Load balancers

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 delagate 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.

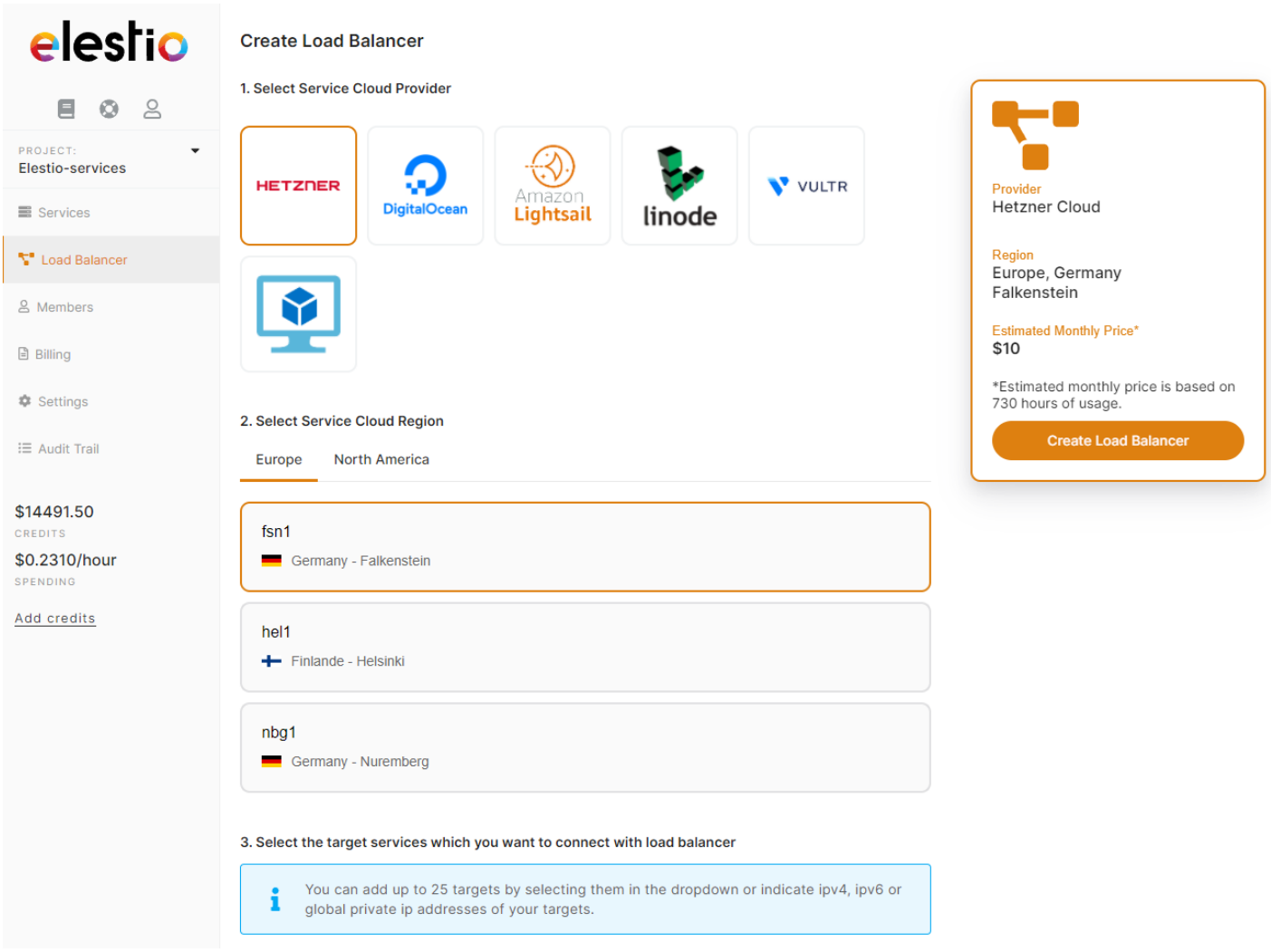
https://dash.elest.io/default/Elestio-services/load-balancer

