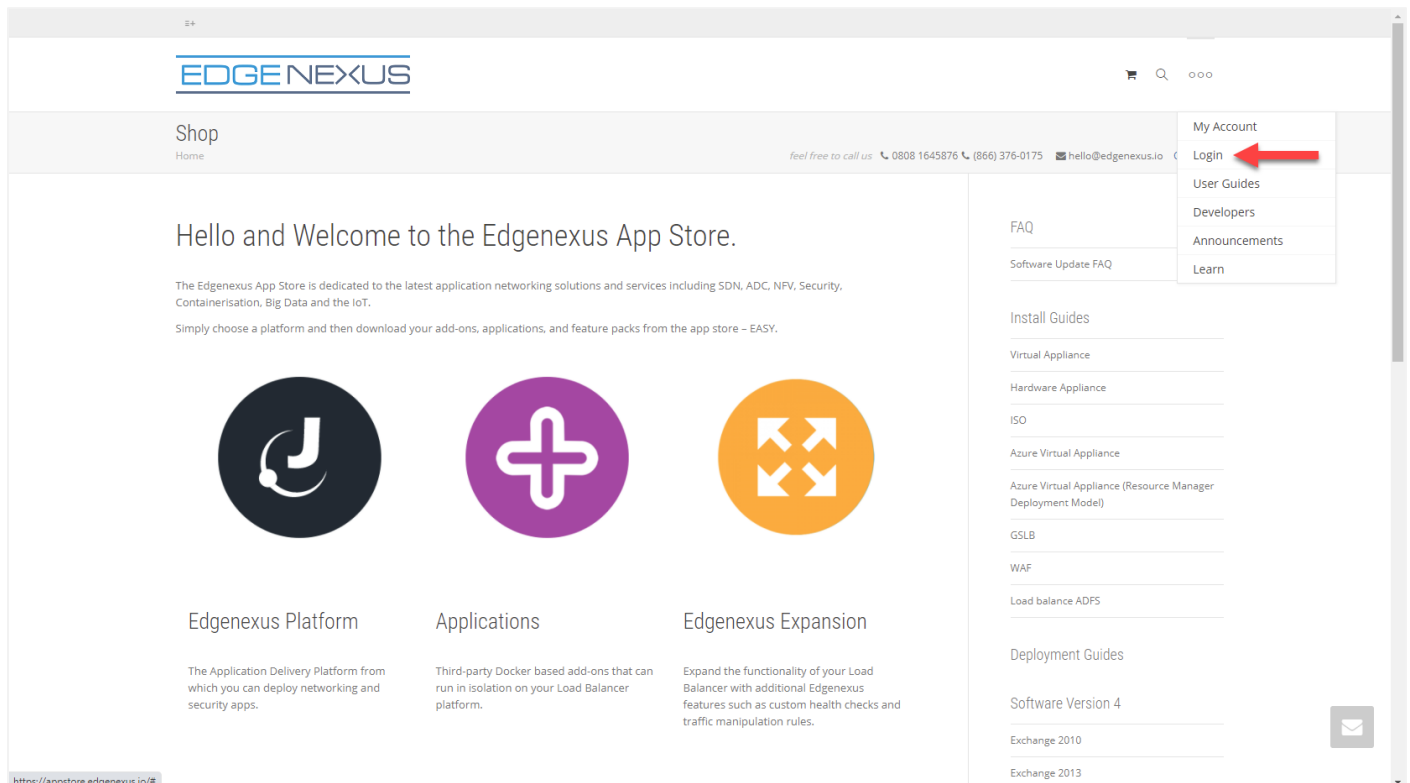
Installing the EdgeGSLB App

Getting the EdgeGSLB from the App Store

Obtaining the EdgeGSLB is very easy.

As with every Edgenexus App, the EdgeGSLB App is available through the App Store and is free of cost.

- The first thing to do is to register for access to the Edgenexus App Store. This process is done by using a browser and navigating to https://appstore.edgenexus.io.
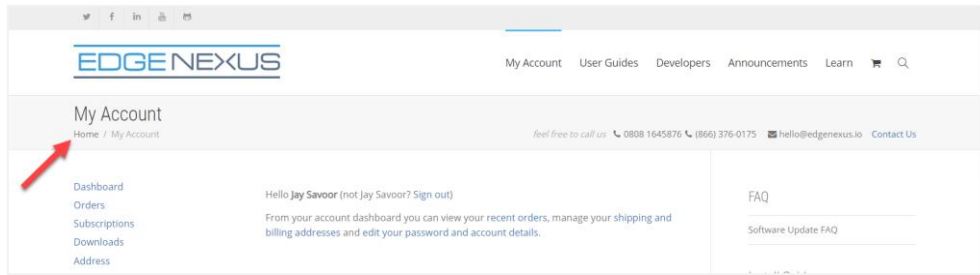

