How-To

LoadBalancer Automatic configuration for HyperBalance

Version 1.3, January 2023

Contents

INTRODUCTION

During our PoC phase or your basic testing, there are sometimes some troubles or questions on how-to configure your load-balancer …

There is a simple way to avoid misconfiguration for HAProxy config file or HyperBalance configuration (typo error, port numbers, etc.) during installation for PoC and test (SE tool or tool provided to a customer) or multi-automated-installation via Vagrant.

This document will show you the HyperBalance details.

Disclaimer / Warning

Use this tool with precautions (review the config file created manually for a double-check) for your environment : it is not an official tool supported by Cloudian.
Cloudian can **NOT** be involved for any bugs or misconfiguration due to this tool. So you are using it at your own risks and be aware of the restrictions.

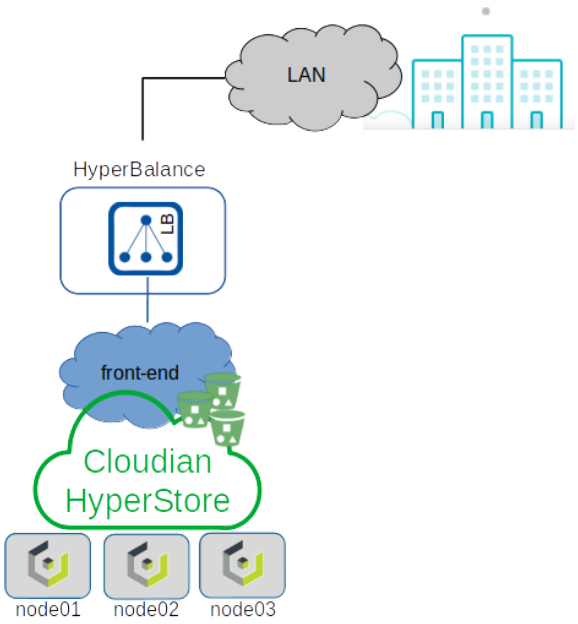
Goal of the document

This document introduces a How-To configuration for the HyperBalance appliance on top of HyperStore.
The goal is to build a HyperBalance configuration file based on the information already in place in a fresh installation of a Cloudian cluster.

The result is a config file for a dedicated load-balancer (1 host or 1 VM) to Cloudian cluster pushed directly to the load-balancer (no GSLB configuration, HA config should be coming soon).

Architecture example

Starting on One-Arm or Two-Arm topologies, the design is similar to the drawing below.



In this document, we will show you an example on the same local subnet (Front-End) but with 2 IPs for the HyperBalance :

- 1 for the LB administration
- 1 for Cloudian - floating IP

The LB administration will be used to change the configuration/administration tasks on the LB

The floating IP is used for the S3 services, CMC, IAM, etc.

Requirements and restrictions

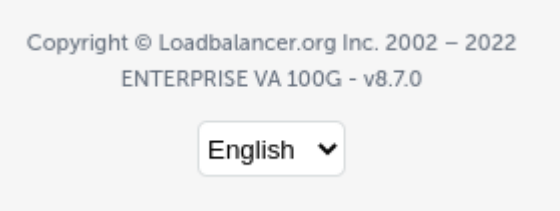
Prerequisites

The HyperBalance python tool is configured to work with **7.4.x** releases and upper (already tested on **HS 7.5**)

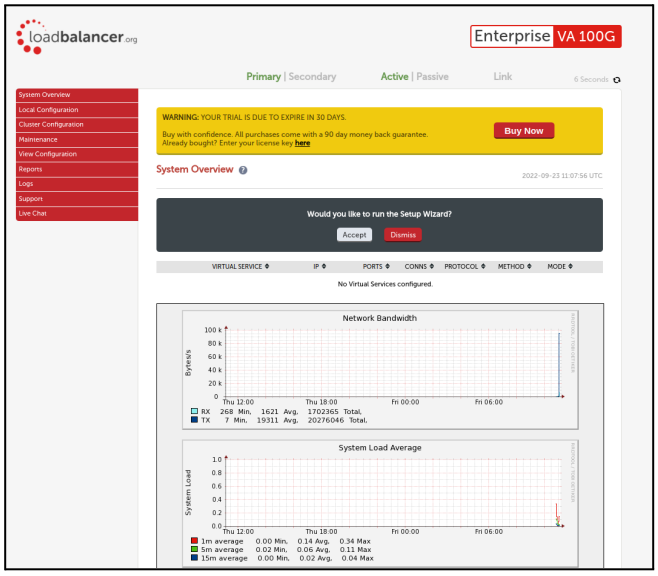
The HyperBalance OVA is imported and deployed matching the prerequisites of the LB.org documentation. You must run the setup script and configure at least the IP for the GUI (login/password as well).

