

Lab – Balanceamento de carga de aplicações TCP com HAProxy

Tópicos

1. Enquadrar o “haproxy” no âmbito das soluções de balanceamento de carga de pedidos TCP.
2. Instalar e configurar o serviço HAProxy e validar o funcionamento num cenário de balanceamento de pedidos HTTP.
3. Identificar as várias opções de parametrização do serviço.
4. Efetuar operações de troubleshooting ao funcionamento do serviço de balanceamento de carga e ao serviço balanceado.
5. Aceder à interface gráfica de estatísticas do HAProxy e interpretar os resultados obtidos.
6. Desativar servidores e avaliar o impacto no desempenho do balanceador e do serviço.

1. Cenário prático

A Figura 1 ilustra uma rede constituída por duas VMs que irão balancear os pedidos efetuados ao servidor web Apache2, configurado no porto TCP 80. O sistema hospedeiro é igualmente uma máquina Linux e assumirá as funções de “balanceador de carga” através da aplicação HAProxy.

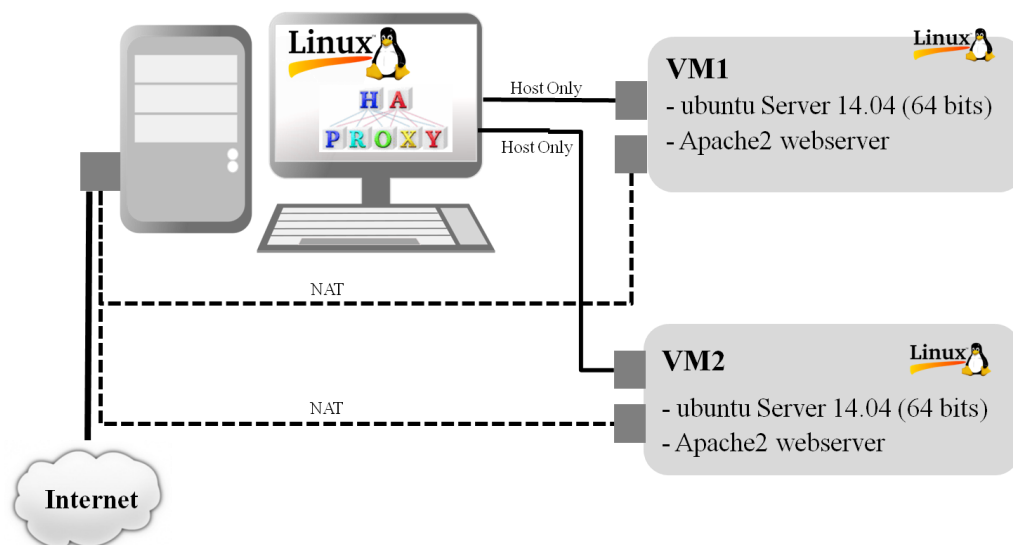


Figura 1 –Cenário prático de balanceamento de carga com HAProxy instalado no sistema hospedeiro

