

HAProxy Config Files

HAProxy

Installation

```
/etc/haproxy/haproxy.cfg
# 0. Settings
global
    log /dev/log local0
    log /dev/log local1 notice
    chroot /var/lib/haproxy
    user haproxy
    group haproxy
    daemon

    # Modern SSL/TLS security settings for Debian 13
    # Minimum TLS version requirement
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

    # TLS 1.3 Cipher Suites (handled by ssl-default-bind-ciphersuites)
    ssl-default-bind-ciphersuites
    TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256

    # TLS 1.2 Ciphers (handled by ssl-default-bind-ciphers)
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-
    GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-
    ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-
    SHA256:DHE-RSA-AES256-GCM-SHA384

    # Administrative socket for stats/management
    stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd
listeners
    stats timeout 30s

defaults
    log          global
    mode         tcp
    log-format   "%ci:%cp [%t] %ft %b/%s %Tw/%Tc/%Tt %B %ts
    %ac/%fc/%bc/%sc/%rc %sq/%bq"
    timeout connect 5s
    timeout client  50s
    timeout server  50s

# 1. Public Interface (SNI Router)
frontend public_inbound_http
    bind *:80
```

```
mode http

# Deny traffic based on origin
http-request deny if { src -f /etc/haproxy/blacklist.lst }

# Redirect all incoming traffic to HTTPS permanently
http-request redirect scheme https code 301 unless { ssl_fc }

frontend public_inbound
  bind :443
  mode tcp
  # Deny traffic based on origin
  tcp-request connection reject if { src -f /etc/haproxy/blacklist.lst }

  # Wait for the TLS client hello to parse the SNI extension
  tcp-request inspect-delay 5s
  tcp-request content accept if { req.ssl_hello_type 1 }

  # Define the domains (one line for each) you want to terminate SSL for
  acl terminate_ssl req.ssl_sni -i bak.condrau.com

  # Route conditionally based on SNI
  use_backend http_dispatcher if terminate_ssl
  default_backend passthrough_server

# 2. Transparent passthrough for encrypted TLS domains
backend passthrough_server
  mode tcp

  # downstream server handling its own TLS certificates (Calypso)
  server calypso 192.168.3.18:443 check

# 3. Intermediary loopback backend
# This forwards traffic to the Layer 7 (HTTP) processing frontend
backend http_dispatcher
  mode tcp
  server loopback 127.0.0.1:8443 send-proxy-v2

# 4. Layer 7 http frontend
# Receives traffic from the loopback, decrypts SSL, and processes HTTP rules
frontend internal_inbound_http

  # pass certs directory path so all pem files will be loaded
  bind 127.0.0.1:8443 ssl crt /etc/haproxy/certs/ accept-proxy
  mode http

  # Deny traffic based on origin (host aware)
  acl trusted_ip src -f /etc/haproxy/whitelist.lst
  acl backuppc hdr(host) -i bak.condrau.com
  http-request deny if backuppc !trusted_ip
```

```
# Inject headers so backend servers know the original request was HTTPS
http-request set-header X-Forwarded-Proto https
http-request set-header X-Forwarded-Port 443
http-request set-header X-Forwarded-For %[src]
```

```
# Route traffic based on HTTP host headers
use_backend backuppc_server if backuppc
# default_backend backuppc_server
```

5. HTTP backends

```
backend backuppc_server
    mode http
    cookie SERVERID insert indirect nocache
    server epione 192.168.1.14:8080 check
```

From:

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

Permanent link:

<https://wiki.condrau.com/deb13:haconfig?rev=1789200075>

Last update: **2026/09/12 15:01**

