By hedieh
Published: 2010-02-15 17:47

Efficient High-Available LoadBalanced Cluster On CentOS 5.3 (Direct Routing Method)

This article explains how to set up an LVS cluster of load balanced virtual servers with Heartbeat and Ldirectord On CentOS 5.3.The load balancer sits between the user and two (or more) backend Apache/IIS web servers that hold the same content. Not only does the load balancer distribute the requests to the two backend Apache/IIS servers, it also checks the health of the backend servers. If one of them is down, all requests will automatically be redirected to the remaining backend server.

Introduction

An LVS cluster consists or one or more virtual services each of which may have zero or more real servers. The IP address of a virtual service is what end-users connect to and is typically advertised over DNS. When a connection is made to a virtual service, it is allocated a real server, and all packets for this connection are forwarded to this real server. Ldirectord is a daemon to monitor and administer real servers in a LVS cluster of load balanced virtual servers. Ldirectord typically used as a resource for Linux-HA. L***directord*** monitors the health of the real servers by periodically requesting a known URL and checking that the response contains an expected response. If a real server fails then the server is removed and will be reactivated once it comes back on line. If all the real servers are down then a fall-back server is inserted into the pool, which will made quiescent one of the real web servers comes back on line. Typically, the fall-back server is localhost. If an HTTP virtual service is being provided then it is useful to run an Apache HTTP server that returns a page indicating that the service is temporarily inaccessible.

Note: This tutorial is based on my personal experience and some other tutorials which is publicly available on Internet. I do not issue any guarantee that this will work for you!.

Preliminary Note

In this tutorial I will use the following 3 hosts:
Virtual IP address (end users connect to this) : *`10.10.10.53`*
Load Balancer: *`ld.example.com`*, IP address: *`10.10.10.52`*
Web Server 1: *`http1.example.com`*, IP address: *`192.168.200.102`*
Web Server 2: *`http2.example.com`*, IP address: *`192.168.200.103`*

Load Balancer Configuration

Install heartbeat,heartbeat-ldirector and ipvsadm packages on your Load Balancer system (ld.example.com).

```
yum install heartbeat heartbeat-ldirector ipvsadm -y

chkconfig ldirectord off

chkconfig heartbeat on

sed -i 's/net.ipv4.ip_forward = 1/net.ipv4.ip_forward = 0' /etc/sysctl.conf

sysctl -p
```

Load Balancer Secondary Ethernet Configuration

Configure secondary eth0 for LVS as its going to be exposed to outside world or your local gateway.

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE=eth0:0
BOOTPROTO=none
ONBOOT=yes
HWADDR=3a:5d:71:ad:67:47
NETMASK=255.255.255.0
IPADDR=10.10.10.52
GATEWAY=10.10.10.1
TYPE=Ethernet
```

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0:0
```

```
DEVICE=eth0:0
BOOTPROTO=none
ONBOOT=yes
HWADDR=3a:5d:71:ad:67:47
NETMASK=255.255.255.0
IPADDR=10.10.10.53
TYPE=Ethernet
```

```
service network restart
```

Configuring ldirectord

Configure ldirectord on your Load Balancer system.

```
vi /etc/ha.d/ldirectord.cf
```

```
checktimeout=30
checkinterval=2
autoreload=yes
```

```
logfile="/var/log/ldirectord.log"
quiescent=no
virtual=10.10.10.53:80
        fallback=127.0.0.1:80
        real=192.168.200.102:80 gate
        real=192.168.200.103:80 gate
        service=http
        request="/check.txt"
        httpmethod=GET
        receive="webserverisworking"
        persistent=100
        scheduler=lblc
        protocol=tcp
        checktype=negotiate
```

Important Note: after virtual=x.x.x.x:80 line , each line MUST start with TAB.Don't forget to press TAB key before each lines.

```
service ldirectord start
```

In the *virtual=* line we put our virtual IP address (*10.10.10.53* in this example), and in the *real=* lines we list the IP addresses of our Apache/IIS nodes (*192.168.200.102* and *192.168.200.103* in this example). In the *request=* line we list the name of a file on *http1* and *http2* that ldirectord will request repeatedly to see if *http1* and *http2* are still alive. That file (that we are going to create later on) must contain the string listed in the *receive=* line.
In the *scheduler=* line you can use one of the following method depending on your needs:rr - wrr - lc - wlc - lblc - lblcr - dh - sh - sed - nq

For more information about scheduler methods visit: ***http://linux.die.net/man/8/ipvsadm***

