globalscape™

How to Configure an HAProxy Load Balancer with EFT HA Cluster

Introduction

HAProxy is an open source, Linux-based load balancer that can be used as a load balancer for traffic inbound to an EFT HA cluster.

From http://www.haproxy.org/: *HAProxy is a free, very fast and reliable solution offering high availability, load balancing, and proxying for TCP and HTTP-based applications. It is particularly suited for very high traffic web sites and powers quite a number of the world's most visited ones. Over the years it has become the de-facto standard opensource load balancer, is now shipped with most mainstream Linux distributions, and is often deployed by default in cloud platforms*

This article provides the steps and some configuration samples for a HAProxy running on CentOS/REHL that can be used to load balance N number of nodes of an EFT in High Availability mode for inbound connections.

Disclaimer:

This article it is intended for technical audience and it is provided "As Is" without any guaranty or support; it is intended for demonstration/educational purposes only. Globalscape recommends using a hardware-based load balancer like Big IP F5 or similar for production environments. Additionally, HAProxy it is open source and should be considered at customer's discretion. Please consult with your Network Administrator or Globalscape Tech Support for more information.

Prerequisites

HAProxy requires all nodes clocks to be synchronized so you will need to enable `ntpd` on each node:

```
# chkconfig ntpd on
# ntpdate pool.ntp.org
# /etc/init.d/ntpd start
```

Step1. Setup an epel Repository

Install an `epel` repository on your CentOS/RHEL Linux OS by using the following:

```
CentOS/RHEL 5 , 32 bit:
# rpm -Uvh http://dl.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm

CentOS/RHEL 5 , 64 bit:
# rpm -Uvh http://dl.fedoraproject.org/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm

CentOS/RHEL 6 , 32 bit:
# rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm

CentOS/RHEL 6 , 64 bit: # rpm -Uvh
http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

Step2. Install HAProxy using Yum

Install HAproxy package using following command

```
# yum install haproxy
```

Step 3. Configure HAProxy

Update your HAProxy configuration file: **/etc/haproxy/haproxy.cfg**

Below is a sample configuration script that can be used.

This configuration shows an EFT cluster with:

- 3 nodes for FTP (21), FTPS Explicit (21), FTPS Implicit (990), HTTP (80), HTTPS (443), and SFTP (23) protocols
- Uses static IPs on each node
- Uses a round-robin balance algorithm.
- Has the same weight for all nodes
- Has a max number of concurrent connections of 4096 for each nodes

```
# Example configuration for a possible EFT Server Cluster.  See the
# full configuration options online.
#
#   http://haproxy.1wt.eu/download/1.4/doc/configuration.txt
#
#---------------------------------------------------------------------
#---------------------------------------------------------------------
# Global settings
#---------------------------------------------------------------------
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     45000
    user        haproxy
    group       haproxy
    daemon
    nbproc      1

    # 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
    mode                    tcp
    log                     global
    option                  tcplog
    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                 4096



### Initial connection and control traffic for FTP
## See http://www.taiter.com/techlog/2012/09/ftp-load-balanced-through-haproxy.html
frontend  ftp-Control
   bind *:21
   default_backend ftp_server_pool


### Initial connection and control traffic for FTPS Implicit
frontend ftps-Control
   bind *:990
   default_backend ftps_server_pool


### Each of these frontends represent a server and its corresponding PASV ports
frontend ftp-server1
   bind *:28000-30000
   default_backend eft_server1


### Each of these frontends represent a server and its corresponding PASV ports
frontend ftp-server2
   bind *:30001-32000
   default_backend eft_server2


### Each of these frontends represent a server and its corresponding PASV ports
frontend ftp-server3
   bind *:32001-34000
   default_backend eft_server3



### Global backend for the FTP control traffic to find a server
```

```
backend ftp_server_pool
   server eft-server1 192.168.105.125 check port 21 inter 10s rise 1 fall
2
   server eft-server2 192.168.105.126 check port 21 inter 10s rise 1 fall
2
   server eft-server3 192.168.105.127 check port 21 inter 10s rise 1 fall
2
### Global backend for the FTPS (Implicit) control traffic to find a
server
backend ftps_server_pool
   server eft-server1 192.168.105.125 check port 990 inter 10s rise 1 fall
2
   server eft-server2 192.168.105.126 check port 990 inter 10s rise 1 fall
2
   server eft-server3 192.168.105.127 check port 990 inter 10s rise 1 fall
2



### Backends for each of our EFT servers
backend eft_server1
   server eft-server1 192.168.105.125

backend eft_server2
   server eft-server2 192.168.105.126

backend eft_server3
   server eft-server3 192.168.105.127

### Configuration for HTTP sites

frontend http
        bind *:80
        default_backend eft-http

frontend https
        bind *:443
```

```
        default_backend eft-https


backend eft-http
        mode http
        balance roundrobin
        stick on src table eft-https
        appsession websessionid len 64 timeout 30m
        server server1 192.168.105.125:80 weight 1 maxconn 512 check
cookie server1
        server server2 192.168.105.126:80 weight 1 maxconn 512 check
cookie server2
        server server3 192.168.105.127:80 weight 1 maxconn 512 check
cookie server3


backend eft-https
        mode tcp
        balance roundrobin
        stick-table type ip size 200k expire 30m
        stick on src
        server server1 192.168.105.125:443 weight 1 maxconn 512 check
        server server2 192.168.105.126:443 weight 1 maxconn 512 check
        server server3 192.168.105.127:443 weight 1 maxconn 512 check



### Configuration for SFTP
## see http://jpmorris-iso.blogspot.com/2013/01/load-balancing-openssh-
sftp-with-haproxy.html
listen sftp_in *:23
        mode tcp
        option tcplog
        balance roundrobin
        server server1 192.168.105.125:23
        server server2 192.168.105.126:23


        server server3 192.168.105.127:23
```

Step 5. Start HAProxy service

Start the HAProxy service using following command.

```
# service haproxy start
 # chkconfig haproxy on
```

You might also consider configure HAProxy to automatically start on system boot. Please consult your OS documentation on how to start services automatically.

Note:

If you are using VM images for your HAProxy service and you are cloning images, please review this article:

http://www.envision-systems.com.au/blog/2012/09/21/fix-eth0-network-interface-when-cloning-redhat-centos-or-scientific-virtual-machines-using-oracle-virtualbox-or-vmware/

Disable Firewall for Troubleshooting Only

You will need to configure your Linux Firewall properly to allow the ports needed on the load balancer, or if you are currently using a dedicated firewall and you might need to disable your Linux Firewall for Troubleshooting, here are some useful commands:

```
# service iptables save
# service iptables stop
# chkconfig iptables off
```

Note: It is NOT recommended that you disable your firewall in your production environment; you should consult with your Network Administrator before make any changes.

References
http://haproxy.1wt.eu/download/1.4/doc/configuration.txt