- Click on the login link in the hamburger icon at the top right.
- Click on the Create an Account, or log in using your account credentials.
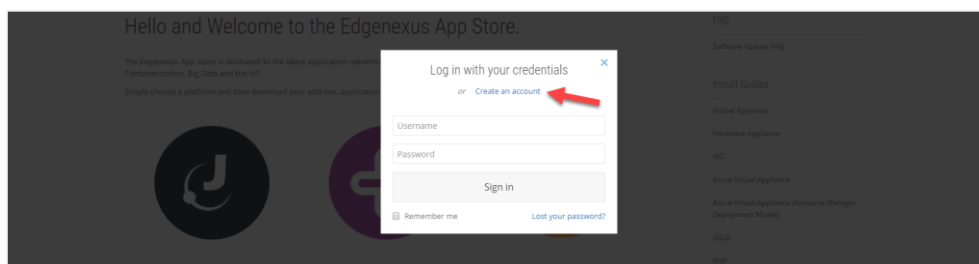
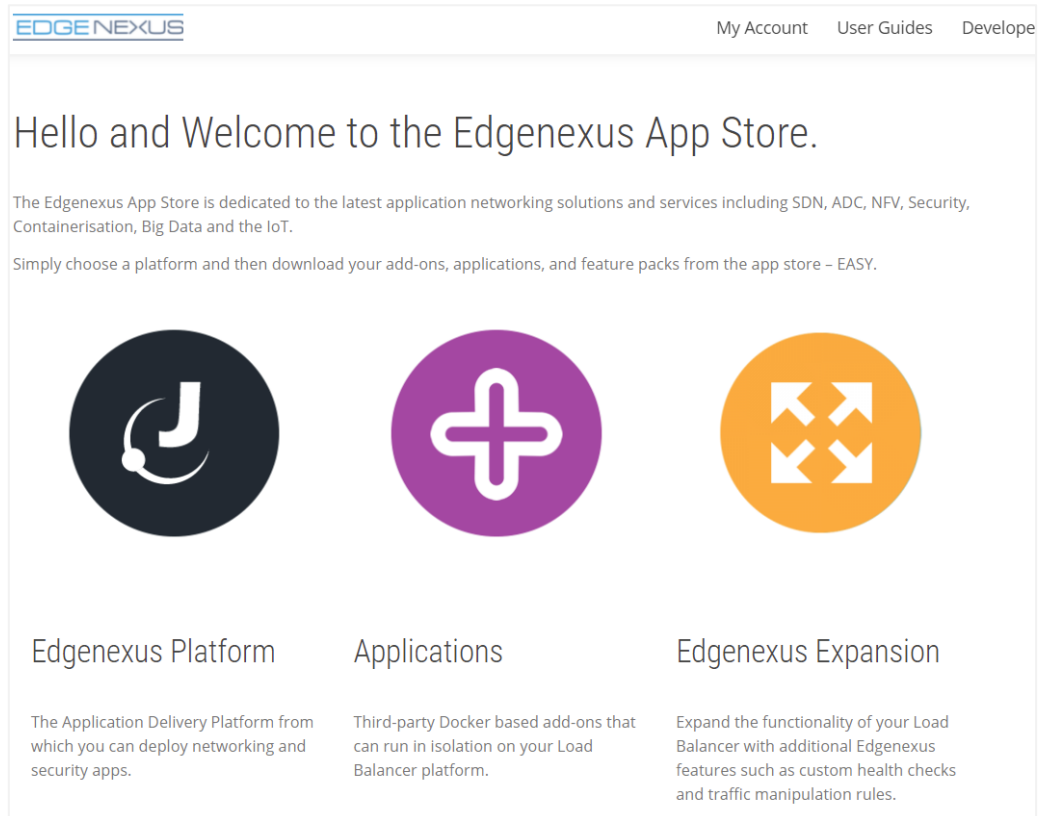
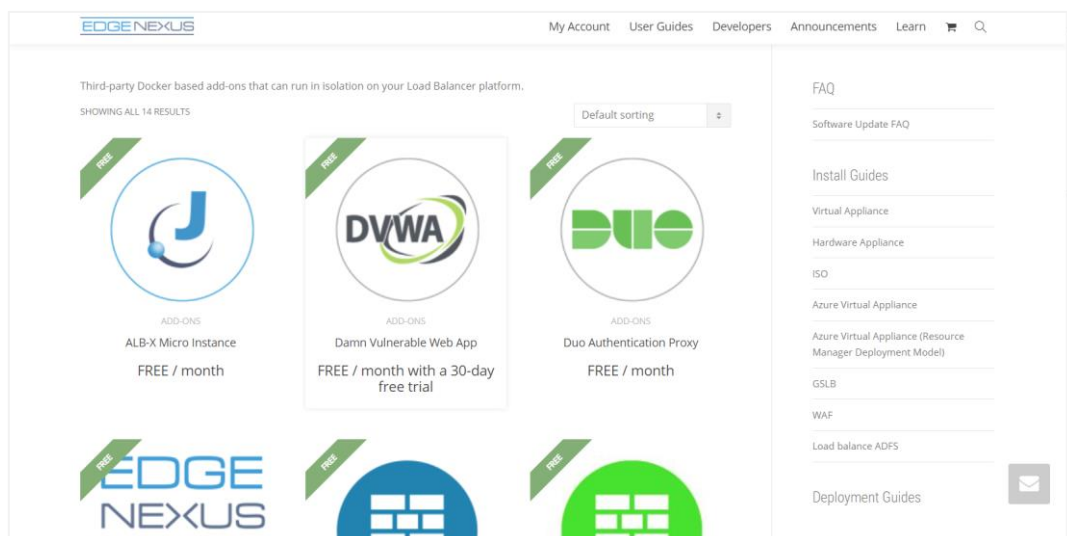- Once you have logged in, please click on the Home link under the logo.

- Next, click on Applications.

- This action will take you to the Applications page, where you can download the EdgeGSLB.

- Within the applications page, you can browse for and order the App.
- The EdgeGSLB is free to try, but you purchase it for full use.
- At this point, you have two options: Using the App Store from within the EdgeADC or directly downloading the App from the App Store and then uploading it to the EdgeADC.

Downloading and importing the App using the EdgeADC

- The first option is to log in using your App Store credentials inside the EdgeADC. The integrated App Store interface is available using Services > App Store.
- This method will allow you to make the purchase, and then find it available within the Purchased Apps section in Library > Apps.
- EdgeGSLB App looks something like the one shown below.
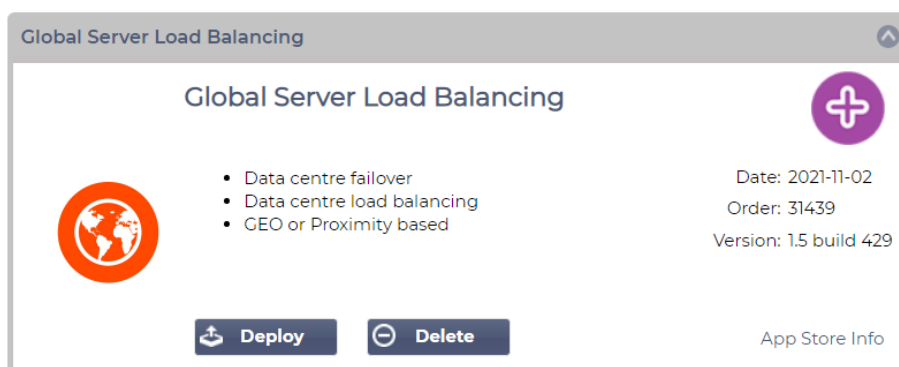
- You can then choose to download the App, and it will appear in the Downloaded Apps section.
- From the Library > Apps > Downloaded Apps section, locate the EdgeGSLB App and then deploy it to the EdgeADC containers by clicking the Deploy button.
- Once deployed, it will be available in the Library > Add-Ons tab