Check our tutorial here about creating a load balancer

Create a new load balancer

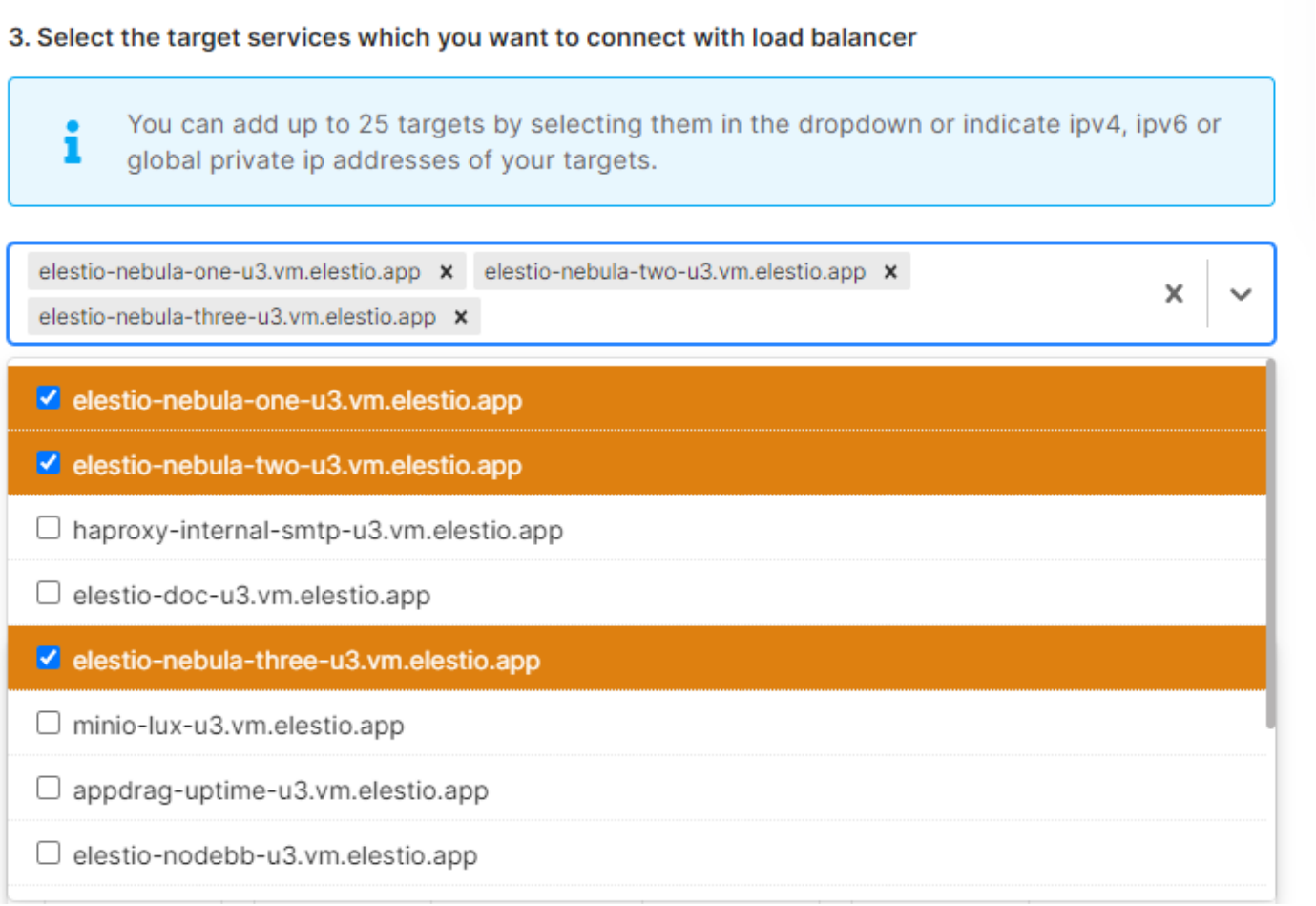
To create a new load balancer, go to the dashboard and click on Load Balancer on the left side

