

Как распределять нагрузку с помощью НАProху

Инструкция для администраторов

Оглавление

Назначение документа	3
Установка и запуск HAProxy	3
Настройка HAProxy	3
Пример конфигурационного файла	4

Назначение документа

HAProxy – программное обеспечение с открытым исходным кодом, которое обеспечивает высокодоступный балансировщик нагрузки и прокси-сервер для приложений на основе TCP и HTTP, распределяющих запросы между несколькими серверами.

В документе описано как распределять нагрузку на Почту с помощью HAProxy.

Установка и запуск HAProxy

HAProxy нужно установить и запустить на отдельном сервере, то есть на сервере не должны быть запущены сервисы Почты. Чтобы установить и запустить HAProxy для ОС Astra Linux:

1. Установите пакет haproxy:

```
sudo apt install haproxy
```

2. Разрешите автозапуск HAProxy:

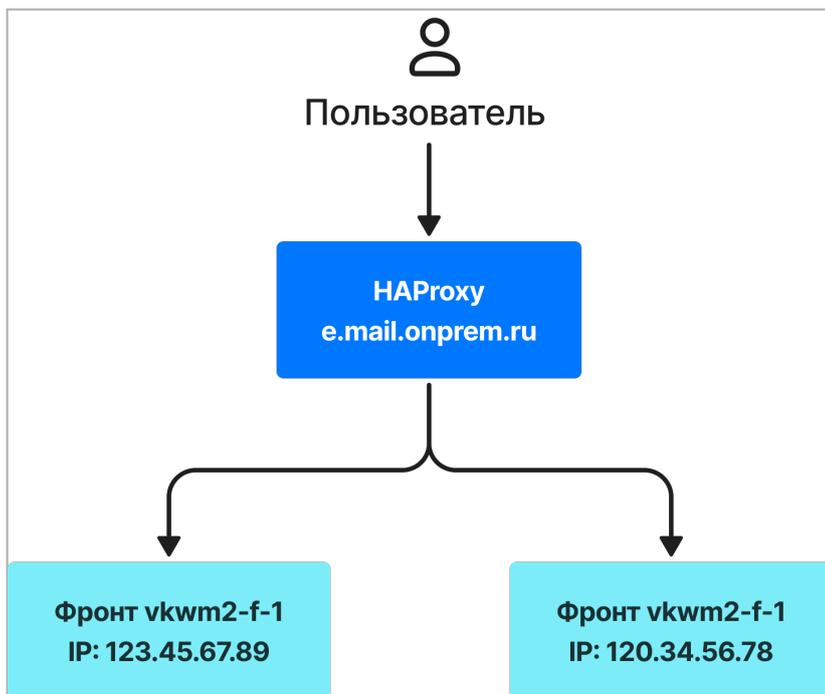
```
systemctl enable haproxy
```

3. Запустите сервис:

```
systemctl start haproxy
```

Настройка HAProxy

При балансировке нагрузки через HAProxy нужно настроить DNS-записи на сервер с HAProxy. Рассмотрим установку Почты на 8 машин, ниже представлена схема работы при балансировке через HAProxy:



По умолчанию конфигурационный файл находится по пути `/etc/haproxy/haproxy.cfg`. Ниже вы найдете [пример конфигурационного файла](#). Этот пример можно взять за основу конфигурации под вашу установку. В конфигурационном файле объявляются IP-адреса backend'ов, с которыми будет взаимодействовать HAProxy:

```
backend be_VK_pop3
  mode tcp
  balance source
  server fe01 *.*.*.*:110 maxconn 10000
  server fe02 *.*.*.*:110 maxconn 10000
  http-request set-header X-REAL-IP %[src]
```

В строках `server fe01 *.*.*.*:110 maxconn 10000` вместо `*.*.*.*` нужно передать IP-адреса машин, отведенных под фронт.

Для проверки состояния сервисов веб-доступа в систему вы можете воспользоваться ссылкой `https://ip-address`, где `ip-address` - это адреса фронтов Почты. В зависимости от ответа необходимо выбирать доступный сервер.