Configure heartbeat

Configure heartbeat on your Load Balancer system.

```
vi /etc/ha.d/ha.cf
```

```
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility     local0
keepalive 2
deadtime 10
bcast   eth0
mcast eth0 225.0.0.1 694 1 0
auto_failback on
respawn hacluster /usr/lib/heartbeat/ipfail
node ld.example.com
```

Important: As nodenames we must use the output of:

```
uname -n
```

```
vi /etc/ha.d/haresources
```

```
ld.example.com ldirectord::ldirectord.cf LVSSyncDaemonSwap::master IPaddr2::10.10.10.53/24/eth0/10.10.10.255
```

The first word in the first line above is the output of

```
uname -n
```

```
vi /etc/ha.d/authkeys
```

```
auth 3
3 md5 somerandomstring
```

```
chmod 600 /etc/ha.d/authkeys
```

Testing

Let's check if load balancer work as expected:

```
ip addr sh eth0
```

The load balancer should list the virtual IP address (*10.10.10.53*):

```
2: eth0: <broadcast,multicast,up,lower_up> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:c8:6f:2f brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.52/24 brd 10.10.10.255 scope global eth0
    inet 10.10.10.53/24 brd 10.10.10.255 scope global secondary eth0:0
```

If your tests went fine, you can now go on and configure the two Apache/IIS nodes.

Cluster Nodes Configurations (Apache Real Web Servers Configuration)

On both web servers http1 and http2, apache should be running having a common serving file (for purpose of get checked by ldirectord).

```
yum install httpd -y

echo "webserverisworking" > /var/www/html/check.txt

service httpd start

chkconfig httpd on
```

Now, Create a loopback interface on each web server, so it doesn(TM)t communicate with your network gateway/router directly.

```
vi /etc/sysconfig/network-scripts/ifcfg-lo:0
```

It must look like this:

```
DEVICE=lo:0
```

```
IPADDR=10.10.10.53
NETMASK=255.255.255.255
ONBOOT=yes
NAME=loopback
```

```
vi /etc/sysctl.conf
```

It must look like this:

```
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.eth0.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.ipv4.conf.eth0.arp_announce = 2
```

```
sysctl -p

ifup lo:0
```

Windows XP/2003/2008 users!: if you are using IIS6/7 as a web server then you should follow the next steps otherwise just skip.

Cluster Nodes Configurations (IIS6/7 Real Web Server Configuration)

1. Create text file by using Notepad and name it "check.txt"

2. Fill this file with "webserverisworking" string.
3. Move file to "C:inetpubwwwroot" or anywhere your web files are.

If you are using Windows XP/2003 IIS web server then you should do these steps:

1. Install "Microsoft Loopback Adapter" by using "Add Hardware" icon in Control Panel.
2. Set IP to 10.10.10.53
3. Set Subnet Mask to 255.255.255.0
4. Don't Set Gateway or DNS
5. Done!

If you are using Windows 2008 IIS web servers then you should do these steps:

1. Install "Microsoft Loopback Adapter" by using "Add Hardware" icon in Control Panel.
2. Set IP to 10.10.10.53
3. Set Subnet Mask to 255.255.255.0
4. Don't Set Gateway or DNS
5. Then you need to use the following command line magic :

```
netsh interface ipv4 set interface "net" weakhostreceive=enabled

netsh interface ipv4 set interface "loopback" weakhostreceive=enabled

netsh interface ipv4 set interface "loopback" weakhostsend=enabled
```

Note: Obviously first you will need to rename the specific adapters from the default of "Local Area Network Connection 1" to either "net" or "loopback" respectively i.e.

See following link for more information ***http://blog.loadbalancer.org/direct-server-return-on-windows-2008-using-loopback-adpter/***

Final Test

Use "ipvsadm" to list down current statistics of ldirectord. Make sure that both real servers IPs are listed there and have non-zero value in weight (since we(TM)ve this default setup, it should be 1). If not, then try checking the log file, tcpdump on ldirector and apache logs on real servers.If everything works good, you(TM)ll see changing content when browsing to http://10.10.10.53/ multiple times (from another system outside these cluster nodes). Then stop httpd on one web server, browse to the URL again and all requests should now be served from the other web server.

```
ipvsadm -L -n
```

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.10.10.53:http lblc
192.168.200.102:http             Route   1      0          0
192.168.200.103:http             Route   1      0          0
```

For more information use following commands:

```
ipvsadm -L -nc

ipvsadm -L -n --rate

ipvsadm -L -n --stats
```

References

http://bashukhan.com/ha-lb-cluster-on-centos5-without-actual-heartbeat-p
http://www.howtoforge.com/setting-up-a-high-availability-load-balancer-with-haproxy-heartbeat-on-debian-lenny
http://www.linuxvirtualserver.org/