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



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

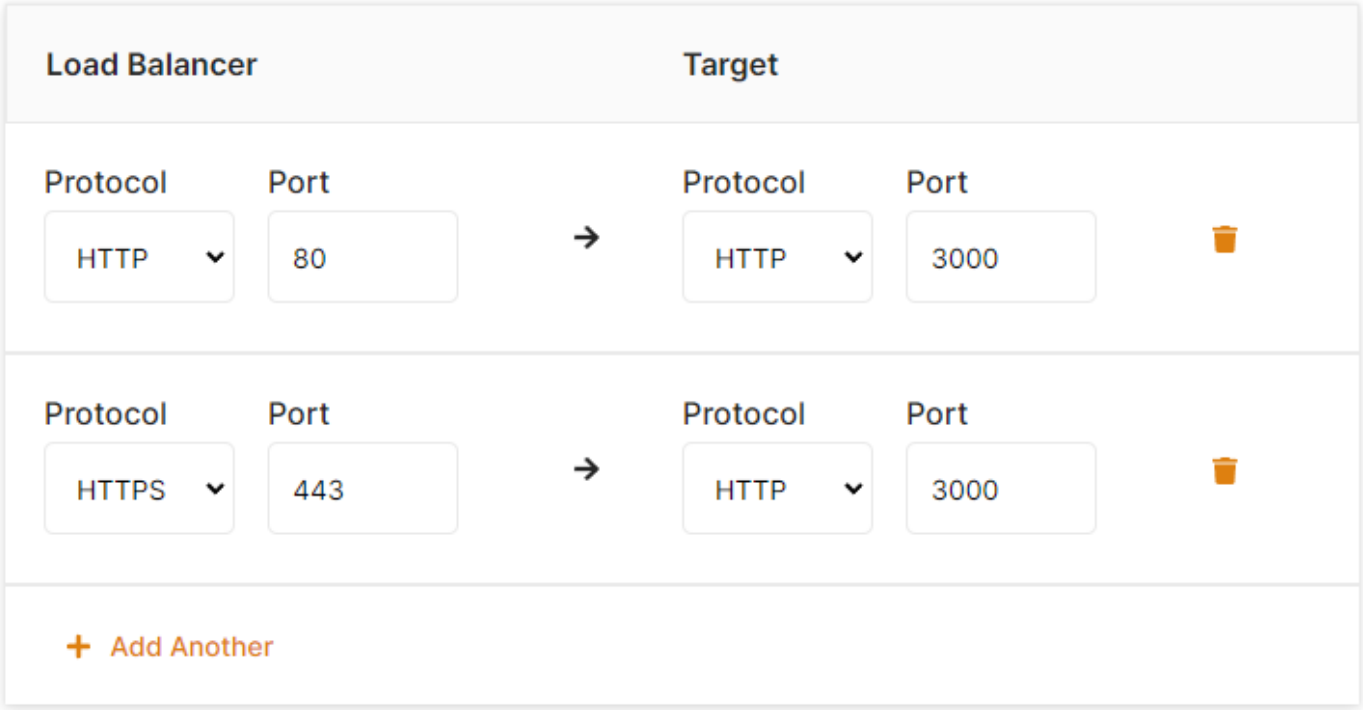
Then select one or multiple targets for your load balancer



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.



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

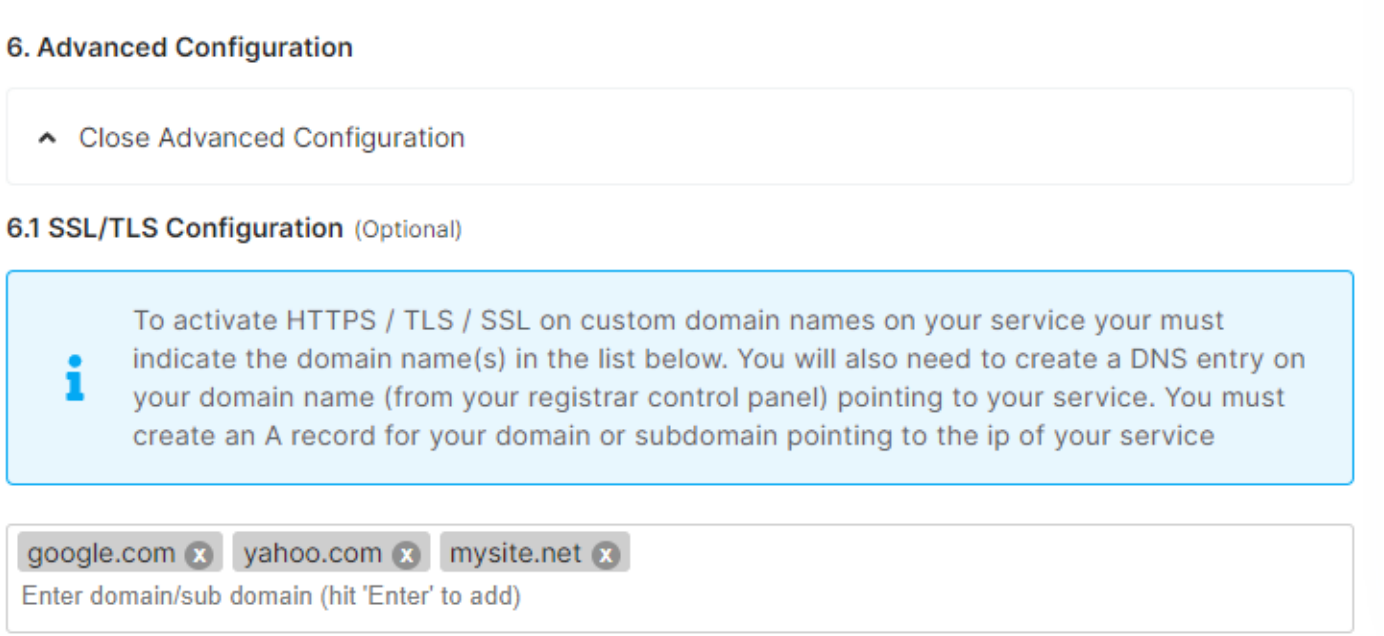
The load balancer name cannot be changed afterwards.