We are supposing for the rest of the documentation, that your LB is connected to a network reachable by the Cloudian nodes with SSH protocol (port 22).

Preambule



We are using the OVA from LoadBalancer.org in this first release of the document.



You should have something similar to the picture here : no LB configuration except the basic needs.

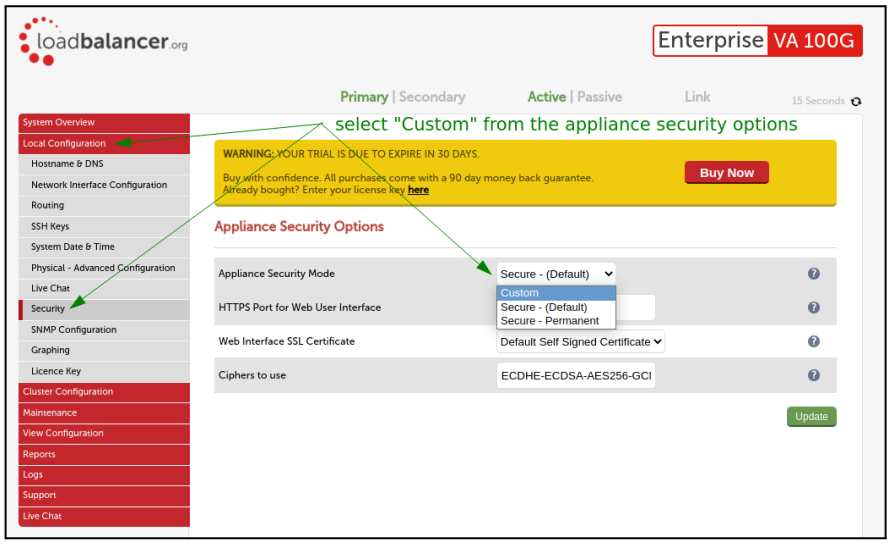
Please avoid to change or manage anything in the HyperBalance configuration before creating the configuration with the tool (especially the High Availability configuration).

We recommend first to create the configuration with this tool and then the HA configuration propagating the new configuration safely.