Download and import the App using direct download

- The secondary method uses your App Store login and directly downloads it to your desktop using a browser.
- Once downloaded, please make sure you save it without altering the filename.
- Please also ensure that there is no (1) or something similar in the filename, possibly indicating a second download, etc.
- With the file downloaded, navigate to Advanced > Software of the EdgeADC GUI using your browser.
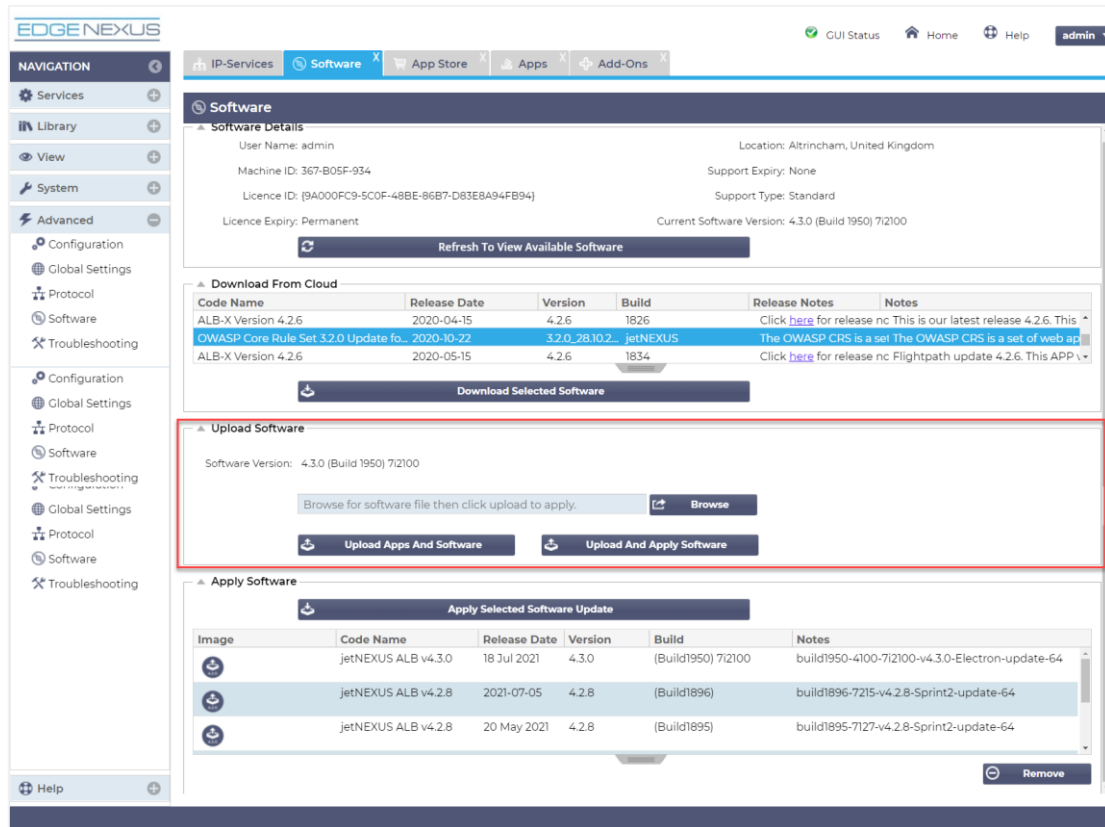
- There are several sections within the Software page, but the one we need is Upload Software.
- First, click the Browse button and find the EdgeGSLB App you downloaded.
- Next, click the Upload and Apply Software.
- The App will be shown in the Downloaded Apps section of Library > Add-Ons.
- From the Library > Apps > Downloaded Apps section, locate the EdgeGSLB App and then deploy it to the EdgeADC containers by clicking the Deploy button.
- Once deployed, it will be available in the Library > Add-Ons tab

Making the EdgeGSB App operational

When an App is downloaded and deployed, it is yet to be operational. It has to be given an IP address in the same subnet as the EdgeADC and ports through which it needs to be accessible.

- Navigate to Library > Add-Ons and locate the EdgeGSLB App.
- It should look something like the image below.

- As shown in the Container Name and External IP field, no name or IP address is allocated.
- Add an appropriate static IP address. This entry is optional for EdgeADC v4.3.x and above but is mandatory for any version lower than 4.3.x.
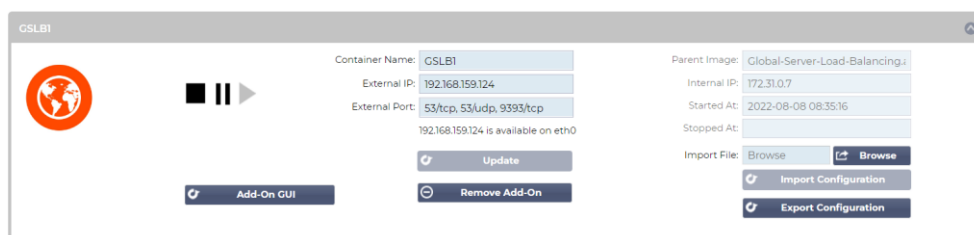- Next, give the App a name – the EdgeADC's internal DNS system uses this to refer to the App when needed.

- You will need to add the relevant ports for DNS. These are 53/tcp, 53/udp and 9393/tcp.
- Once you have done this, click the Update button to initialize the App.
- It should look something like the one below.
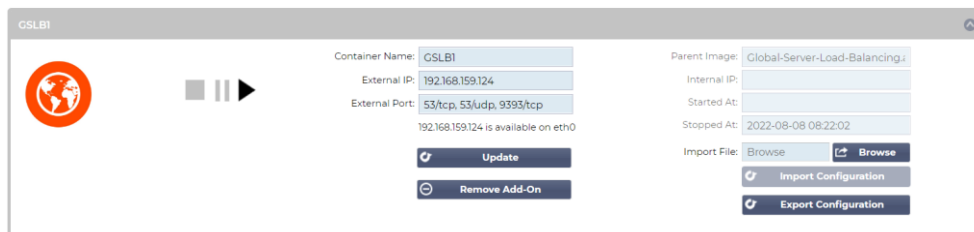
- Click the PLAY icon to activate the App into an operational state.

- You can launch the App GUI using View App or the listing in the IP Services section.

The EdgeGSLB App runs within the ADC's docker container technology, ensuring its safety and integrity. The App uses a separate docker0 network to communicate with the EdgeADC load balancer. When the App is started, it is allocated an IP address from the docker0 pool. This IP address is automatically resolved by the EdgeADC using the docker name you provided in the Container Name field. You can see the internal IP address on the right side of the App.