Name*

lb-cuaqm

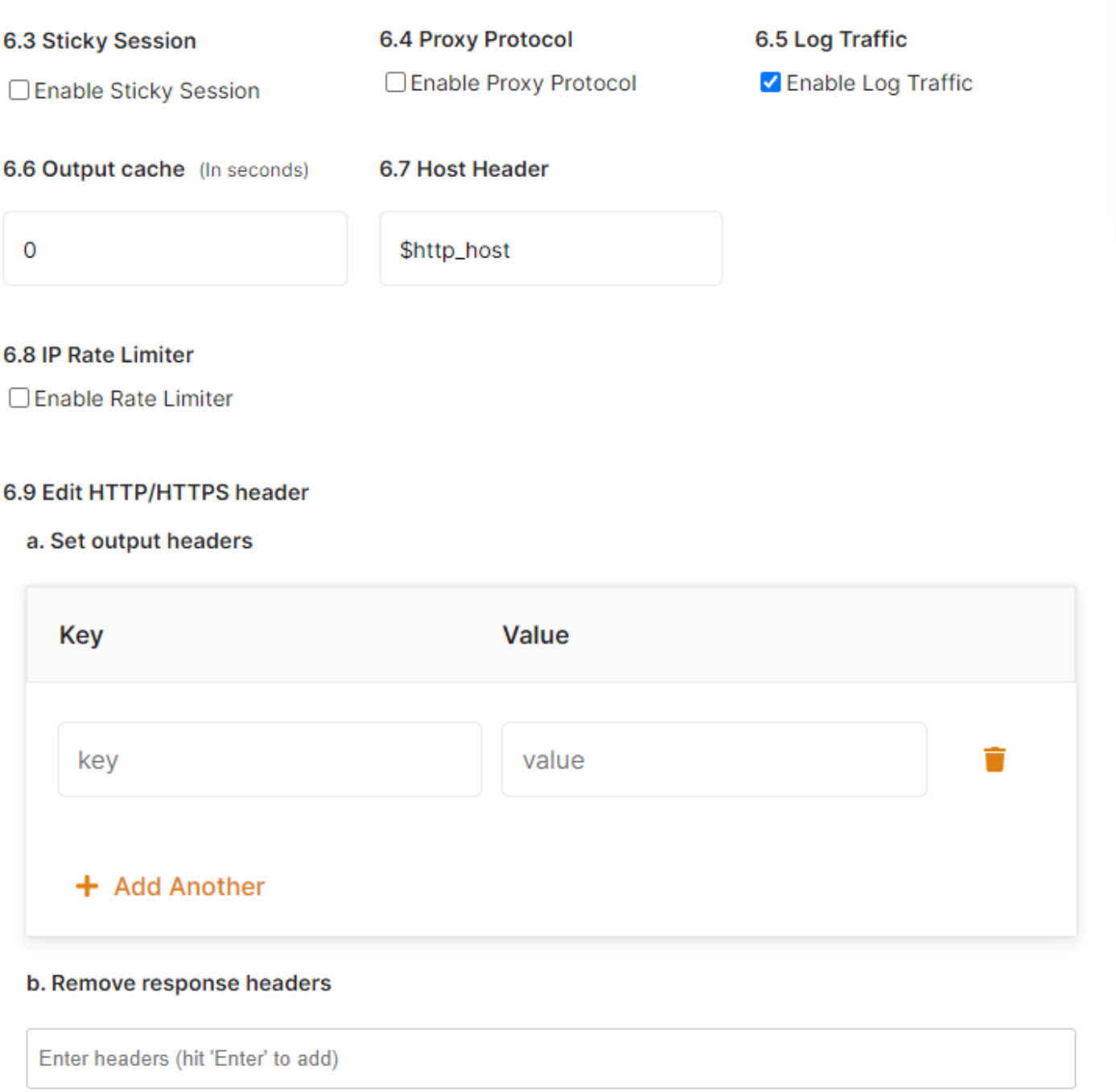
Next, give a name to your load balancer, it cannot be changed, but you can add additional custom domain names to it.

