Allgemeine Konfiguration

https://cbonte.github.io/haproxy-dconv/2.1/configuration.html

HAPROXY als SSL Accelerator

Vorteil: **keine separaten IP Adressen nötig!** Konzept: HAPROXY horcht auf 80 und 443, der Webserver horcht nur auf localhost, z.B: 127.0.0.1:82, 83, 84... HAPROXY erkennt Requests über den Host Header und leitet auf den entsprechenden Port auf localhost um. Zertifikate werden durch haproxy bereitgestellt ===== Frontend =====

```
frontend default
 bind *:80
 # Add multiple certificates, one for each domain.tld
 bind *:443 ssl crt /etc/ssl/haproxy/devel.netzwissen.de.pem crt
/etc/ssl/haproxy/passbolt.netzwissen.de.pem crt /etc/ssl/haproxy/amrae.d$
 mode http
 # global redirect to https
 redirect scheme https code 301 if !{ ssl_fc }
```

ACL und Weiterleitung

```
## ACL for each Subdomain to terminate
 acl devel-acl hdr(host) -i devel.netzwissen.de
 acl le-acl path_beg /.well-known/acme-challenge/
 acl passbolt-acl hdr(host) -i passbolt.netzwissen.de
 acl amrae-acl hdr(host) -i amrae.de
 acl gruenerheiner-acl hdr(host) -i gruenerheiner.netzwissen.de
 acl freifunk-esslingen-acl hdr(host) -i freifunk-esslingen.de

## BACKEND: Use Backend Section
 use_backend devel if devel-acl
 use_backend le-backend if le-acl
 use_backend passbolt if passbolt-acl
 use_backend amrae if amrae-acl
 use_backend gruenerheiner if gruenerheiner-acl
 use_backend freifunk-esslingen if freifunk-esslingen-acl
```

Backend Beispiel

```
backend passbolt
   mode http
```

```
   server passbolt.netzwissen.de 127.0.0.1:83 check
   http-request set-header X-Forwarded-Port %[dst_port]
   http-request add-header X-Forwarded-Proto https if { ssl_fc }
```

Lets Encrypt und HAPROXY

https://serversforhackers.com/c/letsencrypt-with-haproxy

Konfiguration HAPROXY

frontend

```
frontend le-frontend
   bind *:80
   # Test URI to see if its a letsencrypt request
   acl letsencrypt-acl path_beg /.well-known/acme-challenge/
   use_backend le-backend if letsencrypt-acl
   default_backend devel
```

backend

```
backend le-backend
    mode http
    server letsencrypt 138.201.52.38:8888
```

Neue Zertifikate bestellen

```
certbot certonly --standalone -d zammad.netzwissen.de --non-interactive --
agree-tos --email admin@netzwissen.de --http-01-port=8888
```

- `-standalone` - Create a stand-alone web server to listen for the cert authorization HTTP request
- `-d [domain]` - The domain we're creating a cert for. You can use multiple `-d` flags for multiple domains for a single certificate. The domain(s) must route to the server we're creating a cert for (DNS must be setup for the domain).
- `–non-interactive –agree-tos –email admin@example.com` - Make this non-interactive by saying as much, agreeing to the TOS, and informing LetsEncrypt of the email to use to send "YOUR CERT IS EXPIRING" notifications.
- `–http-01-port=8888` - The Magic™. This tells the stand-alone server to listen on port 8888. Note that LetsEncrypt will *still* send the authorization HTTP request over port **80** and haproxy with redirrect requests over port 8888. The flag is `http-01` because it expects an HTTP request, NOT an HTTPS request.

Zertifikate

cert und key müssen in **einer Datei** vorliegen:

```
''cat /etc/letsencrypt/live/demo.scalinglaravel.com/fullchain.pem \
    /etc/letsencrypt/live/demo.scalinglaravel.com/privkey.pem \
    | sudo tee
/etc/ssl/demo.scalinglaravel.com/demo.scalinglaravel.com.pem''
```

==== Erneuern ====

```
''sudo certbot renew --tls-sni-01-port=8888''
```