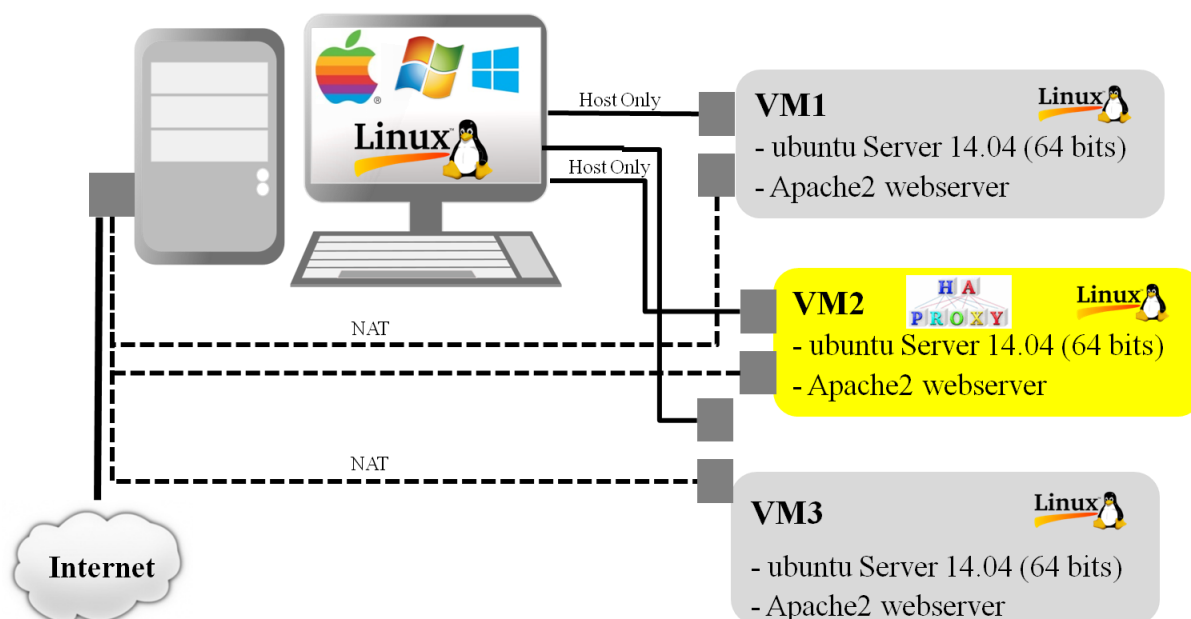


Figura 2 –Cenário prático de balanceamento de carga com HAProxy instalado numa VM.

No caso do sistema hospedeiro ser Microsoft Windows, sugere-se que implemente um cenário com três VMs, sendo que uma delas desempenhará a função de balanceador e as restantes serão os atendedores de pedidos HTTP. Este cenário está ilustrado na Figura 2.

2. Atividades

Descreve-se de seguida um conjunto de atividades para iniciar e utilizar o serviço HAProxy, tendo como ponto de partida o cenário apresentado nas Figuras 1 e 2.

1. Instalar e configurar o servidor aplicativo Apache2 nos servidores Linux de “backend”. Para efeitos de teste, sugere-se uma configuração básica do servidor Apache2, idêntica em todos os servidores Linux de “backend”.
2. Instalar o balanceador HAProxy no servidor Linux de “frontend”, através do comando:
`sudo apt-get update; sudo apt-get install haproxy`
3. Configurar o serviço HAProxy.
 - Definir a flag `ENABLE=1` no ficheiro `/etc/default/haproxy`. Desta forma o serviço HAProxy poderá ser gerido pelo comando “service”.

- Editar o ficheiro `/etc/haproxy/haproxy.cfg` e parametrizar as secções “global”, “defaults”, “listen”, “frontend” e “backend”.

A secção “global” define um conjunto de parâmetros gerais ao funcionamento do HAProxy. Quanto à secção “defaults”, define um conjunto de parâmetros globais que serão usados pelos restantes contextos de configuração do HAProxy.

A secção “frontend” inclui a configuração do HAProxy e a forma como processará os pedidos dos clientes. Já na secção “backend” é possível identificar os servidores para onde serão encaminhados os pedidos dos clientes. Por fim, a secção “listen” permite agrupar o conteúdo das secções “frontend” e “backend”, substituindo-as no ficheiro de configuração.

Apresenta-se de seguida um exemplo das secções “frontend” e “backend” do ficheiro de configuração do HAProxy. Neste exemplo o endereço IP de serviço é o 192.168.1.5 e o serviço balanceado é o que está registado no porto 80.

frontend LB

```
bind 192.168.1.5:80
default_backend LB
```

backend LB 192.168.1.5:80

```
mode http
stats enable
stats hide-version
stats uri /stats
stats auth htproxy:htproxy
balance roundrobin
(...)
server web1 192.168.1.1:80 cookie web1 check
server web2 192.168.1.2:80 cookie web2 check
server web3 192.168.1.3:80 cookie web3 check
(...)
```

Este exemplo pode ser simplificado, incluindo as configurações anteriores numa única secção “listen”, conforme se ilustra de seguida:

listen LB 192.168.1.5:80

```
mode http
stats enable
stats hide-version
stats uri /stats
stats auth htproxy:htproxy
balance roundrobin
(...)
server web1 192.168.1.1:80 cookie web1 check
server web2 192.168.1.2:80 cookie web2 check
server web3 192.168.1.3:80 cookie web3 check
(...)
```

