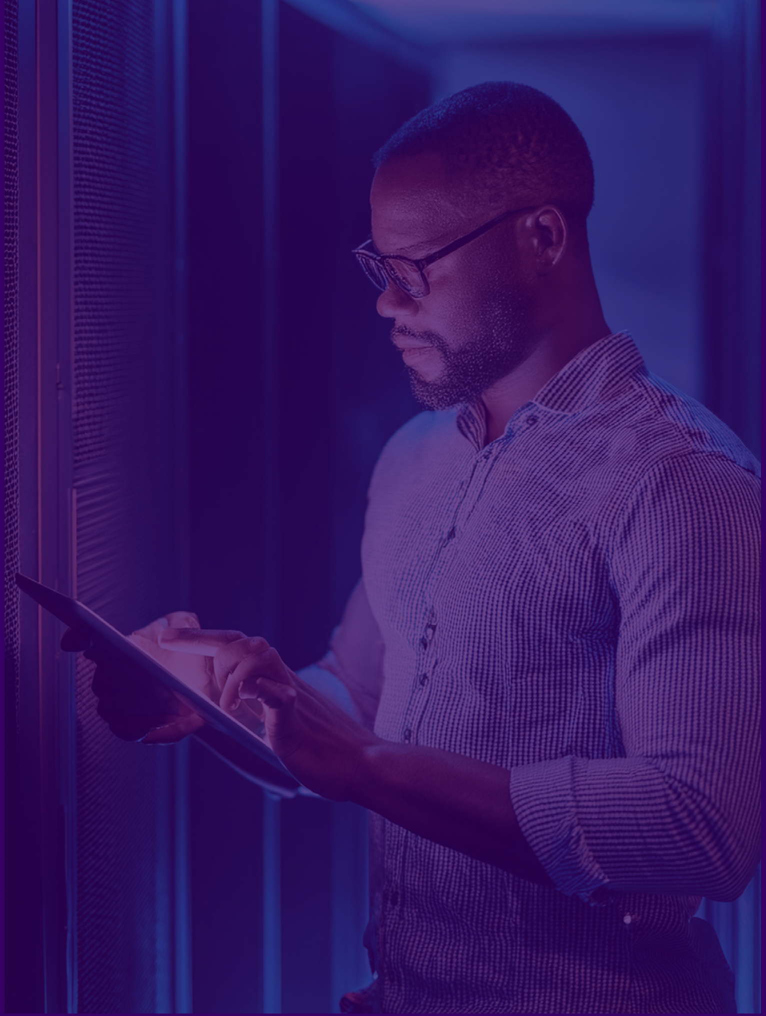
LOADBALANCER



TECHNICAL GUIDE

Active/Active load balancing options for scalable high-performance computing, storage, and AI workloads

Table of contents

INTRODUCTION

To understand how to effectively scale high-performance computing, storage, and AI workloads, you first need to understand the different modes of load balancing that can be used to achieve this objective.

There are five main modes:

- Layer 7 Reverse Proxy
- Layer 4 Direct Server Return (DSR)
- GSLB multi site
- GSLB direct-to-node
- GSLB active-active

The mode recommended will be different for different applications, with the deployment tailored to your specific application requirements.

