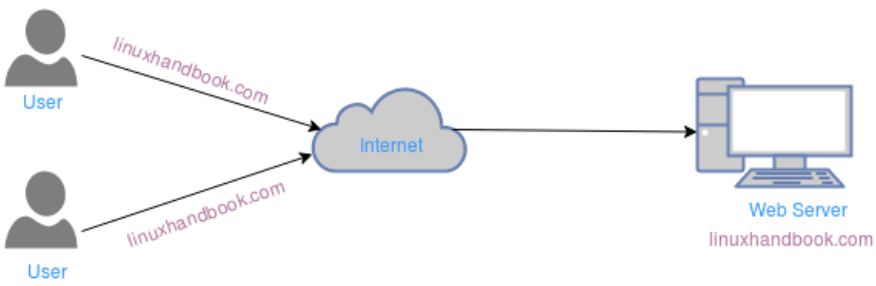
Last Updated on July 11, 2019 By Rishi Raj Gautam 7 Comments

Having a proper set up of load balancer allows your web server to handle high traffic smoothly instead of crashing down. In this tutorial, we'll see how to setup a load balancer with high availability.

What is load balancing?

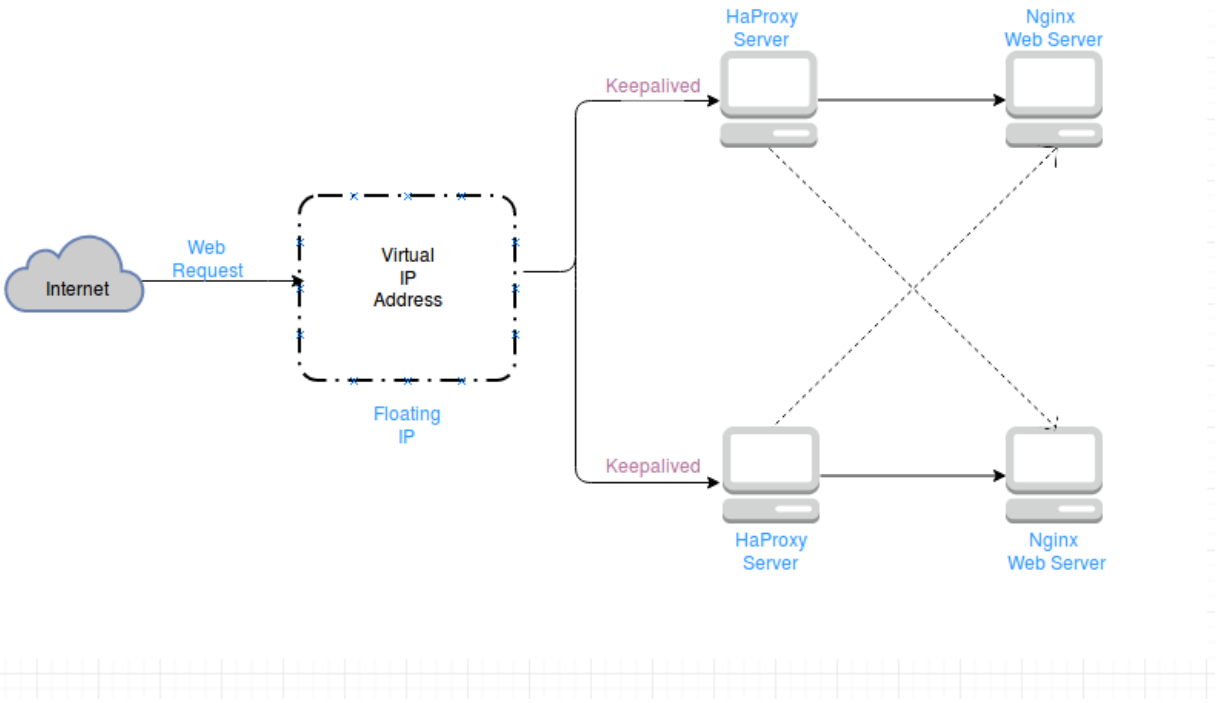
Load balancing is the process of distributing workloads to multiple servers. It is like distributing workloads between day shift and night shift workers in a company. Load balancing improves the server's reliability as it overcomes single point failure. An example of How a server without load balancing looks like is shown below.



A single server handling traffic

In this example, if the web server goes down, the user's web request cannot be accessed in real time. Also if numbers of users request the same web page simultaneously, then serving the user's web request by a single web server can be a slow process. Hence load balancers are used to enhance the server's performance, provide backup and prevent failures.

In this tutorial, we are going to set up a load balancer for web server using Nginx, HAProxy and Keepalived. An example of how servers with load balancers look like is shown below.



