HAProxy

Installation

- add the GPG key for the repository for AWS-LC enabled HAProxy which is required for HTTP/3:

```
sudo wget -qO /usr/share/keyrings/HAPROXY-key-community.asc
https://pks.haproxy.com/linux/community/RPM-GPG-KEY-HAProxy
```

- add the HAProxy 3.4 performance repository for Debian 13 (Trixie) - note that the Debian 13 repository contains HAProxy 3.0:

```
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/HAPROXY-key-
community.asc]
https://www.haproxy.com/download/haproxy/performance/debian/ha34 trixie
main" | sudo tee /etc/apt/sources.list.d/haproxy.list
```

- update the repository:

```
sudo apt update && sudo apt upgrade -y
```

- Install HAProxy dedicated AWS-LC package:

```
sudo apt install haproxy-awslc -y
```

- Check the installed version:

```
haproxy -v
```

- Start and enable the service:

```
sudo systemctl start haproxy
sudo systemctl enable haproxy
```

- Replace the contents of your `/etc/haproxy/haproxy.cfg` file with this HAProxy Configuration configuration.
- Add Whitelist and Blacklist files.
- Check the configuration file:

```
sudo haproxy -c -f /etc/haproxy/haproxy.cfg
```

- Restart the proxy:

```
sudo systemctl restart haproxy
```

- Open port 443 for TCP **and** UDP in your firewall. UDP is required for HTTP/3.
- If you run a mail server behind the proxy you might run into DNS/hairpin NAT issues. You best solve this by adding your mail host to the host overrides table. In pfSense firewall this is at *Services –> DNS Resolver –> General Settings*, scroll down to the bottom of the page.
- If you moved a web site from an old to a new server behind HAProxy you must check persistent

or long-lived connections
    - Check whether the web site is still accessed (occasionally) on the old server:

```
tail -f /var/log/apache2/your-site-access.log
```

    - Remove the site configuration file:

```
rm /etc/apache2/sites-enabled/0xx-your-site.conf
```

    - Check your HAProxy `timeout http-keep-alive` or `timeout client/server` settings
    - Temporarily add `option http-server-close` to the backend server definition in `/etc/haproxy/haproxy.cfg` during transition

SSL certificates

- Install and configure acme.sh to set up automatic Let's Encrypt certificate generation
- If you moved the certificate from a backend web server to HAProxy disable certificate renewal in the backend web server or delete the certificate:

```
sudo certbot delete --cert-name yourdomain.com
/etc/letsencrypt/renewal/mv yourdomain.com.conf
/etc/letsencrypt/renewal/yourdomain.com.disabled
```

- During transition from an old web server not behind HAProxy to a new proxied server you might need to still be able to renew certificates on the old server. Create the following alternative `haproxy.cfg.certbot` config file with a modified rule *frontend public_inbound_http* and a new backend rule *backend http_server_certbot* and copy it to `haproxy.cfg`:

```
frontend public_inbound_http
    bind *:80
    mode http
    default_backend http_server_certbot
backend http_server_certbot
    mode http
    cookie SERVERID insert indirect nocache
    server calypso 192.168.3.101:80 check
```

- Reload the proxy rules:

```
sudo systemctl reload haproxy
```

- SSH into your old server and run `sudo certbot renew`
- After you have successfully renewed the certificates restore the production `haproxy.cfg` and reload the proxy rules
- Note that access to your web sites is always guaranteed through HTTPS, but you create a small window where your sites could be accessed through HTTP directly. If your transition period last longer than a few days you might consider to automate this *hack* to automatically renew certificates on the old server.

Apache2 config

- Apache requires the *mod_remoteip* module to intercept the header and overwrite the connection IP. Enable the module:

```
sudo a2enmod remoteip
```

- Configure the module. Create or edit `/etc/apache2/conf-available/remoteip.conf` and define your HAProxy server as a trusted internal proxy:

```
RemoteIPHeader X-Forwarded-For
RemoteIPInternalProxy 192.168.3.50  # this is your HAProxy private IP
SetEnvIf X-Forwarded-Proto "https" HTTPS=on # Ensure Apache changes the
internal HTTPS environment variable if HAProxy says so
```

- Enable the configuration:

```
sudo a2enconf remoteip
```

- Apache's default log format uses %h (remote host), which still logs HAProxy's IP. You must change it to %a (client IP address).Edit your main Apache configuration or your VirtualHost file. Look for the LogFormat lines and modify %h to %a:

```
LogFormat "%a %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-
Agent}i\"" combined
```

- Test your config and restart Apache:

```
sudo apache2ctl configtest
sudo systemctl restart apache2
```