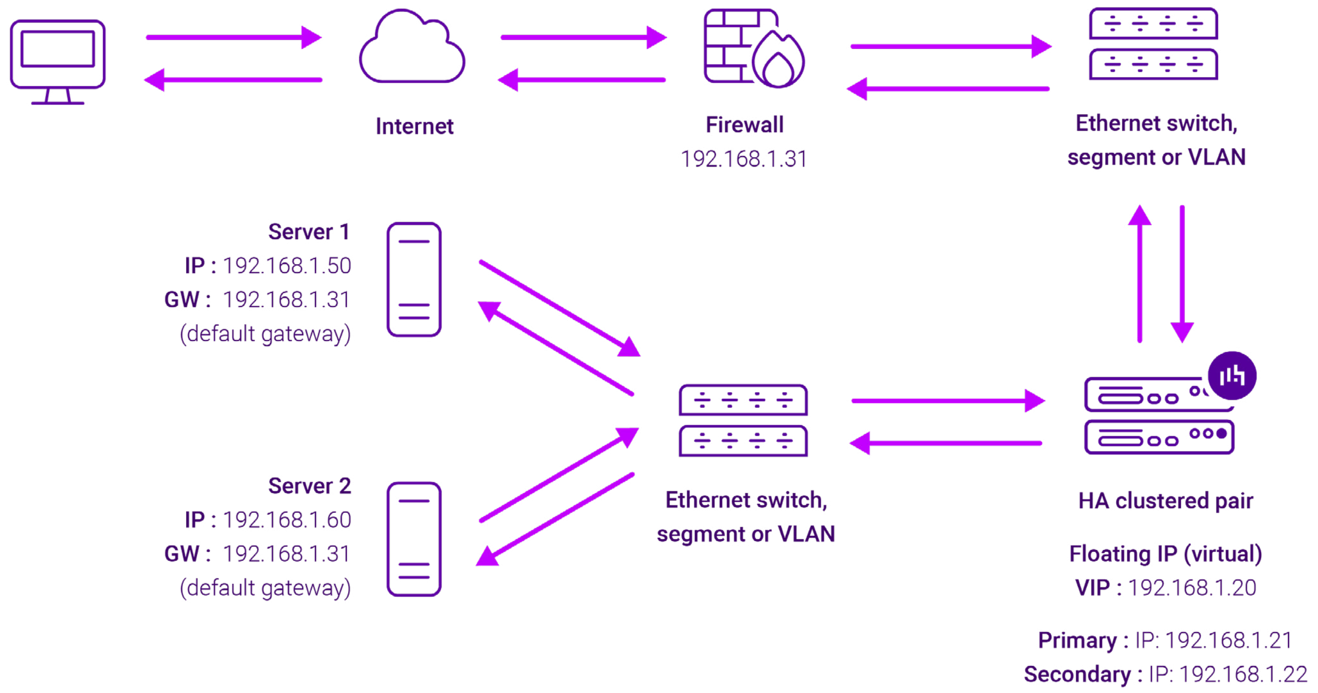
We therefore strongly encourage you to book a meeting with our technical experts so we can advise accordingly.

LAYER 7 REVERSE PROXY

Why use?

Flexible

Layer 7 is the most traditional method of load balancing for providing high availability and scale for applications, used in 90% of application workloads! It is configured in an active/passive pair for built-in HA and resilience. Further reading.



Pros?

- Editable HAProxy ACL rules can be applied to make smarter load balancing decisions to optimize or change content.
- Very flexible persistence methods with cookies / headers etc.
- Easy to implement Quality of Service and Distributed Denial of Service rules.
- No restrictions on the number of services, which can scale up to 90 Gbps throughput via the load balancer.

Cons?

- Can be difficult to see the originating IP address of the client.
- Operates as a full proxy so introduces slightly more latency.
- TCP load balancing only (UDP load balancing is only supported at Layer 4).

Example use cases:

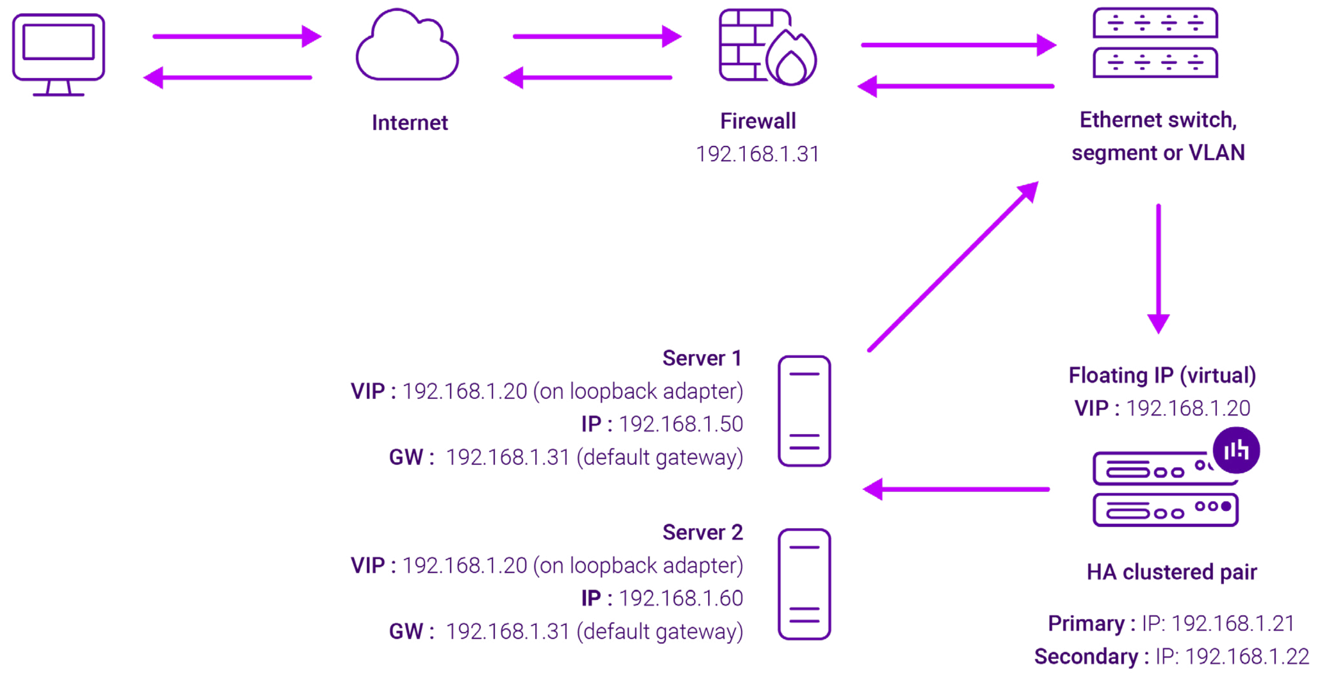
- Cloudian Hyperstore S3 Object Storage

LAYER 4 DIRECT SERVER RETURN (DSR)

Why use?

Fast & transparent

This method goes by many names (e.g. Direct Server Return, Direct Routing) but the concept is the same. Reply traffic flows from the servers straight back to the clients, bypassing the load balancer, maximizing the throughput of return traffic and allowing for near endless scalability. Further reading.



Internet
Firewall
192.168.1.31
Ethernet switch, segment or VLAN
Server 1
VIP : 192.168.1.20 (on loopback adapter)
IP : 192.168.1.50
GW : 192.168.1.31 (default gateway)
Server 2
VIP : 192.168.1.20 (on loopback adapter)
IP : 192.168.1.60
GW : 192.168.1.31 (default gateway)
Floating IP (virtual)
VIP : 192.168.1.20
HA clustered pair
Primary : IP: 192.168.1.21
Secondary : IP: 192.168.1.22

Pros?

- The return response can be 10 times bigger than the original request (i.e. 1GbE->LB->10GbE->Client).
- Simple deployment.
- Lightning-fast transactions, often used for video streaming, e.g. Netflix.
- Cost-effective and less CPU intensive.
- Fully transparent by default i.e. all servers think they are talking directly to the client.

Cons?

- Only supports source IP persistence settings.
- Can't use SSL offloading or WAF's.
- Some environments will not support Layer 4 DR mode.

Example use cases:

- Scality RING Object Storage

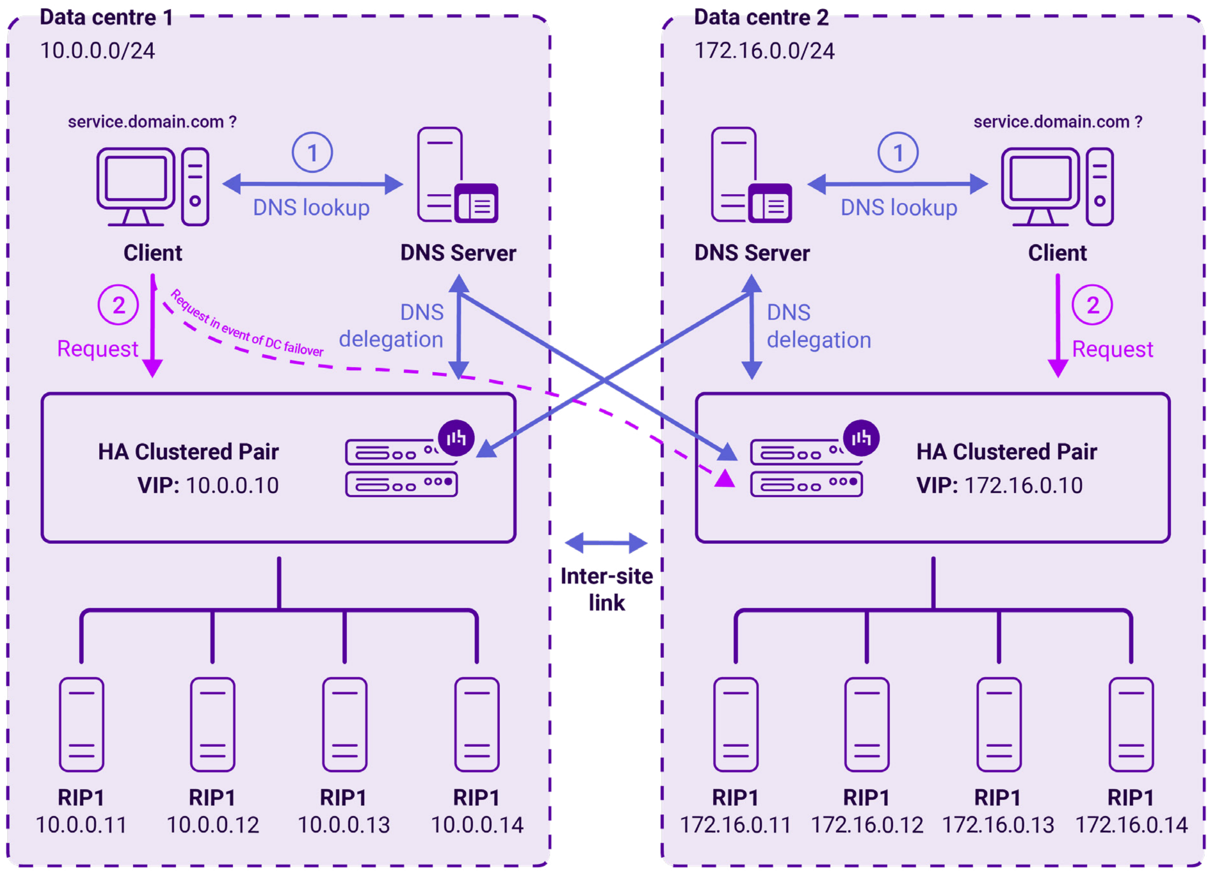
GSLB MULTI-SITE

Why use?

Efficiency, performance, or redundancy

Global Server Load Balancing (GSLB) provides traffic distribution across server resources located in multiple locations, for example, multiple data centers or a single disaster recovery (DR) site for redundancy. It passes the traffic straight to the load balancer which does more granular balancing and health checking. Further reading.

DR site for redundancy example:



Pros?

- Flexible health checks to ensure application uptime.
- Topology based routing ensures that internal traffic uses the local data center, avoiding the cost and performance issues of over-using WAN links.
- It can detect users' locations and automatically route their traffic to the best available server in the nearest data center.

Cons?

- Sometimes applications do not support DNS and therefore can't failover using GSLB (we do however have a solution for that).

Example use cases:

- Disaster recovery or multi-site scenarios

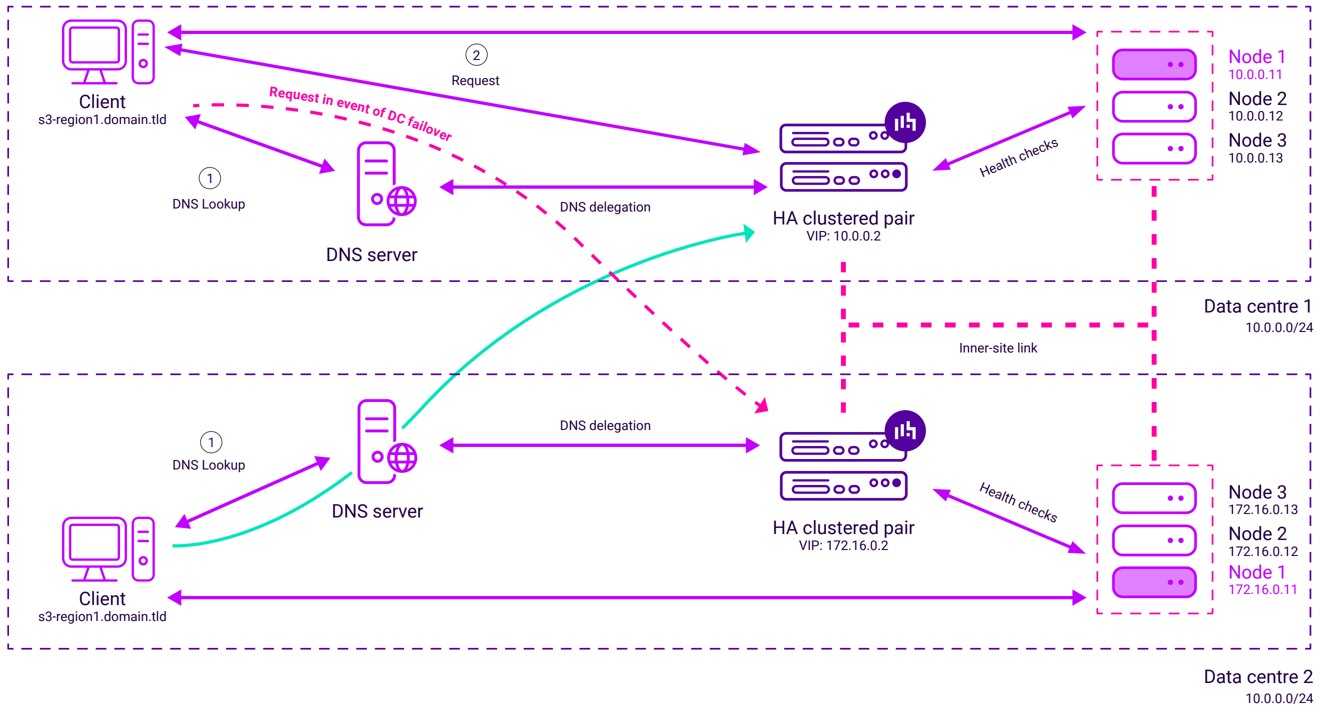
GSLB DIRECT-TO-NODE

Why use?

Best DNS round-robin alternative

In a typical DNS round-robin deployment, requests are sent out to the configured nodes indiscriminately. However, with GSLB direct-to-node, the real-servers undergo health checks to determine their availability. So if a server responds correctly it is kept within the pool of available nodes for the load balancer to send requests to. And if a server fails to respond correctly it is excluded from the pool of available nodes.

GSLB direct-to-node passes the traffic straight to the application servers removing the need for a load balancer and massively increasing performance. However, in order to do that with enough granularity we recommend using a feedback agent on each server to dynamically change the weight in the GSLB. This is on top of the usual flexible health checks. Further reading.



Pros?

- Best alternative to a typical DNS Round-Robin deployment.
- It can provide an unlimited scale of your services as all request traffic is out of the band/ path of the load balancer, meaning there is no bottleneck.
- At scale is surprisingly granular in balancing large amounts of traffic to large amounts of application servers.
- Can be deployed in combination with topology based routing for site affinity.
- Very low cost, low resource, relatively simple and incredibly fast.

Cons?

- Requires careful thought about the sheer number of health checks and feedback agent responses, although this is rarely an issue in production.
- It is out of path, so has less visibility of traffic flows than a load balancer.
- Limited persistence options, and no SSL offload or WAF.
- You should only use GSLB direct-to-node in extremely high throughput environments when your load balancer is no longer able to keep up with throughput.