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



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:



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

Proxy Protocal: 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

PROJECT:
elestio-services

- Services
- Volumes
- Load Balancer
- CI/CD
- Domains
- Members
- Billing
- Project Setting
- Audit Trail
- Account
- Support Tickets
- Documentation

$49.60
CREDITS

lb-drmg7
Load Balancer | Running

Open terminal | Delete service | Clone this service

Overview | Tools | Backups | Metrics | Logs | Audit | Alerts | Notes

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

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 ×

2. 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	
HTTPS	443		HTTP	3000	

+Add Another

3. Advanced Configuration

Close Advanced Configuration

3.1 Custom Domain Names (Optional)

To activate HTTPS / TLS / SSL on custom domain names on your service your 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

When enabled, traffic on port 80 HTTP will be redirected to HTTPS. This have not 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

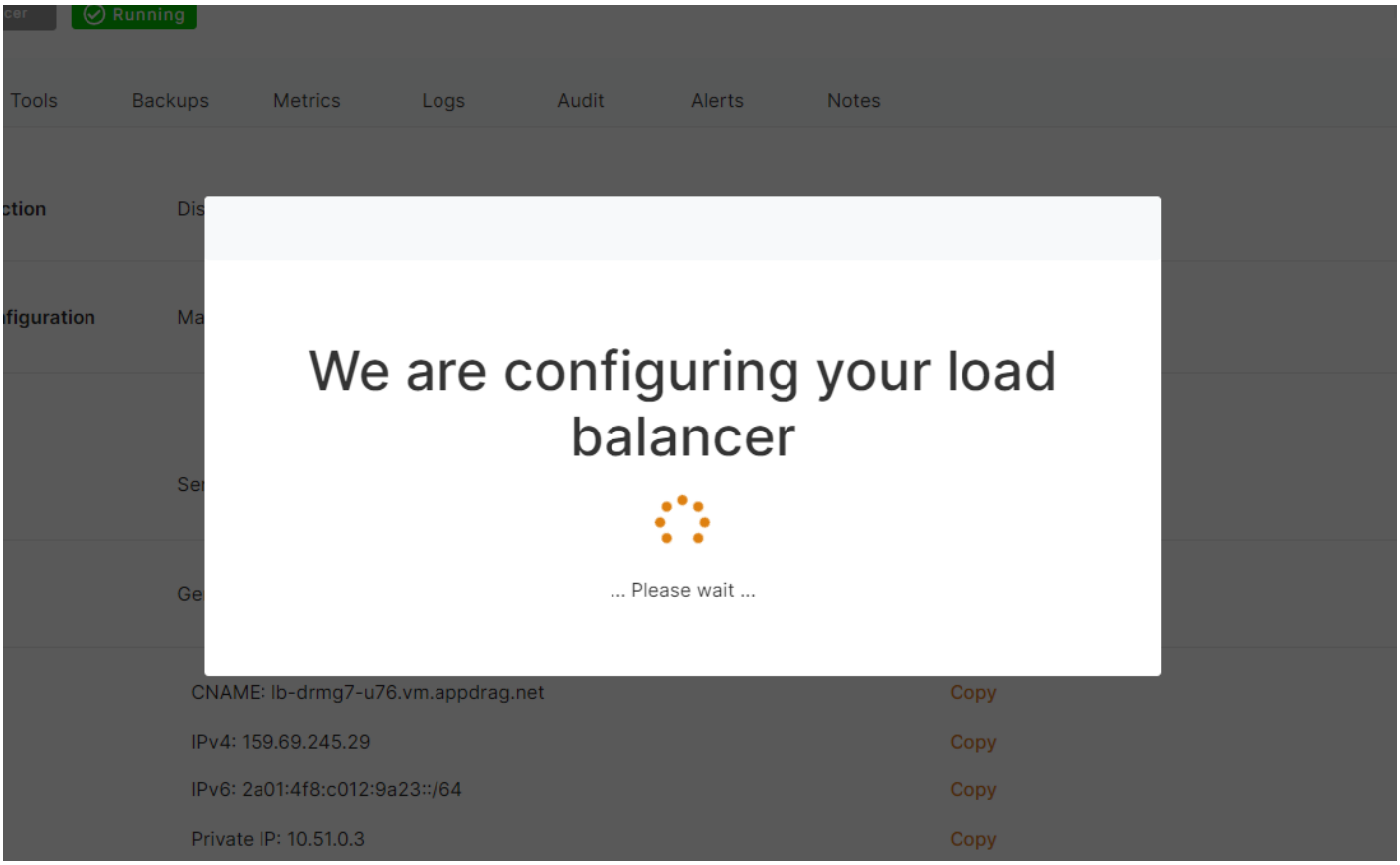
+ Add Another

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.



Done, your configuration is deployed and active!