Using load balancing to effectively handle high traffic

So, what are Nginx, Haproxy and Keepalived?

Nginx

Nginx, pronounced as Engine-x is an open-source Web server. More than just a Web server, it can operate as a reverse proxy server, mail proxy server, load balancer, lightweight file server and HTTP cache. Nginx has been used in many popular sites like BitBucket, WordPress, Pinterest, Quora and GoDaddy.

HAProxy

HAProxy stands for High Availability Proxy. It is an open source load balancer that provides load balancing, high availability and proxy solutions for TCP and HTTP based applications. It is best suited for distributing the workload across multiple servers for performance improvement and reliability of servers.

The function of Haproxy is to forwards the web request from end-user to one of the available web servers. It can use various load balancing algorithms like Round Robin, Least Connections etc.

Keepalived

What if HAProxy load balancer goes down?

Keepalived is an open-source program that supports both load balancing and high availability. It is

basically a routing software and provides two types of load balancing:

- Layer 4 (transport layer)
- Layer 7 (application layer)

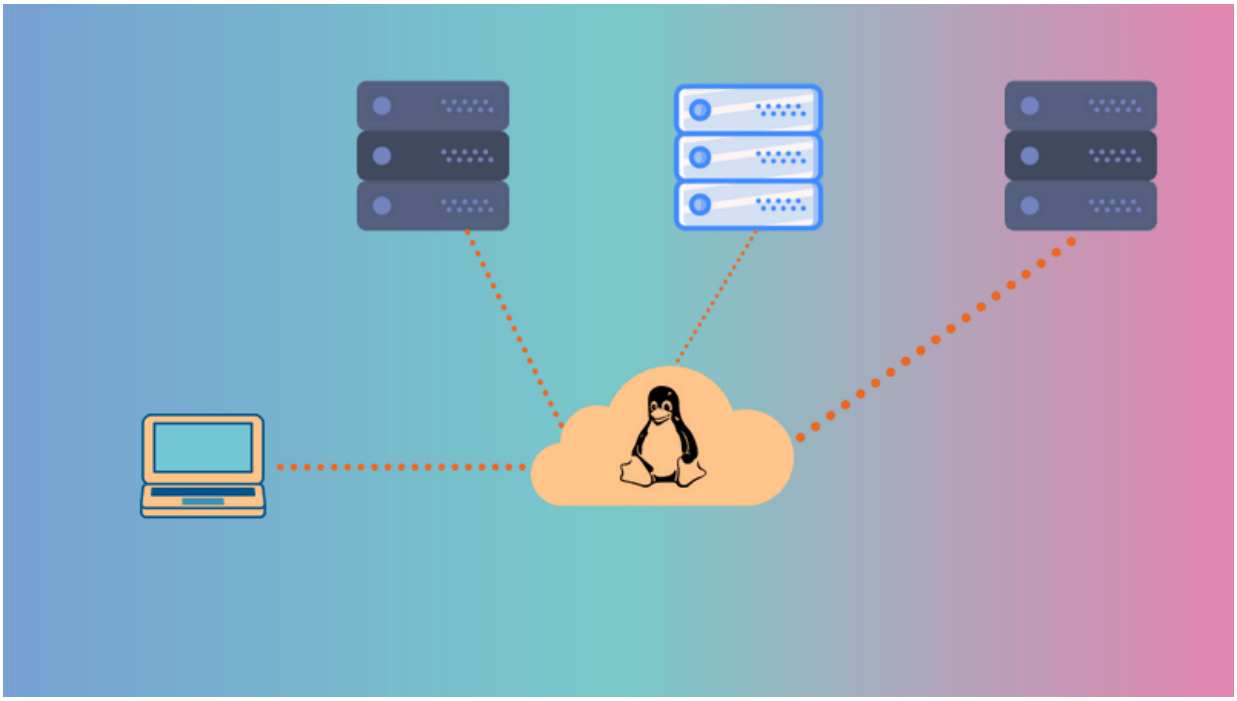
Keepalived can perform the following functions:

- Health checking (whether the servers are up or not)
- Implements VRRP (virtual redundancy routing protocol) to handle load-balance failover

Keepalived uses VIP (Virtual IP Address) as a floating IP that floats between a Master load balancer and Backup load balancer and is used to switch between them. If Master load balancer goes down, then backup load balancer is used to forward web request.

Let's move towards simulation of how high availability and load-balancing is maintained for web servers.

Setting up a load balancer in Linux with Nginx, HAProxy and Keepalived



This is a test lab experiment meaning it's just a test setup to get you started. You may have to do some tweaking if you are implementing it on real servers. Use this tutorial as a learning material instead of blindly following it for your own setup.

