

# Adaptive Load Balancing Framework Setup Guide

Group 21

April 11, 2014

## **Abstract**

Our framework uses two servers for its operation; a load balancer and a load tester. The load balancer runs HAProxy as its load balancing software, and also uses HAProxy stats to provide realtime statistics on the backend load. Munin is used by the load balancer to provide graph information about the load history, and Vegeta is used by the load tester to simulate load on the load balancer.

# 1 Load Balancer

## 1.1 Installing HAProxy

HAProxy can be installed via apt. If Debian Backports are not already listed under apt sources in `/etc/apt/sources.list`, you must add them:

```
# debian backports sources
deb http://mirror.vorboss.net/debian/ wheezy-backports main
deb-src http://mirror.vorboss.net/debian/ wheezy-backports main
```

HAProxy can then simply be installed by running `apt-get install haproxy`

HAProxy must be configured as needed, with all necessary client server information, in `/etc/haproxy/haproxy.cfg`

To start HAProxy, first ensure that `ENABLED=1` is set in `/etc/default/haproxy`, then it can be started using `service haproxy start`

## 1.2 HAProxy Stats

The configuration file must also contain information to start the HAProxy stats page along with the load balancer. To enable the stats page, the following lines must be included:

```
stats enable
stats hide-version
stats refresh 1s
```

This also forces the stats page to refresh with live information every second. The HAProxy stats page will now be live on `/haproxy?stats`. In order for some Web UI algorithm switching scripts to operate, the permissions of the HAProxy config files must be changed. Run `chmod 766 /etc/haproxy/haproxy.cfg` to do this.

## 1.3 Installing Apache and PHP

Apache must be installed if it's not already on the server. Apache is used to serve the Web UI. Install it using `apt-get install apache2`. It will install but fail to start because HAProxy will be bound to port 80. Apache must therefore be configured to run the Web UI on a separate port. In our implementation, we chose port 9001 for the Web UI. Change this configuration in the `/etc/apache2/sites-enabled/000-default` by changing the `VirtualHost` opening tag to:

```
<VirtualHost *:9001>
```

The `/etc/apache2/ports.conf` configuration file must also be changed, by replacing the first two uncommented lines in the file with:

```
NameVirtualHost *:9001
Listen 9001
```

PHP must now be installed: `apt-get install php5-common libapache2-mod-php5 php5-cli`. Apache can now be started using `service apache2 start`.

## 1.4 Installing the Web UI

The Web UI may now be installed on the load balancer. All files in the `loadbalancer` subdirectory of the source folder should be placed in `/var/www`. Any pre-existing files in this directory should be removed.

The IP addresses of the servers intended to run the framework should be included in the source files. This should be done by replacing all occurrences of `_LOADBALANCER_` and `_LOADTESTER_` in the source files with the IPs of the intended deployment servers.

## 1.5 Installing Munin

Munin must now be installed so that graphical load visualisation output can be displayed by the Web UI.

First, the Debian non-free libraries must be added to the apt sources. The following changes must be made to `/etc/apt/sources.list`:

```
change deb http://ftp.ie.debian.org/debian/ wheezy main to
deb http://ftp.ie.debian.org/debian/ wheezy main non-free, and change
deb http://security.debian.org/ wheezy/updates main to
deb http://security.debian.org/ wheezy/updates main contrib non-free.
```

Now reload apt sources with `apt-get update`.

Munin can now be installed by running:

```
apt-get install munin munin-node libapache2-mod-fastcgi
```

And then:

```
a2enmod fastcgi
```

The following symlink must also be added:

```
ln -s /usr/share/munin/plugins/haproxy_ng /etc/munin/plugins/haproxy_ng
```

Add the following lines to `/etc/munin/plugin-conf.d/munin-node`:

```
[haproxy_ng]
env.url http://localhost/haproxy?stats;csv;norefresh
```

HAProxy's config file also need to be configured to work with Munin. The following line must be added to HAProxy's global configuration section, in the `/etc/haproxy/haproxy.cfg` file:

```
stats socket /tmp/haproxy
```

And finally add the following configuration to `munin.conf`:

```
html_strategy cgi
graph_strategy cgi
```

Now copy the provided `apache.conf` into `/etc/munin/apache.conf`, overwriting the default one. Apache and munin-node must be restarted for the Munin installation to take effect. Run:

```
service apache2 restart
service munin-node restart
```

## 1.6 Backend Configuration

Due to the nature of the backend setup, some additional configuration is necessary for all features of the framework to function as expected.

Socat must be installed in order to retrieve HAProxy stats from a Unix socket. Install using `apt-get install socat`.

Finally, install sudo using `apt-get install sudo`. Some additions must be made to the sudo config file. These can be performed by editing `/etc/sudoers/` (or by running `visudo`), and adding the following lines to the file:

```
www-data ALL=NOPASSWD: /etc/init.d/haproxy
www-data ALL=NOPASSWD: /var/www/files/swap_files/graceful_reload.sh
www-data ALL=NOPASSWD: /usr/bin/socat
```

## 2 Load Tester

The load tester setup is relatively simple, as it is accessed by POST requests from the load balancer, and displays an `index.html` page for load test result output.

Please refer to section 1.3 to set up Apache and PHP on the load testing server. Then move all files from the `loadtester` folder in the code bundle to `/var/www` to complete the installation.