Example use cases:

- High-volume request workloads such as Veeam backup and recovery

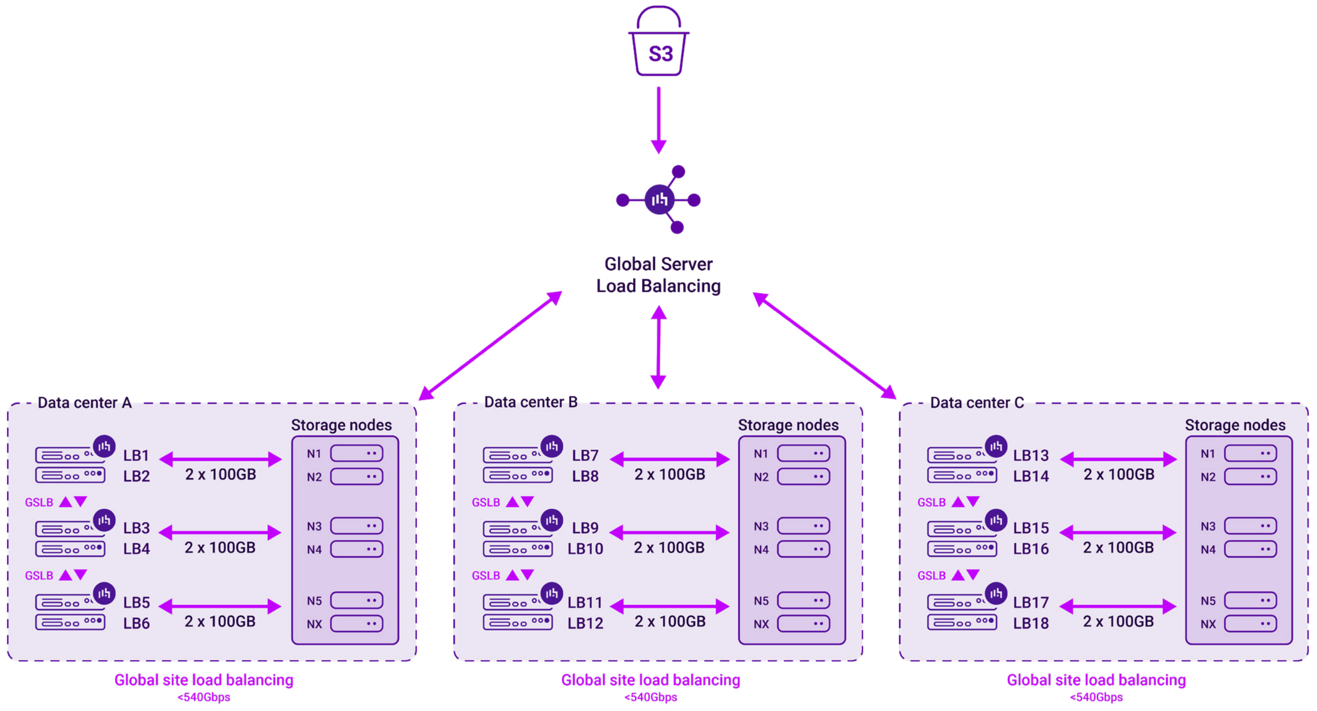
GSLB ACTIVE-ACTIVE

Why use?

Horizontal scaling

Active-active GSLB can be used to scale horizontally when there is a requirement to exceed approximately 88 Gbps through the load balancer (and direct-to-node GSLB isn't suitable). An even distribution across multiple data centers should be considered (if applicable) and redundancy accounted for. [Further reading.]

Single site active/active example:



Pros?

- A simple method of scaling out for bandwidth (<2.8Tbps).
- An exponential increase in total throughput.
- More cost-effective for large-scale deployments.
- A future-proofed configuration using GSLB.

Cons?

- Configuring and managing an active-active GSLB setup is more complex than an active-passive setup.
- If issues arise, troubleshooting can become more complex as you need to consider potential problems with both GSLB devices and the underlying network infrastructure.

Example application:

- High-performance computing (HPC) and advanced AI workloads, including Weka, RedHat, Dremio, and many more...

About the company

Our mission is to ensure your business is never interrupted by downtime — using tailored, high availability solutions to optimize application delivery.

Bringing decades of experience to your deployment, we're here to get to the heart of what matters to you, delivering uptime you measure in years, not months.

Find out if our clever, not complex, Application Delivery Controllers (ADCs) and exceptional, personalized support are the right fit for your application stack.

www.loadbalancer.org

SMART • FLEXIBLE • UNBREAKABLE

LOADBALANCER