

# Load balancers

- [Overview](#)
- [Create a new load balancer](#)
- [Update load balancer configuration](#)

# Overview

**Load balancers are essential to achieve High Availability (HA) and horizontal scalability.**

Most load balancers from cloud providers are limited to balance traffic only inside network of the cloud provider, also to get SSL termination for your custom domains you either have to delegate your domain name servers to the provider or follow a long and complex manual process for each domain to connect.

Finally, advanced features like rate limiting, output caching or headers rewriting are usually requiring to add more infrastructure (WAF, Cache server) or more code to write/deploy/maintain (AWS Lambda, Cloudflare workers)

**At Elestio, we dreamed about a new kind of cloud load balancer able to:**

- Balance traffic to targets on any cloud
- Support targets over IPV4, IPV6 & Global private IP (Nebula) and cname
- Load balance UDP / TCP / HTTP / HTTPS traffic
- Manage SSL certificates automatically with zero setup needed
- Support for "Health Checks", "Sticky sessions", "Proxy Protocol", "Force HTTPS"
- Support for output headers add/edit/remove
- Easy user interface
- Affordable

Load balancers by Elestio are now available on our 5 partner cloud providers (AWS, Digital Ocean, Hetzner, Linode, Vultr) and are also available on your infrastructure with BYOVM.

<https://dash.elest.io/default/Elestio-services/load-balancer>

[Check our tutorial here about creating a load balancer](#)

# Create a new load balancer

To create a new load balancer, go to the dashboard and click on Load Balancer on the left side

<https://dash.elest.io/default/Elestio-services/load-balancer>

**elestio**

PROJECT: Elestio-services

Services

Load Balancer

Members

Billing

Settings

Audit Trail

\$14491.50 CREDITS

\$0.2310/hour SPENDING

[Add credits](#)

### Create Load Balancer

1. Select Service Cloud Provider

HETZNER DigitalOcean Amazon Lightsail linode VULTR

2. Select Service Cloud Region

Europe North America

fsn1  
Germany - Falkenstein

hel1  
Finlande - Helsinki

nbg1  
Germany - Nuremberg

3. Select the target services which you want to connect with load balancer

*i* You can add up to 25 targets by selecting them in the dropdown or indicate ipv4, ipv6 or global private ip addresses of your targets.

Provider: Hetzner Cloud

Region: Europe, Germany Falkenstein

Estimated Monthly Price\* \$10

\*Estimated monthly price is based on 730 hours of usage.

Create Load Balancer

There, select your cloud provider (or BYOVM) then select your preferred region.

Then select one or multiple targets for your load balancer

### 3. Select the target services which you want to connect with load balancer



You can add up to 25 targets by selecting them in the dropdown or indicate ipv4, ipv6 or global private ip addresses of your targets.

Target Service	Selected
elestio-nebula-one-u3.vm.elestio.app	<input checked="" type="checkbox"/>
elestio-nebula-two-u3.vm.elestio.app	<input checked="" type="checkbox"/>
haproxy-internal-smtp-u3.vm.elestio.app	<input type="checkbox"/>
elestio-doc-u3.vm.elestio.app	<input type="checkbox"/>
elestio-nebula-three-u3.vm.elestio.app	<input checked="" type="checkbox"/>
minio-lux-u3.vm.elestio.app	<input type="checkbox"/>
appdrag-uptime-u3.vm.elestio.app	<input type="checkbox"/>
elestio-nodebb-u3.vm.elestio.app	<input type="checkbox"/>

Targets can be services in the same project, or any other target pointed by cname, IPV4, IPV6 or global private IP. You can mix targets from several providers/datacenters, this way you can improve your reliability. Since it's possible to point to the global private ip, your targets don't have to be exposed on internet and can be configured to be reached only through the load balancer.

#### 4. Forwarding Rules



Set how traffic will be routed from the Load Balancer to your service. At least one rule is required.

Load Balancer			Target		
Protocol	Port	→	Protocol	Port	
HTTP	80		HTTP	3000	
Protocol	Port	→	Protocol	Port	
HTTPS	443		HTTP	3000	
<a href="#">+ Add Another</a>					

Next, configure your forwarding rules from the load balancer to the targets. All incoming traffic will be sent to targets based on the rules defined below, the load balancer will use the "least connections" algorithm to decide where to send the traffic, taking account of target health and current load of targets.

It's possible to route TCP, UDP, HTTP & HTTPS traffic from the load balancer to the targets. It's possible to contact the targets over a different protocol if needed, for example from HTTPS on port 443 received on the load balancer to HTTP on port 3000 on the target, this is a very common use case when deploying an application with high availability.

