

# HAProxy: Reverse Proxy dan Load Balancer bagi Container Docker

Husni <husni@if.trunojoyo.ac.id>

Docker semakin banyak digunakan oleh para pengelola sistem hosting. setiap aplikasi dan domain dapat diletakkan di dalam suatu container terisolasi. Imagenya dapat dibawa dan dijalankan (menjadi container) dimana pun (selama ada Docker server) tanpa khawatir adanya ketidak-cocokan library (pada server Linux) dari aplikasi yang akan dijalankan. semua kebutuhan aplikasi telah dimasukkan ke dalam Image. Pengelola hosting cukup menjalankan Image tersebut dan mengatur akses Internet ke containernya. Singkatnya, administrator tinggal export dan import container Docker dan SELESAI. Tidak perlu lagi konfigurasi web server, pluginnya dan database. semuanya setelah setelah aplikasi dinyatakan siap PUBLISH.

ada satu hal penting yang harus dilakukan admnistrators. Kita tidak mungkin (tidak dapat) menjalankan banyak container dan memetakannya pada port (misal Web server) ke port 80 bersama-sama. Kita akan mengatur agar container mendengar koneksi pada port acak (random) atau port tertentu di atas 1024, misalnya: 4553, 4566, 4333 dan seterusnya. Bukankah pengunjung (visitor) web datang menuju port 80? Tidak selain itu kecuali disebutkan secara eksplisit. SOLUSI-nya adalah dengan menempatkan suatu software yang mendengar permintaan (request) pada port 80 dan meneruskan request tersebut ke container Docker pada port yang tepat. Software jenis ini dikategorikan sebagai REVERSE PROXY.

Ada beberapa software yang mampu melakukan ini, diantaranya Apache dan NginX. Namun dua software ini, fitrah-nya adalah Web server, meskipun dapat berfungsi reverse proxy yang baik. Tutorial ini akan menjelaskan cara memanfaatkan HAProxy sebagai Reverse Proxy. Keunggulan HAProxy dibandingkan Apache, Squid dan NginX adalah kehadiran fitur load balancing (awalnya memang dibuat untuk tujuan ini) dan kemudahan konfigurasi (relatif). Sangat mungkin, satu atau berapa aplikasi dijalankan dalam beberapa container pada port berbeda atau host berbeda. Oh ya, HAProxy tidak dibuat khusus untuk kepentingan Docker. HAProxy dapat digunakan sebagai reverse proxy dan load balancer bagi web server, database server, dan layanan lain yang berjalan tanpa virtual mesin atau virtualisasi.

## A. Instalasi dan Konfigurasi HAProxy

### 1. Instalasi HAProxy (Ubuntu)

```
sudo apt-get install haproxy
```

### 2. Edit file konfigurasinya. Secara default hanya ada satu file, yaitu /etc/haproxy/haproxy.cfg. Berikut ini adalah contoh isinya:

```
#/etc/haproxy/haproxy.cfg
global
    daemon
    maxconn 4096
```

```

defaults
    mode http
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms

frontend http-in
    bind *:80
    acl is_site1 hdr_end(host) -i site1.trunojoyo.ac.id
    acl is_site2 hdr_end(host) -i site1.trunojoyo.ac.id

    use_backend site1 if is_site1
    use_backend site2 if is_site2

    default_backend default_servers

backend site1
    balance roundrobin
    option httpclose
    option forwardfor
    server s2 10.1.2.46:12001 cookie maxconn 32
    server s2 10.1.2.46:12002 cookie maxconn 32 backup

backend site2
    balance roundrobin
    option httpclose
    option forwardfor
    server s1 10.1.2.46:5758 maxconn 32

backend default_servers
    balance roundrobin
    option httpclose
    option forwardfor
    server s2 10.1.2.46:8080 maxconn 32 check
    server s2 10.1.2.46:9090 maxconn 32 check

#listen admin 0.0.0.0:80
#http://10.1.2.46/haproxy?stats
listen admin
    bind 0.0.0.0:8080
    mode http
    stats enable
    stats uri /haproxy?stats
    stats realm Strictly\ Private
    stats auth A_Username:YourPassword
    stats auth Another_User:passwd

```

### 3. Aktifasi HAProxy

Kita perlu mengaktifkan HAProxy agar dapat dijalankan oleh skrip init (saat sistem operasi dimulai). Ini dilakukan dengan memberikan nilai 1 untuk variabel `ENABLED` di dalam file `/etc/default/haproxy`.