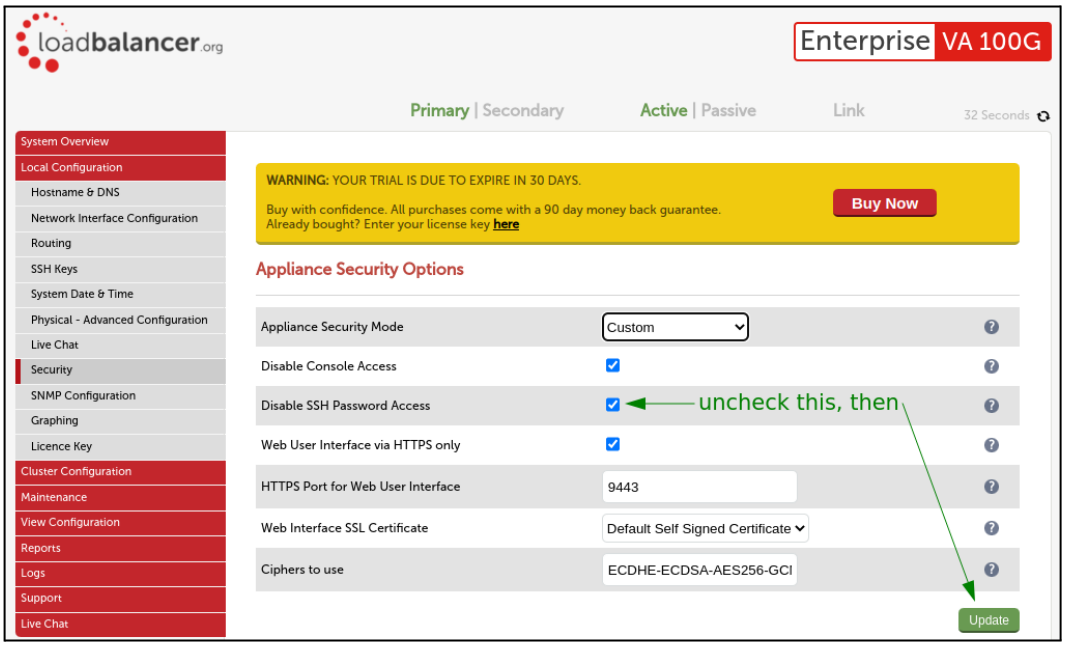
First step - on LB

The Python tool can automatically configure the LB but we have to allow it. So, we must change the security parameters.

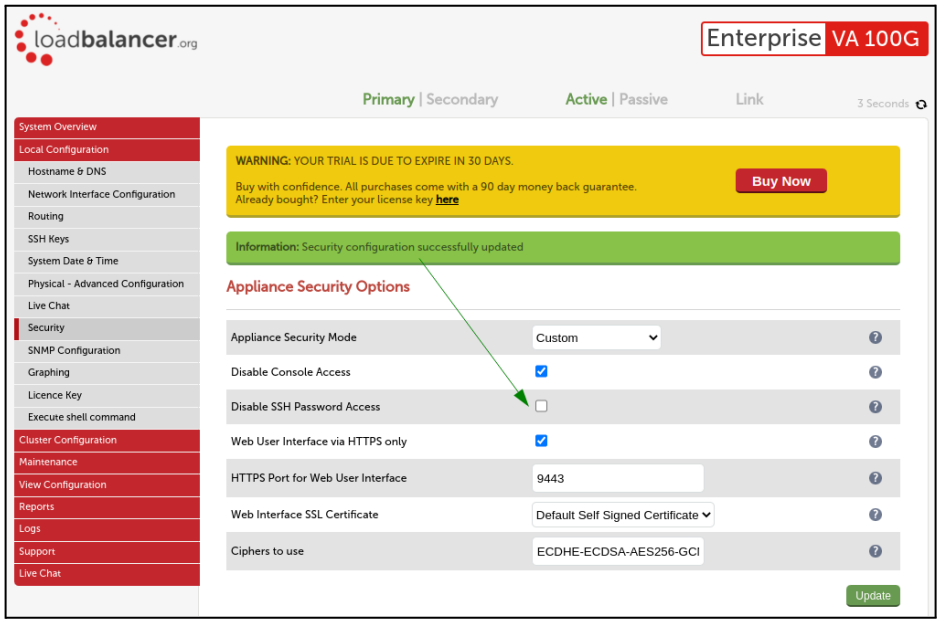
Login on the LB, go to “Local Configuration” →”Security” → Change the “Appliance Security Mode” to “Custom”.



Next, uncheck the “Disable SSH password access” to allow our python tool to send the configuration we want via SSH. Don’t forget to “Update” the configuration on the LB.



The result should be similar to this :



We are done for the LB part now. Proceed to the second step in the next chapter.

We can disable again the SSH password access after the configuration applied if that matches your security rules.

Again, DO NOT configure the High Availability feature (or something else). The HyperBalance configuration must be basically what we configured until now.

Second step - on Puppet master node

Download the python tool or Upload it on the puppet master

Download the last zip archive from the Github repository and upload it to the Cloudian puppet master for example.

command line :

wget https://github.com/pitdive/haproxy_config/archive/refs/heads/master.zip -O haproxy_config.zip

Example :

```
# wget
https://github.com/pitdive/haproxy_config/archive/refs/heads/master.zip -O
haproxy_config.zip

…

2022-09-26 15:37:30 (1.15 MB/s) - ‘haproxy_config.zip’ saved [14475]
```

Extract the files and go to the directory :

```
[root@cloudlab01 ~]# unzip -qqo ./haproxy_config.zip

[root@cloudlab01 ~]# cd haproxy_config-master/

[root@cloudlab01 haproxy_config-master]# ls
haproxy_config.py  haproxy_config_template.cfg  README.md
```