- De salientar que os servidores web terão de estar disponíveis através dos endereços IP. A correspondente resolução de nomes em IPs deverá igualmente ser assegurada pelo serviço de DNS.
 - Após as alterações efetuadas, ativar o serviço HAProxy através do seguinte comando:
`sudo service haproxy start.`
 - As diretivas de configuração do serviço HAProxy estão disponíveis em <http://cbonte.github.io/haproxy-dconv/configuration-1.6.html>. Aconselha-se uma leitura a este documento com vista a parametrizar convenientemente o serviço de acordo com o serviço balanceado e o cenário de rede em causa.
4. Analisar o funcionamento do balanceador e confirmar que os pedidos HTTP efetuados ao endereço IP de serviço são, de facto, distribuídos pelos vários servidores de “backend”. Para tal, efetuar vários pedidos ao servidor de “frontend” do HAProxy a partir do sistema hospedeiro ou de outro computador da rede. No cenário em configuração os servidores Web não partilham a área de disco onde estão guardadas as páginas relativas ao site web que é acedido. Nesse caso, poderá alterar o “index.html” de cada um dos servidores, permitindo-lhe verificar qual dos servidores é que responde ao pedido. Num cenário real os dados são partilhados pelos vários servidores Apache2 através de uma unidade de armazenamento comum.
5. Aceder às estatísticas do HAProxy através da interface web disponibilizada para o efeito. Para tal, deverá proceder às seguintes alterações:
- Adicionar um novo “listener” no ficheiro de configuração do HAProxy, dedicado exclusivamente ao serviço de estatísticas. O porto utilizado no exemplo seguinte é o 1936.

```
listen stats :1936  
    mode http  
    stats enable  
    stats hide-version  
    stats realm Haproxy\ Statistics  
    stats uri /status  
    stats auth haproxy:haproxy
```

Para a autenticação deverão usar-se as credenciais definidas na opção “stats auth”. A opção “stats uri” define o URL que deverá ser utilizado para aceder à consola de estatísticas do HAProxy. No exemplo apresentado deverá usar-se o seguinte URL: `http://localhost:1936/status`.

- Reiniciar o serviço, através do comando `"sudo service haproxy restart"`.
- Tomando como exemplo a configuração anterior, é possível aceder à interface web de estatísticas através do endereço `http://192.168.1.5:1936/status`. Através desta interface gráfica é possível verificar o funcionamento do servidor de "frontend" e dos vários servidores de "backend" que respondem aos pedidos HTTP.

3. Exercícios complementares

Apresentam-se de seguida três desafios relacionados com o cenário prático configurado anteriormente:

1. Pretende-se catalogar no serviço `rsyslog` as mensagens de log geradas pelo HAProxy com a etiqueta `"local1"`. Para tal deverão realizar-se as seguintes alterações:
 - a) alterar o parâmetro `"log"` no contexto `"global"` do HAProxy para o valor `"local1"`. Essa linha ficará: `"log 127.0.0.1 local1"`.
 - b) Adicionar a seguinte entrada ao ficheiro `/etc/rsyslog.d/haproxy.conf`:
`"local1.* /var/log/haproxy.log"`
 - c) Arrancar com o serviço `rsyslog`, utilizando para tal o comando `"sudo service rsyslog restart"`.
2. Uma solução de balanceamento de carga através do HAProxy inclui um ponto crítico de falha na rede: o servidor onde é feito o balanceamento de carga e onde está instalado o HAProxy. Para ultrapassar este problema e mitigar assim eventuais falhas que possam ocorrer no serviço de balanceamento de carga, pode recorrer-se à integração na rede, de um novo balanceador de carga, utilizado para backup. No caso de falha do servidor principal de HAProxy, o serviço de balanceamento de carga passa a ser assegurado pelo servidor de backup. Para tal, pretende-se utilizar uma estratégia de failover entre os dois servidores de HAProxy, utilizando um cluster de alta disponibilidade com o Heartbeat.
3. Na categoria de balanceadores de carga opensource para servidores aplicacionais, além do HAProxy podemos encontrar outras aplicações igualmente interessantes, como é o caso do Varnish (<https://www.varnish-cache.org>). Como trabalho de consolidação desta matéria propõe-se que os alunos efetuem um estudo comparativo entre as duas aplicações:

HAProxy e Varnish. Nesse sentido, pretende-se que configurem o cenário apresentado nesta ficha de trabalho com o Varnish para balancear pedidos HTTP.

4. Documentos de apoio

- Documentação oficial do HAProxy:
<http://www.haproxy.org/#docs>
- Documentação oficial do HAProxy disponibilizada no Github
<http://cbonte.github.io/haproxy-dconv/>
- Documentação oficial do Varnish
<https://www.varnish-cache.org/>