You will also need to ensure that the EdgeADC can resolve DNS names using the GSLB's PowerDNS module. The External IP address of our EdgeGSLB App, 192.168.159.124 in our example shown above, needs to be added as the Primary DNS in the System > Networking section. See the example below.

Configuring the EdgeGSLB

Example network

In this example, we are using a single ADC-based GSLB topology. We will deal with clustered Adc topologies later in this guide. See the network example below.
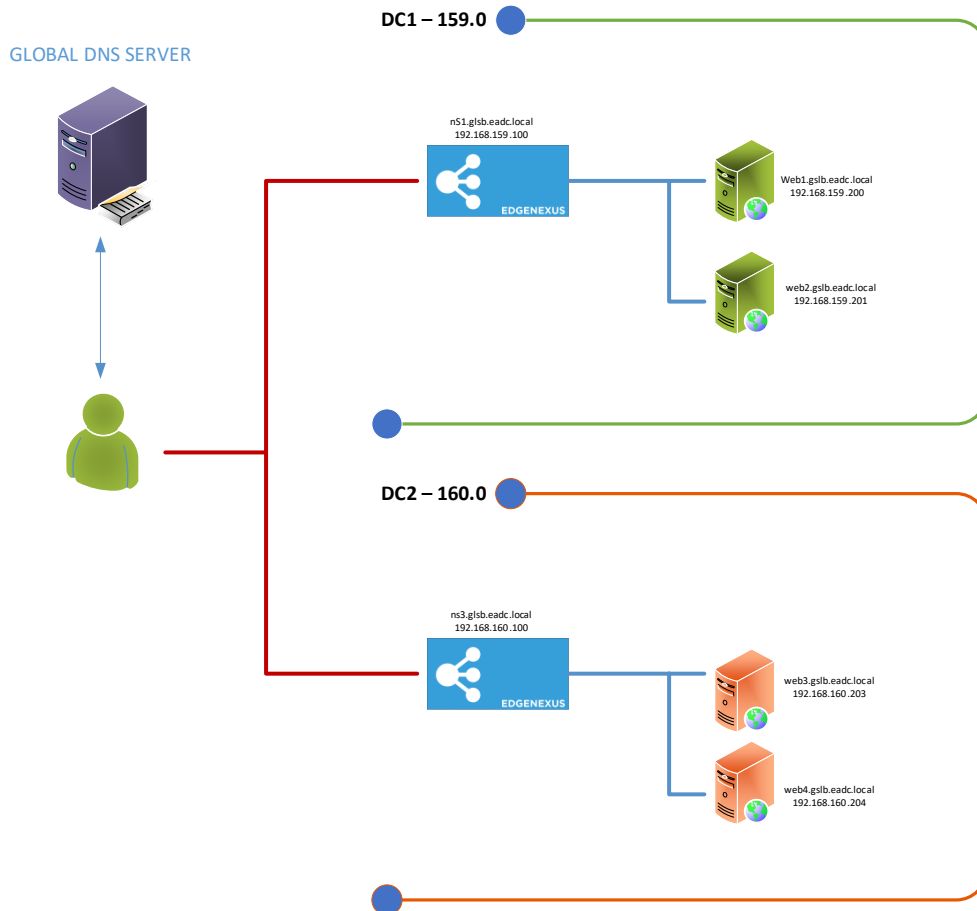
Configuring your EdgeGSLB

You will need to use your browser to access the EdgeGSLB App for the configuration needed.

To access the App, use the Add-on GUI button. You can also access this using the browser and the following URL, https://{EXTERNAL_IP_ADDRESS}:9393.

This action will open a browser window or tab and display the page below.

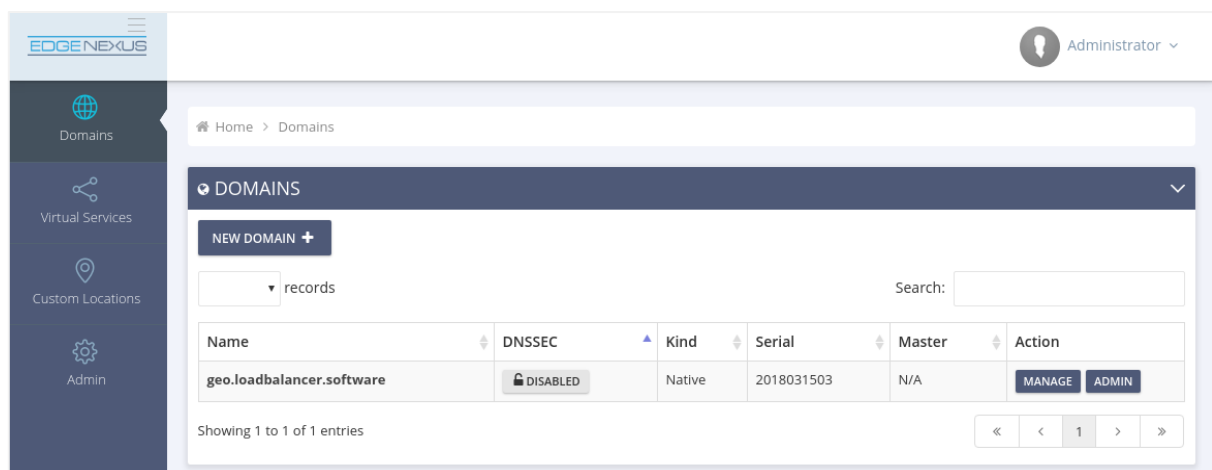The default username and password are *admin* and *jetnexus*, the same as the EdgeADC. You can change this once you have logged in.

Once logged in, you will be presented with a page similar to the one shown below.

As you can see, a dummy domain name has been preconfigured as an illustration.

Note: It is impossible to edit a domain entry once it has been created. You can only delete the domain entry.