Buka text editor, tuliskan baris ini:

```
ENABLED=1
```

dan simpan sebagai `/etc/default/haproxy`.

Cara paling cepat adalah menggunakan perintah echo:

```
echo "ENABLED=1" > /etc/default/haproxy
```

Apakah perubahan ini sudah berlaku? dapat dicoba dengan mengeksekusi skrip ini HAProxy tanpa parameter sama sekali.

```
service haproxy
```

harusnya diperoleh hasil berikut:

```
Usage: /etc/init.d/haproxy {start|stop|reload|restart|status}
```

#### 4. Jalankan HAProxy

```
sudo /usr/sbin/haproxy -f /etc/haproxy.cfg -D -p /var/run/haproxy.pid
```

atau gunakan cara standard Ubuntu:

```
service haproxy start
```

Setelah 4 langkah di atas, HAProxy sudah dapat sebagai *reverse proxy* sekaligus *load balancer*.

## B. Membuat Image (Docker) HAProxy

Bagaimana jika HAProxy tidak diinstall di mesin Host tetapi akan dijadikan Image Docker?

Berikut ini adalah langkah-langkah membuat Image Docker untuk HAProxy:

1. Buat direktori khusus untuk keperluan pembuatan image, misalnya ~/dockker/haproxy.
2. Masuk ke dalam direktori tersebut (dengan cd).
3. Buat atau *copy*kan file haproxy.cfg (ke dalam direktori kerja). File haproxy.cfg di atas dapat digunakan sebagai contoh.
4. Buat Dockerfile, misalnya seperti berikut ini:

```
# Haproxy Dockerfile

# Pull base image.
FROM ubuntu

MAINTAINER Husnni <husni@if.trunojoyo.ac.id>

# Install Haproxy.
RUN \
  add-apt-repository ppa:vbernat/haproxy-1.5 && \
  apt-get update && \
  apt-get install -y haproxy && \
  sed -i 's/^ENABLED=.*\/ENABLED=1\/' /etc/default/haproxy && \
  rm -rf /var/lib/apt/lists/*
```

```
# Add files. Konfigurasi default
ADD haproxy.cfg /etc/haproxy/haproxy.cfg

# Define working directory.
WORKDIR /etc/haproxy

# Define default command.
CMD ["haproxy -f /etc/haproxy/haproxy.cfg"]
#CMD ["haproxy", "-f", "/etc/haproxy/haproxy.cfg"]

# Expose ports.
EXPOSE 80
#EXPOSE 443
#EXPOSE 3306
```

5. Buat image (misalnya bernama "img-haproxy-2015") dengan perintah build (sekali lagi, pastikan file Dockerfile dan haproxy.cfg ada di dalam direktori kerja):

```
sudo docker build -t img-haproxy-2015 .
```

6. Jalankan image (menjadi container)

```
sudo docker run -d -v ~/docker/haproxy:/etc/haproxy \
--name co-haproxy img-haproxy-2015
```

7. Jika diperlukan perubahan konfigurasi, edit file haproxy.cfg atau file lain (di direktori ~/docker/haproxy, pada Host). Kemudian gunakan perintah `docker exec` untuk mereload konfigurasi yang baru.