I have used CentOS Linux distribution in this tutorial. You can use other Linux distributions but I cannot guarantee if all the commands (especially the installation ones) will work in other distributions.

Requirements for load balancer setup

4 CentOS installed systems (minimal installation is enough for this tutorial)

- 2 CentOS to be set up with nginx
- 2 CentOS to be set up with HAProxy and Keepalived

In this tutorial, we have worked on the following IP addresses as an example. These can be changed as per your system. Don't think these are the static IPs.

Web servers:

- 10.13.211.169
- 10.13.211.158

LoadBalancer:

- 10.13.211.194
- 10.13.211.120

Virtual IP:

- 10.13.211.10

You can easily get IP address in Linux command line.

Step 1: Setup the web servers with Nginx

In this part, we'll use two CentOS systems as the web server. We need to install Nginx on them first.

For that, add a repository containing nginx and then install it from there:

```
yum install epel-release
yum install nginx
```

After installing nginx, start the Nginx service:

```
systemctl start nginx
```

Make nginx service to be enabled even after every boot:

```
systemctl enable nginx
```

Check the status of nginx service:

```
systemctl status nginx
```

Allow the web traffics in nginx that is by default block by CentOS firewall.

```
firewall-cmd --zone=public --permanent --add-service=http
firewall-cmd --zone=public --permanent --add-service=https
firewall-cmd –reload
```

Repeat the above steps on the second CentOS web server as well.

Now pay attention to the next steps.

The web files for nginx is located in `/usr/share/nginx/html` Change the content of index.html file just to identify the webservers.

For the first web server:

```
echo “this is first webserver” > /usr/share/nginx/html/index.html
```

For the second web server:

```
echo “this is second webserver” > /usr/share/nginx/html/index.html
```

> NOTE: If you are on a virtual machine, it is better to install and configure Nginx on one system and then clone the system. Afterward, you can reconfigure on the second system. Saves time and errors.

Now confirm the web server status by going to the following URL in your browser: http://SERVER_DOMAIN_NAME or Local_IP_Address. Example here:

```
http://10.13.211.169
```

or in the terminal, curl Local_IP_Address. Example here:

```
curl 10.13.211.169
```

You will get the output like:

Step 2: Setup load balancers with HAProxy

On the other two systems, use the following commands to install HAProxy:

```
yum -y update
yum -y install haproxy
```

HAProxy configuration file is located at /etc/haproxy. Use the cd command to go to the directory and backup the file before edit.

```
cd /etc/haproxy/
mv haproxy.cfg haproxy.cfg_bac
```

Create a new haproxy.cfg file and open the file with any editor you like.

```
touch haproxy.cfg
vim haproxy.cfg
```

Now, paste the following lines into the file:

```
 global
   log /dev/log local0
   log /dev/log local1 notice
   chroot /var/lib/haproxy
   stats timeout 30s
   user haproxy
   group haproxy
   daemon
defaults
log global
mode http
option httplog
option dontlognull
timeout connect 5000
timeout client 50000
timeout server 50000

#frontend
#---------------------------------
frontend http_front
bind *:80
stats uri /haproxy?stats
default_backend http_back

#round robin balancing backend http
#---------------------------------
backend http_back
balance roundrobin
#balance leastconn
mode http
server webserver1 10.13.211.169:80 check    # ip_address_of_1st_centos_webserver
server webserver2 10.13.211.158:80 check    # ip_address_of_2nd_centos_webserver
```

Now, enable and start HAProxy service.

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

Check the status of HAProxy:

```
systemctl status haproxy
```

Go to url in your browser to confirm the service of haproxy: http://load balancer's IP Address/haproxy?stats. Example used here:

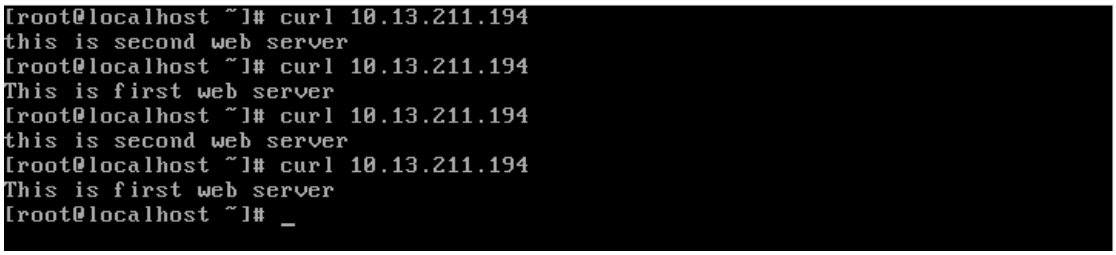
```
http://10.13.211.194/haproxy?stats
```

or in the terminal, use command $ curl LoadBalancer_IP_Address

```
curl 10.13.211.194
curl 10.13.211.194
```

curl two times and you will see different outputs for the curl command. It is because of the response is coming from different web servers (one at a time), for your request at the load balancer.