This guide will show you how to configure the GSLB using the dummy configuration and will allow you to get a good idea of how to use your domain and VIP entries when configuring it for POC or production.
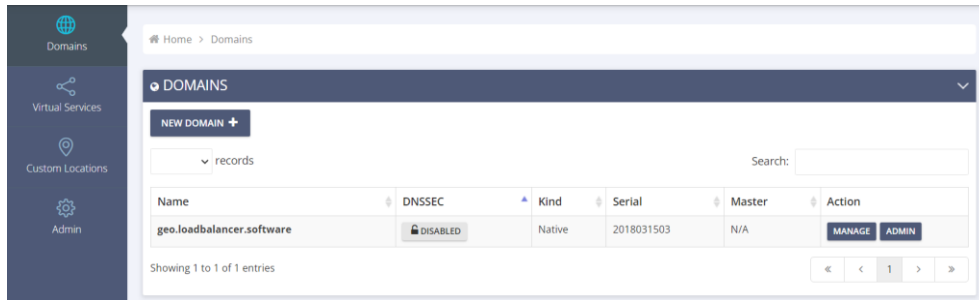
Adding a new domain

The domain name (in our example, eadc.local) is entered in this stage will be the one that the GSLB uses for resolving. You may opt to use the prefix 'gslb' in front of your domain name as a subdomain, as we have in this example, and it will then be 'gslb.eadc.local'.

- Click the New Domain button located in the Domains section.

- You will then see the form shown below.
- Enter your domain name in the field shown in the example.

- Leave the TYPE settings in the form as they are if this is a single ADC GSLB. However, if you use clustered ADCs, you must follow the directions in the GSLB Replication section.

You should now see the following in the Domains section.

- Click on the newly created gslb.eadc.local entry.

You will see that you have a preconfigured SOA record and the ability to enter A records required for proper operation.

You will need to enter A records for the Real Servers used for the Virtual Services and the IP addresses for the NS records, which will correspond to the VIPs that point to the GSLB.

In our example, we have the records as per the image below.

In our example, please observe the following:

- The SOA record has been changed to assign ns1.gslb.eadc.local as the primary name server.
- We have added the IP addresses for ns1 and ns2, where ns1 is in datacenter 1, and ns2 is in datacenter 2.
- We also the A records for the real servers – we will have similar records on the second load balancer in datacenter 2.

We will now switch back to the IP Services tab of the ADC, where some VIP entries need to be made.

We will make three entries that correspond to the required Virtual services.

1. GSLB DNS VIP – This VIP is used to access the DNS server integrated into the GSLB App. The VIP needs to access DNS using port 53 TCP and UDP.


2. GSLB UI VIP – This VIP uses port 80 and HTTPS services to access the admin user interface of the GSLB App. You may wish to make this VIP accessible from specific source IP ranges using flightPATH traffic rules for security measures.
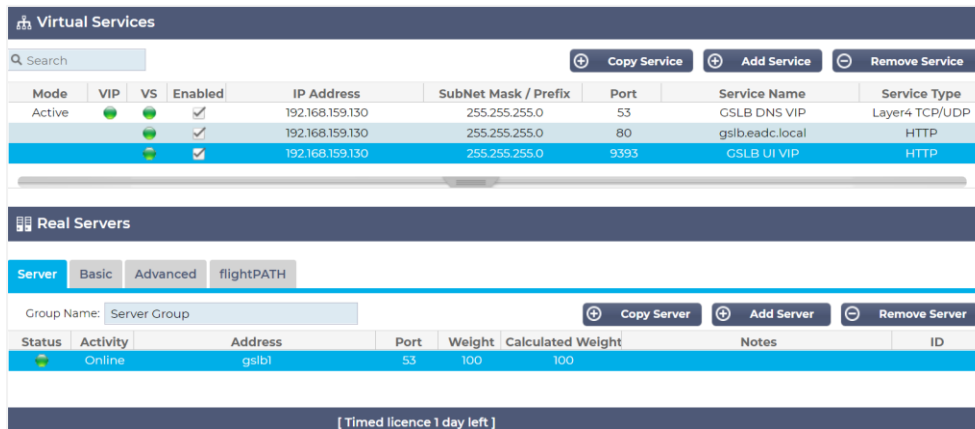
3. GSLB Services Health VIP – This VIP is critical to the operation of the GSLB as it uses health checks that you specify to determine the health of the real servers. The GSLB module uses FQDNs to access the real servers, as seen in our example.
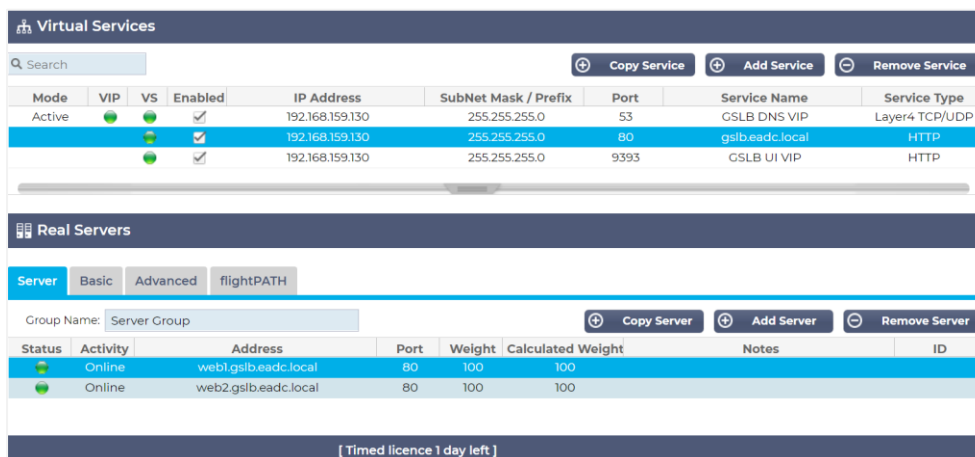
There are some important points to note when configuring the GSLB.

1. Virtual Service Name – This represents what the CNAME entry in the upstream DNS server will be pointing to resolve the IP address for the service the client is trying to access. So in our example, this will be *gslb.eadc.local*.