### Catatan:

1. Algoritma *load balancing*. Direktif `balance` menentukan algoritma *load balancing* yang akan digunakan pada suatu backend. Pilihan yang tersedia adalah Round Robin (`roundrobin`), Static Round Robin (`static-rr`), Least Connections (`leastconn`), Source (`source`), URI (`uri`) dan URL parameter (`url_param`).

```
balance roundrobin
balance url_param userid
balance url_param session_id check_post 64
balance hdr(User-Agent)
balance hdr(host)
balance hdr(Host) use_domain_only
```

2. Cara pengujian sederhana. Buat file berikut:

```
<?php
    header('Content-Type: text/plain');
    echo "Server IP: " . $_SERVER['SERVER_ADDR'];
    echo "\nClient IP: " . $_SERVER['REMOTE_ADDR'];
    echo "\nX-Forwarded-for: " . $_SERVER['HTTP_X_FORWARDED_FOR'];
?>
```

dan simpan sebagai `index.php` di dalam direktori document root dari Web server (biasanya `/var/www/`). Jika ada 5 web server yang akan *diload-balancing*kan, maka letakkan pada

kelimanya. Jika digunakan bersama container Docker, maka petakan file `index.php` ini ke setiap container yang dijalankan (`docker run -v`).

Gunakan web browser atau curl, akses URL web yang ditentukan dalam HAProxy, misalnya `site1.trunojoyo.ac.id`. Coba akses URL yang sama berulang-ulang. Perhatikan perbedaan setiap halaman yang dihasilkan. Kemudian matikan salah satu Web server (atau container). Akses kembali URL terakhir (mungkin dua atau 3 kali). Bagaimana hasilnya?

## PRAKTEK LANGSUNG (DEMO)

Universitas Trunojoyo Madura mempunyai domain Internet `trunojoyo.ac.id`. Sistem hosting berbasis docker akan menyediakan 2 aplikasi/halaman web, masing-masing diberi subdomain `kamal.trunojoyo.ac.id` yang terdiri dari 3 container (atau server web) dan `telang.trunojoyo.ac.id` yang mempunyai 2 server. Server *domain name service* (DNS) dari `trunojoyo.ac.id` telah dikonfigurasi agar permintaan koneksi ke dua subdomain tersebut di arahkan ke mesin ber-IP Address 10.1.2.46. Pada mesin 10.1.2.46 tersebut terdapat HAProxy dan Docker. Docker bertanggungjawab menjalankan 5 web server dalam satu mesin, sedangkan HAProxy bekerja sebagai REVERSE Proxy dan Load Balancer.

Berikut ini adalah langkah-langkah membangun sistem hosting yang demikian:

1. Pastikan Docker dan HAProxy telah terinstall di mesin tersebut. Jika belum gunakan perintah berikut:

```
sudo apt-get update && sudo apt-get install docker.io haproxy
```

2. Pada home directory (~), misalnya `/home/admin` (sesuai user name), buat direktori `docker/webserver` yang akan menyimpan konfigurasi image `img-webserver`, termasuk konfigurasi virtual host apache dan file `index.php` default dari image.
3. Buat Dockerfile, `webkita.conf` dan `index.php` di dalam direktori `~/docker/webserver`, seperti berikut:

```
-----  
#Docker file untuk membuat web server apache + php  
# Build the image of ubuntu 14.04 LTS  
FROM ubuntu:14.04  
  
# Run apt-get update  
RUN apt-get -y update  
  
# Install Apache, PHP and stuff  
RUN apt-get -y install apache2  
RUN apt-get -y install php5 libapache2-mod-php5 php5-mcrypt  
RUN apt-get -y install libapache2-mod-auth-mysql php5-mysql  
  
ADD webkita.conf /etc/apache2/sites-available/  
RUN a2dissite 000-default  
RUN a2ensite webkita  
  
# Expose port 80 to the host machine  
EXPOSE 80
```

```
ENTRYPOINT ["/usr/sbin/apache2ctl"]
CMD ["-D", "FOREGROUND"]
```

```
-----
#webkita.conf
<VirtualHost *:80>
    DocumentRoot /var/www/

    LogLevel warn
    ErrorLog /dev/stdout
    CustomLog /dev/stdout combined
</VirtualHost>

-----
//index.php
<?php
    print("<h1>Situs ini sedang MAINTENANCE...</h1>");
    echo "<h3 center>admin.min at puskom.kom</h3>";
?>
```