The Output would look like this:



Step 3: Set up high availability with Keepalived

Keepalived must be installed to both HAProxy load balancer CentOS systems (which we have just configured above). One acts a master (main load-balancer) and another acts as the backup load-balancer.

On both system, run the following command:

```
yum install -y keepalived
```

The configuration file of Keepalived is located at `/etc/keepalived/keepalived.conf`. Backup the original keepalived.conf file and use the following configuration at new keepalived .conf file.

```
mv /etc/keepalived/keepalived.conf /etc/keepalived/keepalived.conf_bac
touch /etc/keepalived/keepalived.conf
vim /etc/keepalived/keepalived.conf
```

Paste the following lines to the configuration file (don't forget to change the email addresses):

```
 global_defs {
notification_email {
linuxhandbook.com
linuxhandbook@gmail.com
}
notification_email_from thunderdfrost@gmail.com
smtp_server 10.13.211.1
smtp_connect_timeout 30
router_id LVS_DEVEL
}

vrrp_instance VI_1 {
state MASTER
interface eth0 #put your interface name here. [to see interface name: $ ip a ]
virtual_router_id 51
priority 101 # 101 for master. 100 for backup. [priority of master> priority of backup]
advert_int 1
authentication {
auth_type PASS
auth_pass 1111 #password
}
virtual_ipaddress {
10.13.211.10 # use the virtual ip address.
}
}
```

Note: Virtual IPs can be any live IP inside your network. Near about the range of Loadbalancer's IP Address. Here, the load balancer's IP are: 10.13.211.194 & 10.13.211.120, and VIP is 10.13.211.10 Edit the configuration file as per the system assumption. Take care on master and backup configuration. Save the file and start and enable the Keepalived process:

```
 systemctl start keepalived
systemctl enable keepalived
```

To view the status of Keepalived:

```
 systemctl status keepalived
```

Note: If you are on a virtual machine, it is better to install and configure Haproxy and Keepalived on one system and then clone the system. Afterward, you can reconfigure on

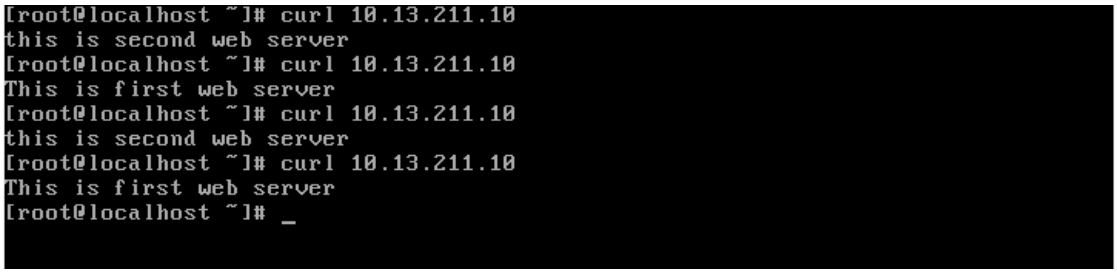
```
the second system. Saves time and errors.
```

Now to check the status of your high-availability load-balancer, go to terminal and hit:

```
$ while true; do ; curl 10.13.211.10 ; sleep 1; done;
```

Hit `ctrl+c` to stop the terminal run.

The output will look like this:



If you feel uncomfortable in installing and configuring the files, download the scripts form my GitHub repository and simply run them.

I hope this tutorial helped you to set up a load balancer in Linux with high availability. Of course, it was a simple setup but it definitely gives an idea about load balancing and handling high availability.

If you have questions or suggestions please leave a comment below.

Filed Under: Tutorial

Tagged With: CentOS, Haproxy, High Availability, Keepalived, Linux system Administration, Load-balancing, Nginx, System Administration, Virtual Machine, VM, Webserver

ABOUT RISHI RAJ GAUTAM

I am a Linux lover and an open-source activist. I believe that doing what we love is the source of inspiration of doing more. Apart from computer stuffs, I like to travel and feel like traveling is one of the ways to seek myself.

Search this website

46K Followers

Follow us on Facebook **44.3K** Followers

Follow us on Twitter **1.2K** Followers

Follow us on YouTube **430** Followers

DON’T MISS THIS LIMITED TIME OFFER