First run, try to have the HELP informations (if needed) :

```
[root@cloudlab01 haproxy_config-master]# python ./haproxy_config.py --help
usage: haproxy_config.py [-h] [-s SURVEY] [-i INSTALL] [-c COMMON] [-f FOLDER]
[-hb] [-hbr] [-bs3 BACKUPS3] [-ms MAILSERVER]                    [-mf
MAILFROM] [-mt MAILTO]

parameters for the script

optional arguments:
  -h, --help            show this help message and exit
  -s SURVEY, --survey SURVEY
                        indicate the survey file, default = survey.csv
  -i INSTALL, --install INSTALL
                        indicate the installation file, default =
                        CloudianInstallConfiguration.txt
  -c COMMON, --common COMMON
                        indicate the common.csv file, default = common.csv
  -f FOLDER, --folder FOLDER
                        indicate the folder including all config files,
                        default = local folder
  -hb, --hyperbalance   specify you want to create a HyperBalance
                        configuration
  -hbr, --hbrevert      revert the hyperbalance config applied
  -bs3 BACKUPS3, --backups3 BACKUPS3
                        indicate the DC in backup/stand-by mode for s3,
                        default=none
  -ms MAILSERVER, --mailserver MAILSERVER
                        mail server name or @IP for alerts
  -mf MAILFROM, --mailfrom MAILFROM
                        indicate the sender, default = haproxy@localhost
  -mt MAILTO, --mailto MAILTO
                        indicate the recipient, default = root@localhost
```

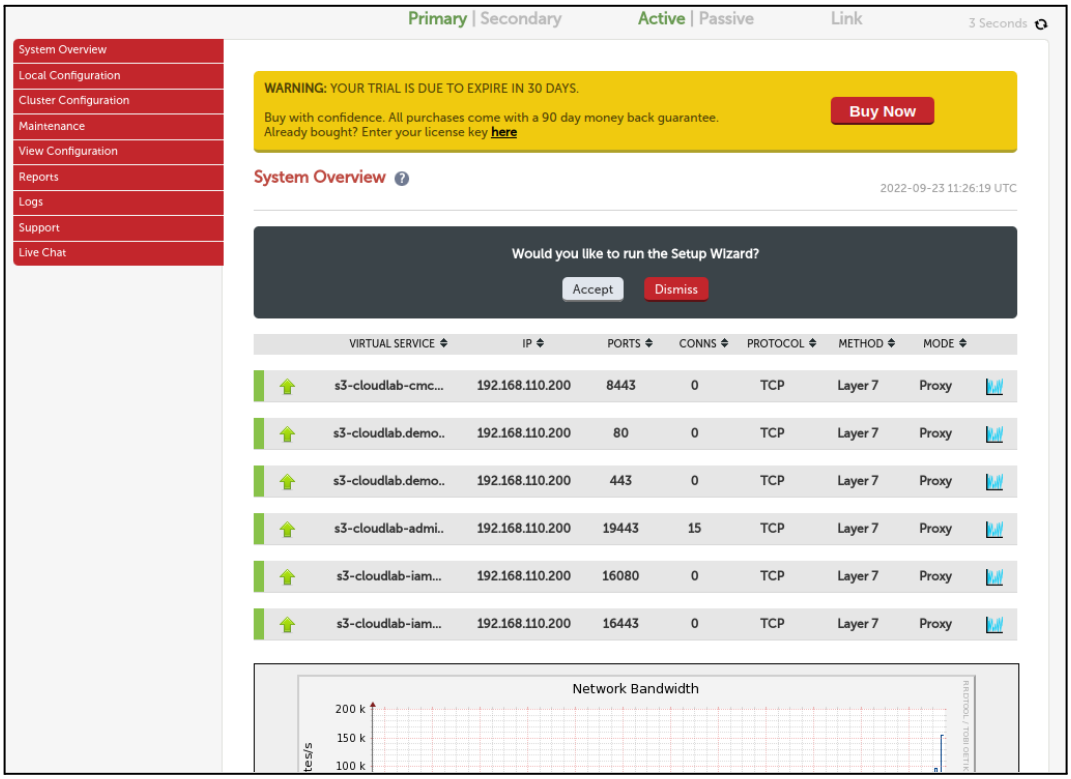
if you are ready and your loadbalancer too, let's try to create the configuration automatically with the command line :