2. Address entries in the Real Server section are required as the EdgeADC looks up the IP addresses relating to the hostnames configured. It does this by referring to the GSLB configured as the ADC's primary DNS server. The ADC will perform the health check configured in the Real Servers Basic tab to confirm the reachability of the returned server IP and port.
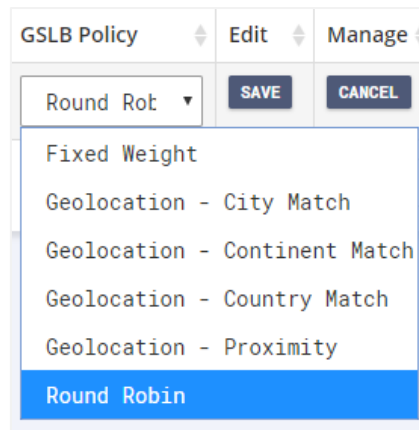
The information you enter here will be scraped by the GSLB using the EdgeADC REST API and entered in the GSLB Virtual Services section. The image below shows the Virtual Services section of the GSLB once the data has been transferred.
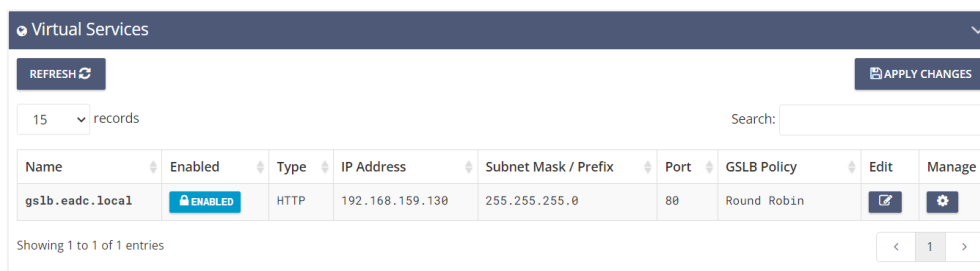
GSLB Policies

The default GSLB policy created for the Virtual Service is Round Robin, as this is the defacto one used by DNS. As the name suggests, any queries made to the domain name will be cycled through the available hosts/IP addresses.

Clicking on the entry will display a dropdown menu, as shown below. The options available are different GSLB policies that the GSLB App supports.

Fixed Weight

When Fixed Weight is selected as the GSLB policy, the ADC and GSLB will work in tandem to ensure that the requests are distributed per the weighting provided.

Geo Location Policies

As the name suggests, these policies are all related to the client's geographical location. Depending on the source IP location, the GSLB and the EdgeADC will decide to which data centre the client requests will be sent.

Custom Locations

The EdgeGSLB can also be configured to use custom locations so that you can use it on internal "private" networks. In the scenario above, the GSLB determines the client's location by cross-referencing the client's public IP address with a database to work out its location. It also works out the service IP address location from the same database; if the load balancing policy is set to a GEO policy, it will return the closest IP address. This method works perfectly well with public IP addresses, but there is no such database for internal private addresses that conform to RFC 1918 for IPv4 addresses and RFC 4193 for IPv6 addresses.

Please see the Wikipedia page explaining private addressing https://en.wikipedia.org/wiki/Private_network

How the Custom Locations feature works

Typically the idea behind using our GSLB for internal networks is that users from certain addresses will receive a different answer for a service depending on which network they are located in, so if we consider two datacentres, one called "North" and the other called "South", providing a service called north.service1.gslb.com and south.service1.gslb.com respectively. When a user from the Northern data centre queries the GSLB, we want the GSLB to respond with the IP address associated with north.service1.gslb.com, provided the service is working correctly. Alternatively, if a user from the Southern data centre queries the GSLB, we want the GSLB to respond with the IP address associated with south.service1.gslb.com again, providing the service is working properly.

So what do we need to do to make the above scenario work?

1. We need to have at least two Custom Locations, one for each data centre
2. Assign the various private networks to these locations
3. Assign each service to the respective location

How do we configure this scenario on the GSLB?

1. Add a location for the Northern Data Center
   - Click on Custom Locations on the left-hand side
   - Click Add Location
   - Name = North
   - Add a private IP address and subnet mask for your Northern network. For this exercise, we will assume that the service and the client IP addresses are in the same private network, 10.1.1.0/24
   - Continent Code = EU
   - Country Code = UK
   - City = Enfield
   - Latitude (obtained from Google) = 51.6523
   - Longitude (obtained from Google) = 0.0807

Note: Please use the correct code, which can be obtained here

Add a location for the Southern Data Center

   - Click on Custom Locations on the left-hand side
   - Click Add Location
   - Name = South
   - Add a private IP address and subnet mask for your Southern network. For this exercise, we will assume that the service and the client IP addresses are in the same private network. 192.168.1.0/24
   - Continent Code = EU
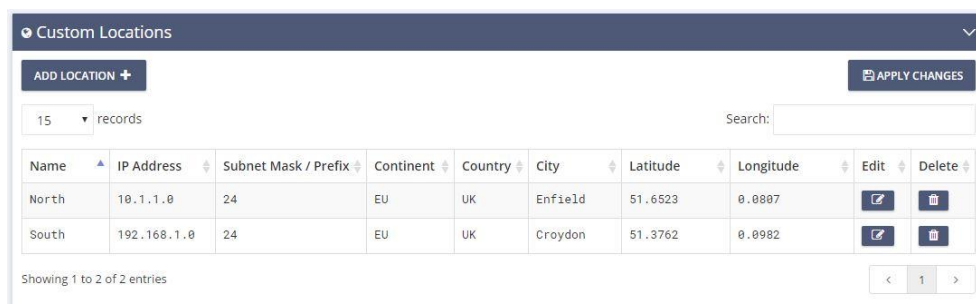   - Country Code = UK
   - City = Croydon

- Latitude (obtained from Google) = 51.3762
- Longitude (obtained from Google) = 0.0982

Note: Please use the correct code, which can be obtained here

The result should appear something like the image below.
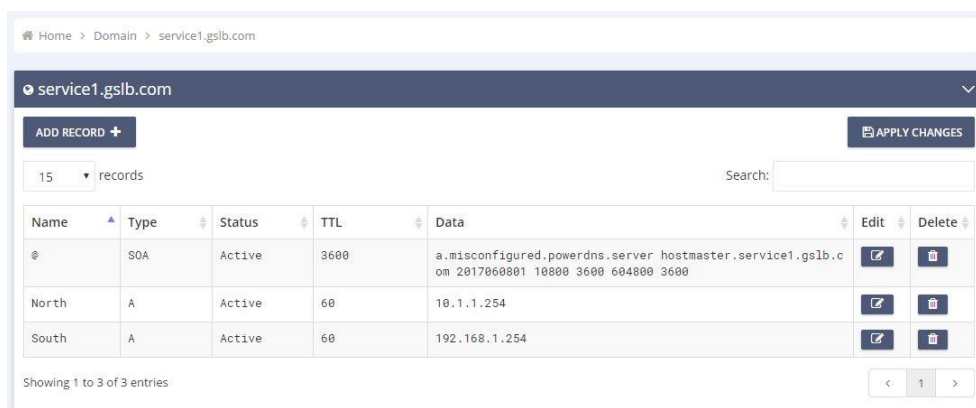

Adding A Records for the custom locations

Add an A record for north.service1.gslb.com

- Click on the domain service1.gslb.com
- Click Add Record
- Name = North
- Type = A
- Status = Active
- TTL = 1 Minute
- IP Address = 10.1.1.254
  (Note this is in the same network as the location Enfield)

Add an A record for south.service1.gslb.com

- Click on the domain service1.gslb.com
- Click Add Record
- Name = South
- Type = A
- Status = Active
- TTL = 1 Minute
- IP Address = 192.168.1.254
  (Note this is in the same network as the location Croydon)

The final table should look like the image below.

