**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_HAproxy\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**HAProxy is a very fast and reliable solution for high**

**availability, load balancing, It supports TCP and HTTP-based applications.**

**Nowadays maximizing websites up-time is very crucial for heavy traffic websites.**

**This is not possible with a single server setup. Then we need some high availability**

**environment which can easily manage with a single server failure.**

**Load balancing is a common solution for distributing web applications horizontally**

**across multiple hosts while providing the users with a single point of access to the service.**

**HAProxy is one of the most popular open source load balancing software,**

**which also offers high availability and proxy functionality**.

**install the 2 web servers and 1 haproxy server on ubuntu...**

**{note:do not install apache on haproxy server}**

**There r 2 web servers running with Apache2 and listening on port 80 and one HAProxy server.**

**Server 1: web1.sam.com 192.168.72.81**

**Server 2: web2.sam.com 192.168.72.82**

**server 3: haproxy 192.168.72.85**

**Hostname of both servers**

web1.sam.com

web2.sam.com

#apt install apache2

#ufw allow 80/tcp

#cd /var/www/html/

#mkdir sam.com

#cd sam.com

#vi index.html

#cd ..

#chown -Rf www-data sam.com/

#chmod -Rf 775 sam.com/

#vi /etc/apache2/sites-available/sam.com.conf

<VirtualHost 192.168.72.81:80>

ServerAdmin sam.com

ServerName web1.sam.com

DirectoryIndex index.html

DocumentRoot /var/www/html/sam.com

</VirtualHost>

#vi /etc/apache2/apache2.conf

<Directory /var/www/html/sam.com>

Options Indexes MultiViews FollowSymLinks

DirectoryIndex index.php index.html

AllowOverride all

Allow From all

</Directory>

#a2ensite sam.com

#apache2ctl configtest

#systemctl restart apache2

**add the ip and domain in windows /etc file**

**after configuration of web servers we have to configure HAproxy server**

**install HAProxy using following commands**.

#apt-get install haproxy

**Now edit haproxy default configuration file**

#vi /etc/haproxy/haproxy.cfg

**add the following lines at the bottm....**

global

log /dev/log local0

log /dev/log local1 notice

chroot /var/lib/haproxy

stats socket /run/haproxy/admin.sock mode 660 level admin

stats timeout 30s

user haproxy

group haproxy

daemon

# Default SSL material locations

ca-base /etc/ssl/certs

crt-base /etc/ssl/private

# Default ciphers to use on SSL-enabled listening sockets.

# For more information, see ciphers(1SSL). This list is from:

# https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/

# An alternative list with additional directives can be obtained from

# https://mozilla.github.io/server-side-tls/ssl-config-generator/?server=haproxy

ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS

ssl-default-bind-options no-sslv3

defaults

log global

mode http

option httplog

option dontlognull

timeout connect 5000

timeout client 50000

timeout server 50000

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

frontend haproxynode

bind \*:80

mode http

default\_backend backendnodes

backend backendnodes

balance roundrobin

option forwardfor

http-request set-header X-Forwarded-Port %[dst\_port]

http-request add-header X-Forwarded-Proto https if { ssl\_fc }

option httpchk HEAD / HTTP/1.1\r\nHost:localhost

server web1.sam.com 192.168.72.81:8080 check

server web2.sam.com 192.168.72.82:8080 check

listen stats

bind :8080

stats enable

stats uri /haproxy?stats

stats hide-version

stats auth abhi:admin

Now verify configuration file before restarting service using the following command.

#haproxy -c -f /etc/haproxy/haproxy.cfg

#service haproxy restart

**Now access port 80 on IP 192.168.72.85 (as configured above) in the web browser and hit refresh.**

**You will see that HAProxy is**

**sending requests to backend server one by one (as per round robin algorithm).**

**With each refresh you can that HAProxy is sending request one by one to a backend server.**

**for more details u can ckeck following links(websites)....**

https://tecadmin.net/how-to-setup-haproxy-load-balancing-on-ubuntu-linuxmint/

https://upcloud.com/community/tutorials/haproxy-load-balancer-ubuntu/

**best web site for all versions.....**

http://cbonte.github.io/haproxy-dconv/