#### 5. Provide Load Balancer Name



The load balancer name cannot be changed afterwards.

Name\*

lb-cuaqm

Next, give a name to your load balancer, it cannot be changed, but you can add additional custom domain names to it.

From there you can click on "**Create Load Balancer**" button; or check the optional advanced configuration below.

## 6. Advanced Configuration

▲ Close Advanced Configuration

### 6.1 SSL/TLS Configuration (Optional)



To activate HTTPS / TLS / SSL on custom domain names on your service you must indicate the domain name(s) in the list below. You will also need to create a DNS entry on your domain name (from your registrar control panel) pointing to your service. You must create an A record for your domain or subdomain pointing to the ip of your service

google.com x yahoo.com x mysite.net x

Enter domain/sub domain (hit 'Enter' to add)

In advanced options, you can add up to 1000 custom domain names per load balancer

More advanced options are available to fine tune your load balancer:

### 6.3 Sticky Session

Enable Sticky Session

### 6.4 Proxy Protocol

Enable Proxy Protocol

### 6.5 Log Traffic

Enable Log Traffic

### 6.6 Output cache (In seconds)

0

### 6.7 Host Header


\$http\_host

### 6.8 IP Rate Limiter

Enable Rate Limiter

### 6.9 Edit HTTP/HTTPS header

#### a. Set output headers

Key	Value	
<input type="text" value="key"/>	<input type="text" value="value"/>	
<a href="#">+ Add Another</a>		

#### b. Remove response headers

**Sticky sessions:** useful for backends that require visitors to be always forwarded to the same node for session management.

**Proxy Protocol:** forward the original visitor ip address to the backend, Switching proxy protocol on for targets that do not support it will render the whole service inaccessible. Alternatively the original visitor IP is available in the header "x-forwarded-for".

**Log traffic:** log all the traffic (visible in the log tab)

**Output cache:** Serve cache for a duration configured in seconds for all GET requests

**Host header:** by default (\$http\_host) will pass to the targets the original host received by the load balancer, this can be defined to a fixed value expected by the targets if required.

**IP Rate Limiter:** limit the maximum number of requests allowed per second per IP address.

**Add/Edit HTTP headers:** allow to add or edit http headers returned by the targets to the visitors

**Remove response headers:** allow to remove http headers returned by the targets to the visitors

To achieve full HA you will need at least 2 load balancers in different regions and point your DNS to both load balancers with round robin DNS entries.

# Update load balancer configuration

Once your load balancer is deployed, you will be able to edit its configuration at any time from the UI

**Termination protection** Disabled. VM can be powered off and terminated. Protection deactivated

**Load Balancer configuration** Manage load balancer config Close

Apply Changes

### 1. Select the target services which you want to connect with load balancer

**i** You can add up to 25 targets by selecting them in the dropdown or indicate ipv4, ipv6, cname or global private ip addresses of your targets.

redis-bgrkj-u76.vm.appdrag.net x

### 2. Forwarding Rules

**i** Set how traffic will be routed from the Load Balancer to your service. At least one rule is required.

Load Balancer		→	Target	
Protocol	Port		Protocol	Port
HTTP	80		HTTP	3000
Protocol	Port		Protocol	Port
HTTPS	443		HTTP	3000

[+Add Another](#)

### 3. Advanced Configuration

Close Advanced Configuration

#### 3.1 Custom Domain Names (Optional)

**i** To activate HTTPS / TLS / SSL on custom domain names on your service you must indicate the domain name(s) in the list below. You will also need to create a DNS entry on your domain name (from your registrar control panel) pointing to your service. You must create a **CNAME** record pointing to your service cname. Alternatively, you can create an **A record** pointing to the ip of your service.

Enter domain/sub domain (hit 'Enter' to add)

#### 3.2 Force HTTPS

**i** When enabled, traffic on port 80 HTTP will be redirected to HTTPS. This have no effect if you have manually defined a forwarding rule listening on port 80.