- If HAProxy is forwarding raw TCP streams (e.g., passing SSL directly to Apache), it cannot inject an HTTP header. You must use the PROXY protocol instead. Append `send-proxy` or `send-proxy-v2` to your backend server line:

```
backend passthrough_server
    mode tcp
    server web1 192.168.3.90:443 send-proxy check
```

- **If not evaluated properly this will break access to sites on such a passthrough server**. Since I block access and check IP addresses at the proxy server at the earliest possible moment (e.g. `tcp-request connection reject`) I chose a different approach. I linked a temporary domain name to the backend server and transfer HTTP to access restricted sub-pages like phpmyadmin. For the configuration follow the settings used for *backuppc*.
- Restart HAProxy

```
sudo systemctl restart haproxy
```

- On your apache server ensure *mod_remoteip* is enabled. Open your Apache <VirtualHost> configuration and define the `RemoteIPProxyProtocol` directive:

```
<VirtualHost *:443>
```

```
    ServerName example.com

    # Enable Proxy Protocol interpretation
    RemoteIPProxyProtocol On
    RemoteIPProxyProtocolExceptions 127.0.0.1

    # ... your SSL and document root settings ...
</VirtualHost>
```

- Update your LogFormat to use %a instead of %h as described above, then restart Apache.
- To be able to restrict access to domains with query string, which is not possible in HAProxy, you need to add IP addresses to a apache config file, for example `/etc/apache2/conf-available/bco-trusted-ip.conf`:

```
<RequireAny>
    # Local subnet
    Require ip 192.168.1
    # Fixed ip (if you have any)
    Require ip my.fixed.ip.addr
</RequireAny>
```

- To add an IP address for a remote computer not in one of your trusted IP ranges create an empty file `/etc/apache2/conf-available/bco-current-ip.conf`.
- Add an `Include` to every directory you want to restrict access to in your <VirtualHost> file:

```
...
<Directory /usr/share/phpmyadmin>
    Include conf-available/bco-trusted-ip.conf
    Include conf-available/bco-current-ip.conf
</Directory>
...
```

- On the computer you need to get access for add the following command to update and remove your current IP address:

```
IP=$(curl -s https://api.ipify.org) && ssh -p 50322 your.domain.name
"echo \"Require ip $IP\" > /etc/apache2/conf-available/bco-current-
ip.conf && sudo /usr/bin/systemctl reload apache2"
ssh -p 22 your.domain.name "echo \"\" > /etc/apache2/conf-
available/bco-current-ip.conf && sudo /usr/bin/systemctl reload
apache2"
```

- To make this work you need to add <user> to file `apache` in `/etc/sudoers.d` so <user> can reload the apache config as root without password:

```
vim /etc/sudoers.d/apache
<user> ALL=(root) NOPASSWD: /usr/bin/systemctl reload apache2
```

- Update the IP from another linux computer or WSL with the following file:

```
#!/bin/bash
# bco, 2026-09-20
```

```
# update my current IP address in Apache config
#
IP=$(curl -s https://api.ipify.org)

# Regex pattern matching 0-255 for individual octets
octet='(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[1-9][0-9]|[0-9])'
pattern="^${octet}\.${octet}\.${octet}\.${octet}$"

if [[ $IP =~ $pattern ]]; then
    ssh -p 22 your.domain.name "echo \"Require ip $IP\" >
/etc/apache2/conf-available/bco-current-ip.conf && sudo
/usr/bin/systemctl reload apache2"
else
    echo "Invalid IP address"
    sleep 3
    exit 1
fi
exit 0
```

- Put a shortcut on the Windows Desktop for easy access:

```
C:\Windows\System32\wsl.exe bash ~/batch/updateApacheIP
```

- You should also put a shortcut on the Windows Desktop to remove the IP address after you don't need access anymore or move somewhere else.
- This way you can add your computer to the trusted IP list (whitelist) no matter what is the current IP address.

Joomla6 config

-
```
// Tells Joomla to trust proxy headers (X-Forwarded-For, X-Forwarded-
Proto)
public $behind_loadbalancer = '1';

// Forces Joomla to generate secure HTTPS URLs globally
public $force_ssl = '2';

// Optional: If Joomla still struggles to map assets correctly,
explicitly define your live site URL
public $live_site = 'https://yourdomain.com';
```

Links

- What Is a Reverse Proxy and Why Do You Need One?
- Configuration is explained in HAProxy configuration file: global, defaults, front and backend

From:
https://wiki.condrau.com/ - **Bernard's Wiki**

Permanent link:
https://wiki.condrau.com/deb13:haproxy

Last update: **2026/10/05 13:37**