

Настройка HAProxy для HA-кластера JumpServer

HAProxy (High Availability Proxy) — это **программный инструмент с открытым исходным кодом** для балансировки нагрузки и проксирования трафика на уровне сетевого протокола, обычно используемый для распределения нагрузки между несколькими серверами. Он является одним из самых популярных решений для повышения доступности и производительности веб-приложений и служб.

Установка HAProxy, на примере Ubuntu:

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

После установки нужно отредактировать файл конфигурации, это вызывает основные вопросы по настройке.

Файл конфигурации обычно находится в **/etc/haproxy/haproxy.cfg**

Пример файла конфигурации с сайта вендора:

```
global
    # to have these messages end up in /var/log/haproxy.log you will
    # need to:
    #
    # 1) configure syslog to accept network log events. This is done
    #    by adding the '-r' option to the SYSLOGD_OPTIONS in
    #    /etc/sysconfig/syslog
    #
    # 2) configure local2 events to go to the /var/log/haproxy.log
    #    file. A line like the following can be added to
    #    /etc/sysconfig/syslog
    #
    #    local2.*                /var/log/haproxy.log
    #
log    127.0.0.1 local2

chroot    /var/lib/haproxy
pidfile   /var/run/haproxy.pid
maxconn   4000
user      haproxy
group     haproxy
daemon

# turn on stats unix socket
stats socket /var/lib/haproxy/stats
```

```

#-----
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#-----

defaults
    log                global
    option              dontlognull
    option              redispatch
    retries             3
    timeout http-request 10s
    timeout queue       1m
    timeout connect     10s
    timeout client      1m
    timeout server      1m
    timeout http-keep-alive 10s
    timeout check       10s
    maxconn             3000

listen stats
    bind *:8080
    mode http
    stats enable
    stats uri /haproxy          # Страница мониторинга, измените по необходимости: http://
192.168.100.100:8080/haproxy
    stats refresh 5s
    stats realm haproxy-status
    stats auth admin:KXOeyNgDeTdpeu9q    # # Учетная запись и пароль для доступа к http://
192.168.100.100:8080/haproxy

#-----
# check  Параметры проверки активности
# inter  Интервал времени, единица: миллисекунды
# rise   Количество последовательных успешных попыток, единица: раз
# fall   Количество последовательных неудачных попыток, единица: раз
# Пример: inter 2s rise 2 fall 3
# Означает проверку состояния каждые 2 секунды, 2 последовательных успеха указывают на
нормальную работу сервиса, 3 последовательных неудачи указывают на ошибку в сервисе
#
# server  Параметры сервера
# server 192.168.100.21 192.168.100.21:80 weight 1 cookie web01
# Первый 192.168.100.21 используется как идентификатор для отображения, его можно
заменить на любой другой строковый идентификатор
# Второй 192.168.100.21:80 — это фактический адрес и порт задействованного сервиса
# weight указывает на вес сервера, при наличии нескольких узлов балансировка нагрузки
будет зависеть от веса
# cookie определяет, что на стороне клиента в cookie будет содержаться этот идентификатор,
чтобы различать текущий используемый задний узел
# Пример: server db01 192.168.100.21:3306 weight 1 cookie db_01

```

#-----

listen jms-web

bind *:80 # Прослушивание порта 80

mode http

redirect scheme https if !{ ssl_fc } # Перенаправление на https

bind *:443 ssl crt /opt/ssl.pem # Настройка https

option httpchk GET /api/health/ # Интерфейс проверки активности Core

stick-table type ip size 200k expire 30m

stick on src

balance leastconn

server 192.168.100.21 192.168.100.21:80 weight 1 cookie web01 check inter 2s rise 2 fall 3 #

JumpServer 07A1E8

server 192.168.100.22 192.168.100.22:80 weight 1 cookie web02 check inter 2s rise 2 fall 3

server 192.168.100.23 192.168.100.23:80 weight 1 cookie web03 check inter 2s rise 2 fall 3

server 192.168.100.24 192.168.100.24:80 weight 1 cookie web03 check inter 2s rise 2 fall 3

listen jms-ssh

bind *:2222

mode tcp

option tcp-check

fullconn 500

balance source

server 192.168.100.21 192.168.100.21:2222 weight 1 check inter 2s rise 2 fall 3 send-proxy

server 192.168.100.22 192.168.100.22:2222 weight 1 check inter 2s rise 2 fall 3 send-proxy

server 192.168.100.23 192.168.100.23:2222 weight 1 check inter 2s rise 2 fall 3 send-proxy

server 192.168.100.24 192.168.100.24:2222 weight 1 check inter 2s rise 2 fall 3 send-proxy

listen jms-koko

mode http

option httpclose

option forwardfor

option httpchk GET /koko/health/ HTTP/1.1\r\nHost:\ 192.168.100.100 # KoKo 0800A3E3, host 0800

HAProxy 69 ip 5606

cookie SERVERID insert indirect

hash-type consistent

fullconn 500

balance leastconn

server 192.168.100.21 192.168.100.21:80 weight 1 cookie web01 check inter 2s rise 2 fall 3

server 192.168.100.22 192.168.100.22:80 weight 1 cookie web02 check inter 2s rise 2 fall 3

server 192.168.100.23 192.168.100.23:80 weight 1 cookie web03 check inter 2s rise 2 fall 3

server 192.168.100.24 192.168.100.24:80 weight 1 cookie web03 check inter 2s rise 2 fall 3

```
listen jms-lion
  mode http

  option httpclose
  option forwardfor
  option httpchk GET /lion/health/ HTTP/1.1\r\nHost:\ 192.168.100.100 # Lion 88903E3, host 8890
HAProxy 74 ip 5757

  cookie SERVERID insert indirect
  hash-type consistent
  fullconn 500
  balance leastconn
  server 192.168.100.21 192.168.100.21:80 weight 1 cookie web01 check inter 2s rise 2 fall 3
  server 192.168.100.22 192.168.100.22:80 weight 1 cookie web02 check inter 2s rise 2 fall 3
  server 192.168.100.23 192.168.100.23:80 weight 1 cookie web03 check inter 2s rise 2 fall 3
  server 192.168.100.24 192.168.100.24:80 weight 1 cookie web03 check inter 2s rise 2 fall 3

listen jms-magnus
  bind *:30000
  mode tcp

  option tcp-check

  fullconn 500
  balance source
  server 192.168.100.21 192.168.100.21:30000 weight 1 check inter 2s rise 2 fall 3 send-proxy
  server 192.168.100.22 192.168.100.22:30000 weight 1 check inter 2s rise 2 fall 3 send-proxy
  server 192.168.100.23 192.168.100.23:30000 weight 1 check inter 2s rise 2 fall 3 send-proxy
  server 192.168.100.24 192.168.100.24:30000 weight 1 check inter 2s rise 2 fall 3 send-proxy
```

После изменения файла конфигурации запустите службу haproxy:

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

Версия #1
Сергей Попцов создал 30 октября 2024 12:56:39
Сергей Попцов обновил 30 октября 2024 13:08:41