Enable Force HTTPS

#### 3.3 Sticky Session

Enable Sticky Session

#### 3.4 Proxy Protocol

Enable Proxy Protocol

#### 3.5 Log Traffic

Enable Log Traffic

#### 3.6 Output cache (in seconds)

0

#### 3.7 Host Header

\$http\_host

#### 3.8 IP Rate Limiter

Enable Rate Limiter

#### 3.9 Edit HTTP/HTTPS header

##### a. Set output headers

Key	Value
key	value

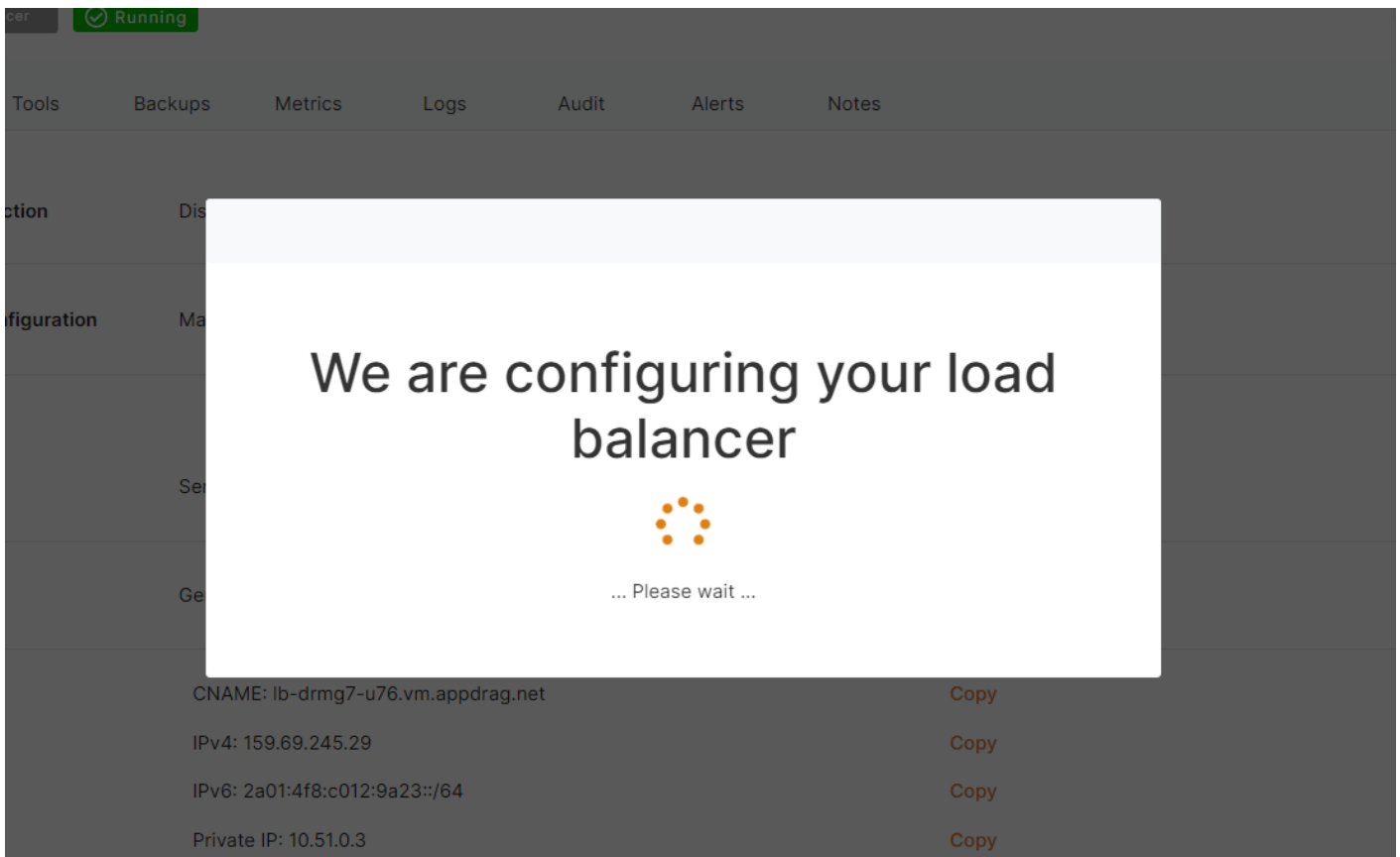
[+ Add Another](#)

##### b. Remove response headers

Enter headers (hit 'Enter' to add)

Apply Changes

After making changes to your configuration (targets, forwarding rules, advanced configuration), click on the "Apply changes" button, this should take 5-10 seconds to deploy your new configuration to the load balancer.



The screenshot shows a dark-themed interface with a top navigation bar containing 'Tools', 'Backups', 'Metrics', 'Logs', 'Audit', 'Alerts', and 'Notes'. A green 'Running' status indicator is visible in the top left. A central white modal box displays the text 'We are configuring your load balancer' with a loading spinner icon and the text '... Please wait ...'. Below the modal, a list of network details is shown with 'Copy' buttons next to each item:

CNAME: lb-drmg7-u76.vm.appdrag.net	Copy
IPv4: 159.69.245.29	Copy
IPv6: 2a01:4f8:c012:9a23::/64	Copy
Private IP: 10.51.0.3	Copy

Done, your configuration is deployed and active!