Traffic Flow

Example 1 – Client in Northern Data-Center

1. Client IP 10.1.1.23 queries GSLB for service1.gslb.com
2. GSLB looks up the IP address 10.1.1.23 and matches it with Custom Location Enfield 10.1.1.0/24
3. GSLB looks at its A records for the service1.gslb.com and matches north.service1.gslb.com as it is also in the network 10.1.1.0/24
4. GSLB responds to 10.1.1.23 with the IP address 10.1.1.254 for service1.gslb.com

Example 2 – Client in Southern Data-Center

1. Client IP 192.168.1.23 queries GSLB for service1.gslb.com
2. GSLB looks up the IP address 192.168.1.23 and matches it with Custom Location Croydon 192.168.1.0/24
3. GSLB looks at its A records for the service1.gslb.com and matches south.service1.gslb.com as it is also in the network 192.168.1.0/24
4. GSLB responds to 192.168.1.23 with the IP address 192.168.1.254 for service1.gslb.com

Master-Slave Configuration for DNS Replication

You may have instances when you have a pair of GSLB modules running in HA within the same datacentre or across multiple data centres running in Active-Passive or Active-Active. This scenario, coupled with having a large number of DNS records, creates logistical problems for the administrator.
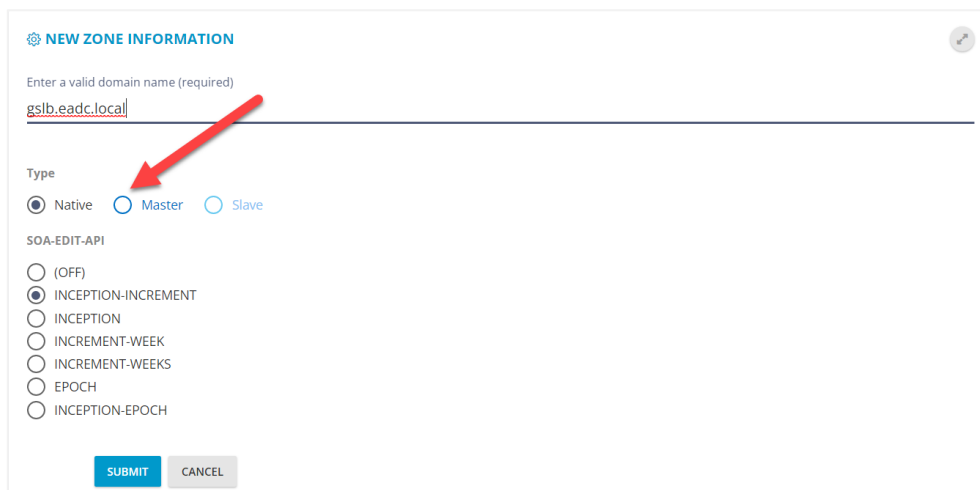
IT staff can end up typing in all the records without an adequate solution for replicating the DNS records, leading to inefficient usage.

The EdgeGSLB has a built-in method of replicating the DNS records to Slave modules when changes are made to the Master, using the Master-Slave(s) concept to replicate DNS data.
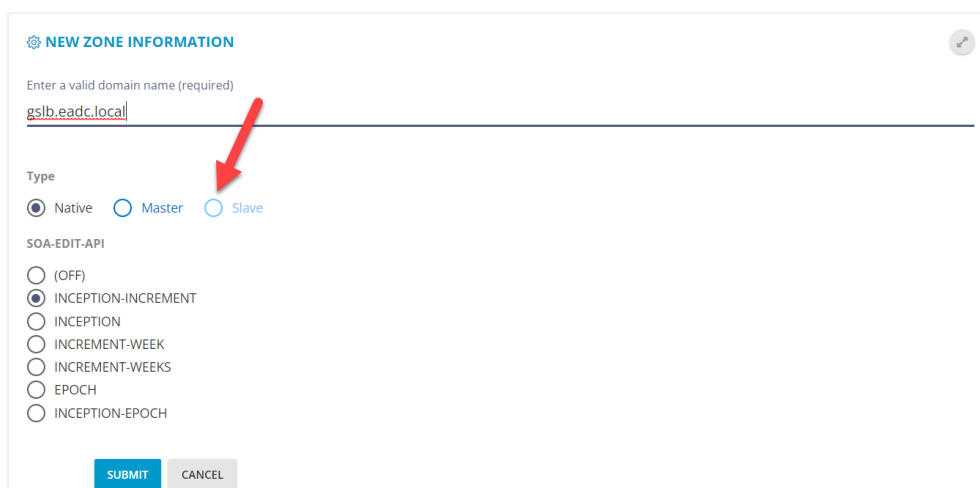
Setting up the Master and Slave(s()

To set up Master-Slave replication, you must first specify the Master GSLB followed by any Slave GSLB module(s).

To do this, ensure you click the Master radio button when creating the domain. This will ensure that this GSLB becomes the Master. This setting cannot be changed once the domain is created.

Once you have configured the Master, the next step will be to create and configure the Slave(s). The same step shown in the image above is used when you create the domain on the Slave GSLB, except for selecting the Slave radio button (see image below).

Configuring for DNS Replication

Let's assume you have created your Master and the Slave GSLB modules. The method below works for an HA pair and cross-data centre replication. The only proviso for multiple data centre usage is that all the GSLB modules must be able to see and talk to each other.

In our explanation, we assume you have two GSLB modules, each with an IP address of 192.168.159.100 and 192.168.159.102, respectively.