python ./haproxy_config.py --hyperbalance

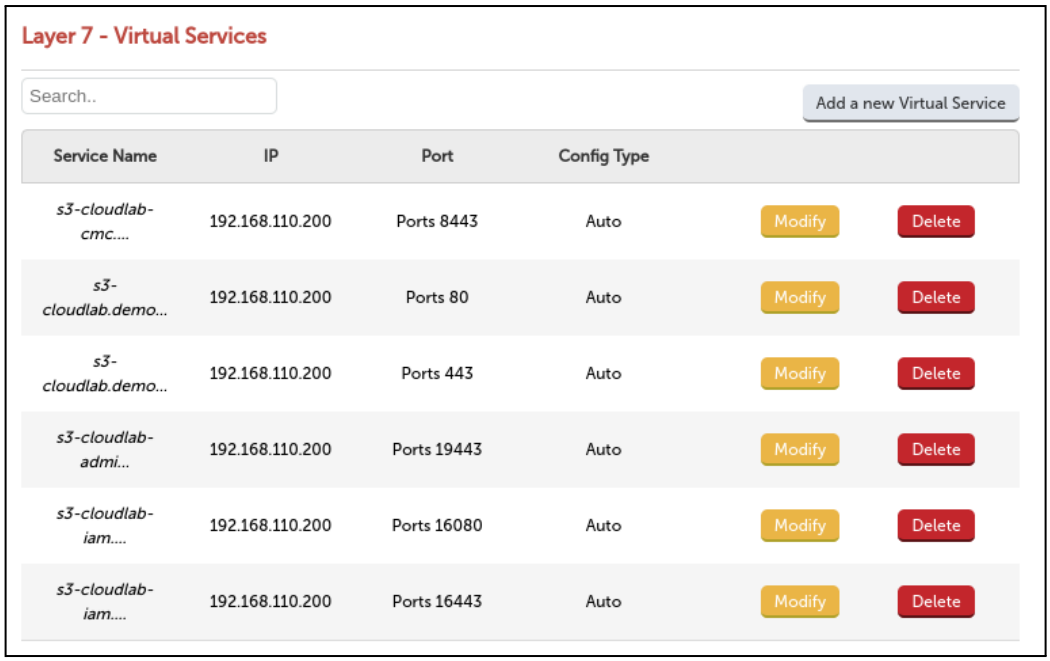
Notice : In the example below, we are using the @IP : 192.168.110.199 for the LB administration and the @IP : 192.168.110.200 as the floating IP.

```
[root@cloudlab01]# python ./haproxy_config.py --hyperbalance
You requested a configuration for a HyperBalance appliance

We are considering the following path as the current path for the cloudian
installation : /opt/cloudian-staging/7.5/

/opt/cloudian-staging/7.5/survey.csv FOUND - OK
/opt/cloudian-staging/7.5/CloudianInstallConfiguration.txt FOUND - OK
/etc/cloudian-7.5-puppet/manifests/extdata/common.csv FOUND - OK

HyperStore release detected : 7.5
You need to have the root access.
Please, enter the IP address of your HyperBalance appliance : 192.168.110.199
Enter the root password for the connection ...
Password:
Trying to connect to the host : 192.168.110.199 with the root password ... and
then checking some parameters for you ...
processing, please wait...
OK. HyperBalance is reachable...
Please, enter the IP address for the VIP (floating IP) : 192.168.110.200
processing, please wait...
HyperBalance configuration is applied.
```

Take a look at the LB GUI, we should have 6 virtual services created and ready to serve requests.



The virtual Services are up and running with the standard configuration from the Cloudian nodes.



WANT TO REVERT THE CONFIGURATION ?

For any reason, you want to revert the configuration pushed on the LB, no problem, use the revert command.

```
[root@cloudlab01]# python ./haproxy_config.py --hbrevert
hb-config-revert.cmd FOUND - OK
You need to have the root access.
Please, enter the IP address of your HyperBalance appliance :
192.168.110.199
Enter the root password for the connection ...
Password:

Trying to connect to the host : 192.168.110.199 with the root password ...
and then checking some parameters for you ...
processing, please wait...
OK. HyperBalance is reachable...
processing, please wait...
HyperBalance configuration removed.
```

Then, if you check your HyperBalancer GUI, everything is clear : no more real servers, no more virtual services. The tool targets only the Cloudian standard configuration applied by itself previously :

- virtual services : CMC, S3, Admin API, IAM
- floating IP for Cloudian
- restart the hyperbalance « haproxy » service