4. Pastikan berada di dalam direktori `~/docker/webserver`. Jalankan docker untuk membuat image `img-webserver`:

```
docker build -t img-webserver .
```

5. Pada home directory (`~`), di bawah direktori `docker/` buat direktori `kamal1`, `kamal2`, `kamal3`, `telang1`, dan `telang2`. Lima direktori ini yang akan menyimpan file-file halaman atau aplikasi web dari lima web server yang akan dijalankan oleh Docker.
6. Di dalam semua direktori berawalan `kamal`, buat file seperti ini:

```
//index.php
<?php
    print("<h1>Selamat datang di KAMAL</h1>");
    echo "<h5>hosted by: kamal1 </h5>";
?>
```

Ganti teks “`kamal1`” sesuai dengan nama direktori dimana file tersebut ditempatkan (`kamal2` dan `kamal3`).

7. Lakukan hal yang sama untuk membuat file `index.php` di dalam direktori `telang1` dan `telang2`.
8. Jalankan image `img-webserver` (menjadi container) sebanyak lima kali, masing-masing dengan data volume dan port host berbeda:

```
docker run -d -v ~/docker/kamal1:/var/www -p 8001:80 \
--name co-kamal1 img-webserver
docker run -d -v ~/docker/kamal2:/var/www -p 8002:80 \
--name co-kamal2 img-webserver
docker run -d -v ~/docker/kamal3:/var/www -p 8003:80 \
--name co-kamal3 img-webserver
docker run -d -v ~/docker/telang1:/var/www -p 9001:80 \
--name co-telang1 img-webserver
docker run -d -v ~/docker/telang2:/var/www -p 9002:80 \
--name co-telang2 img-webserver
```

9. Konfigurasi frontend dari HAProxy agar dapat bekerja sebagai Reverse proxy:

```
frontend http-in
  bind *:80
  acl is_kamal hdr_end(host) -i kamal.trunojoyo.ac.id
  acl is_telang hdr_end(host) -i telang.trunojoyo.ac.id

  use_backend kamal if is_kamal
  use_backend telang if is_telang

  default_backend default_servers
```

10. Konfigurasi backend dari HAProxy untuk mengarahkan request ke server web yang bersesuaian termasuk penanganan *load balancing*:

```
backend kamal
  balance roundrobin
  option httpclose
  option forwardfor
  server kamal1 10.1.2.46:8001 check maxconn 32
  server kamal2 10.1.2.46:8002 check maxconn 32
  server kamal3 10.1.2.46:8003 maxconn 32

backend telang
  balance roundrobin
  option httpclose
  option forwardfor
  server telang1 10.1.2.46:9001 check maxconn 32
  server telang2 10.1.2.46:9002 check maxconn 32

backend default_servers
  option httpclose
  option forwardfor
  server s2 10.1.2.46:80 maxconn 32 check

#http://10.1.2.46/haproxy?stats
listen admin
  mode http
  stats enable
  stats uri /haproxy?stats
  stats realm Strictly\ Private
  stats auth admin:admin123
```

11. Jalankan atau restart HAProxy

```
sudo service haproxy restart
```

12. Buka web browser dan arahkan ke `telang.trunojoyo.ac.id` atau `kamal.trunojoyo.ac.id`. Coba *refresh* berulang-ulang dan perhatikan halaman web yang diperoleh.

13. Pada web browser, akses URL `http://10.1.2.46/haproxy?stats` untuk melihat statistik dari *request* yang telah ditangani oleh HAProxy.

Semoga tutorial ini berguna...selamat belajar ☺