Log onto the Master GSLB and proceed to Admin > Settings, located in the side menu bar. You will see something like the example below.
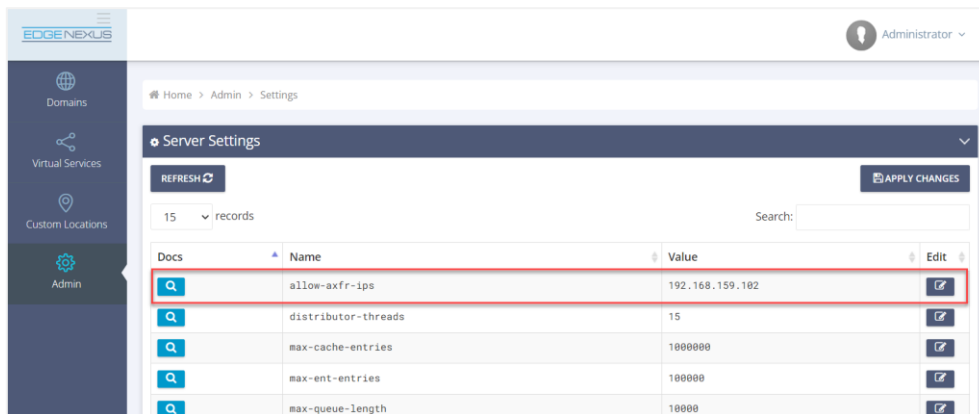

The section we are interested in is highlighted with a red box.

- Click the Edit icon and enter the IP address of the Slave GSLB. If there are multiple Slaves, then you can enter the IP addresses of each, separated with a comma.
- Click the Save button once the entry has been completed.

This will now begin the replication process from Master to Slave.

Dual Data Centre – Dual Arm Example

Sometimes, you may require an Active-Active GSLB architecture that crosses data centres. In the drawing below, we have outlined an example of two data centres with external and internal networks.
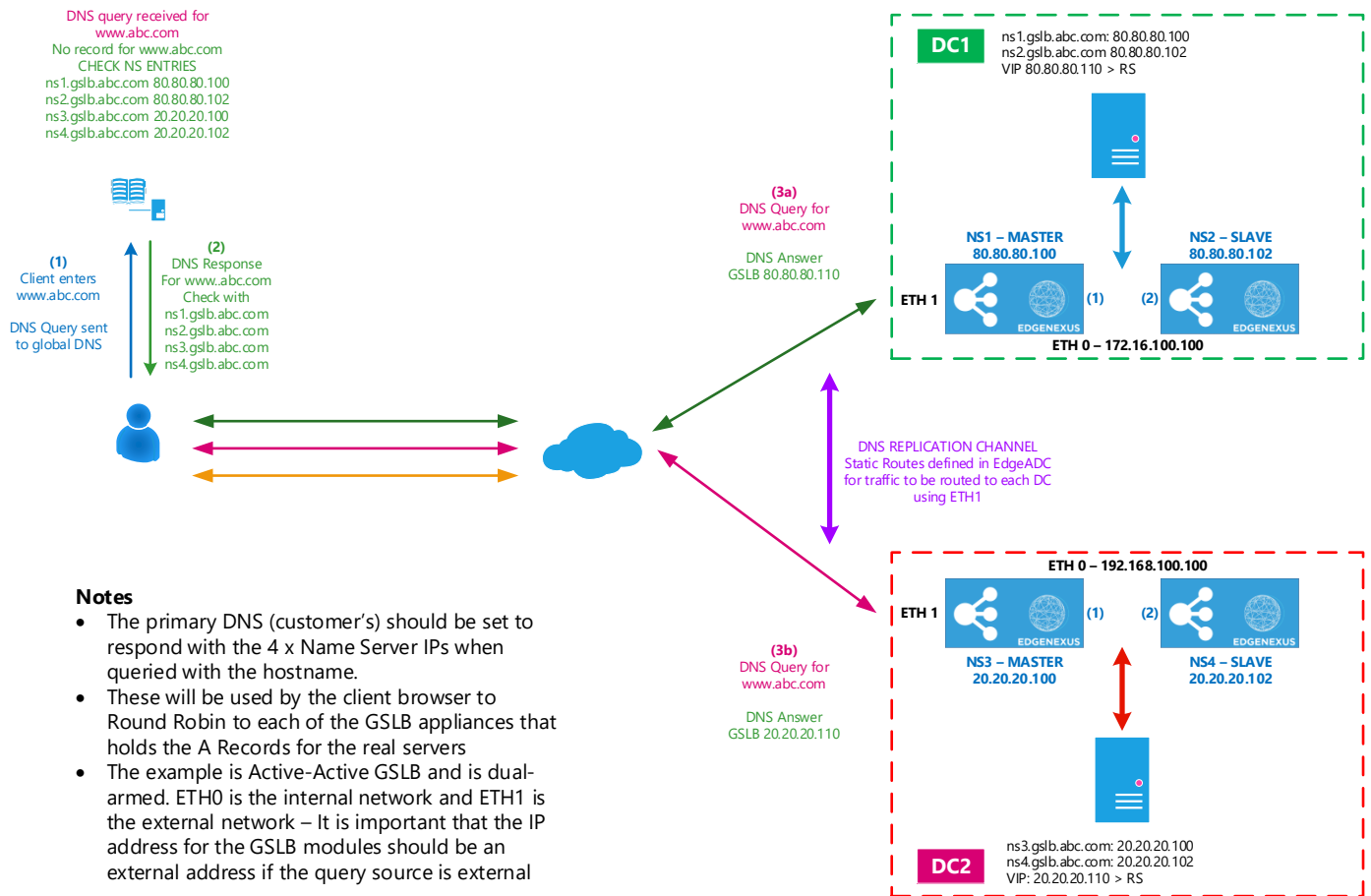
So let's see what we have in this example.

- The user in this example wishes to access a website called www.abc.com (not ABC NEWS!).
- They enter this into their browser, which queries their central DNS for the IP address.
- The central DNS responds to the user's machine that it does not have the IP address for www.abc.com, but it does have a list of Name Servers that may have the IP address for www.abc.com.
- It responds (in the example) with ns1.gslb.abc.com, ns2.gslb.abc.com, ns3.gslb.abc.com and ns4.gslb.abc.com.
- These Name Servers correspond to gslb1 and gslb2 in DC1 and gslb3 and gslb4 in DC2.
- DNS records on NS1 and NS3 Masters will be automatically replicated to NS2 and NS4, respectively.
- You could also have a single Master, say NS1 and the remaining GSLBs as Slaves. The Master will then replicate the records to the Slaves when updates are made. It is important to note that ANY changes to the records are only made on the Master. Also, it is important to understand that should the Master fail; it will need to be brought back up to make record changes.

When you wish to replicate DNS entries from the Master to Slaves that sit across data centres, you must add static routes that are fixed to use ETH1. The routes can be added in the System > Networking section.