Пример конфигурационного файла

```
global
  maxconn 100000
  log /dev/log    local0
  log /dev/log    local1 notice
  chroot /var/lib/haproxy
  stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
  stats timeout 30s
  user haproxy
  group haproxy
  daemon
```

```
ca-base /etc/ssl/certs
crt-base /etc/ssl/private
```

```
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
ssl-default-bind-ciphersuites
TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets
```

defaults

```
log global
mode http
option httplog
option dontlognull
option forwardfor
```

```
retries 3
timeout connect 5s
timeout http-keep-alive 1s
timeout http-request 15s
timeout queue 30s
timeout client 300s
timeout server 300s
```

```
errorfile 400 /etc/haproxy/errors/400.http
errorfile 403 /etc/haproxy/errors/403.http
errorfile 408 /etc/haproxy/errors/408.http
errorfile 500 /etc/haproxy/errors/500.http
errorfile 502 /etc/haproxy/errors/502.http
errorfile 503 /etc/haproxy/errors/503.http
errorfile 504 /etc/haproxy/errors/504.http
```

listen stats

```
bind :1111
mode http
option forwardfor
option httpclose
stats enable
stats uri /
stats refresh 5s
stats show-legends
stats realm Haproxy\ Statistics
stats auth admin:1password2$
```

frontend fe_VK-http

```
mode tcp
bind *:80
use_backend be_VK_http
```

frontend fe_VK_https

```
mode tcp
bind *:443
use_backend be_VK_https
```

frontend fe_VK-pop3

```
mode tcp
option tcplog
bind *:110
use_backend be_VK_pop3
```

frontend fe_VK-pop3s

```
mode tcp
```

```
option tcplog
bind *:995
use_backend be_VK_pop3s

frontend fe_VK-imap4
mode tcp
option tcplog
bind *:143
use_backend be_VK_imap4

frontend fe_VK-imap4s
mode tcp
option tcplog
bind *:993
use_backend be_VK_imap4s

frontend fe_VK-smtp
mode tcp
option tcplog
bind *:25
use_backend be_VK-smtp

frontend fe_VK-smtp1025
mode tcp
option tcplog
bind *:1025
use_backend be_VK-smtp

frontend fe_VK-smtp465
mode tcp
option tcplog
bind *:465
use_backend be_VK-smtp465

backend be_VK_https
mode tcp
balance source
server fe01 *.*.*.:443 maxconn 10000
server fe02 *.*.*.:443 maxconn 10000
http-request set-header X-REAL-IP %[src]

backend be_VK_http
mode tcp
balance source
http-request set-header X-REAL-IP %[src]

backend be_VK_pop3
mode tcp
balance source
server fe01 *.*.*.:110 maxconn 10000
server fe02 *.*.*.:110 maxconn 10000
http-request set-header X-REAL-IP %[src]

backend be_VK_pop3s
mode tcp
balance source
server fe01 *.*.*.:995 maxconn 10000
server fe02 *.*.*.:995 maxconn 10000
http-request set-header X-REAL-IP %[src]

backend be_VK_imap4
mode tcp
balance source
```

```
server fe01 *.*.*.*:143 maxconn 10000
server fe02 *.*.*.*:143 maxconn 10000
http-request set-header X-REAL-IP %[src]
```

```
backend be_VK_imap4s
mode tcp
balance source
server fe01 *.*.*.*:993 maxconn 10000
server fe02 *.*.*.*:993 maxconn 10000
http-request set-header X-REAL-IP %[src]
```

```
backend be_VK_smtp
mode tcp
balance source
server fe01 *.*.*.*:25 maxconn 10000
server fe02 *.*.*.*:25 maxconn 10000
http-request set-header X-REAL-IP %[src]
```

```
backend be_VK_smtp1025
mode tcp
balance source
server fe01 *.*.*.*:1025 maxconn 10000
server fe02 *.*.*.*:1025 maxconn 10000
http-request set-header X-REAL-IP %[src]
```

```
backend be_VK_smtp465
mode tcp
balance source
server fe01 *.*.*.*:465 maxconn 10000
server fe02 *.*.*.*:465 maxconn 10000
http-request set-header X-REAL-IP %[src]
```

 Автор: Груздев Никита

 18 декабря 2024г.