Setting Up Web Servers Load Balancing Using 'POUND' on RHEL/CentOS

by Thilina Uvindasiri | Published: July 10, 2015 | Last Updated: July 16, 2015

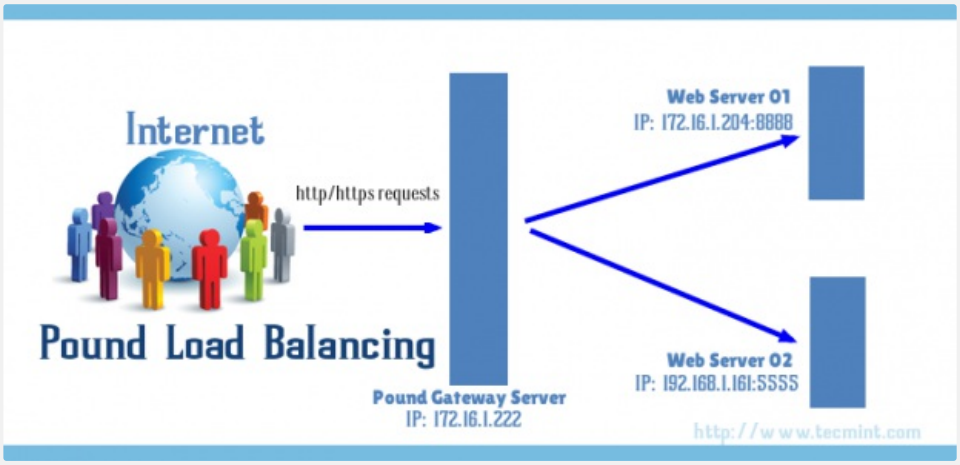
POUND is a load balancing program developed by ITSECURITY Company. It is a lightweight open source reverse proxy tool which can be used as a web-server load balancer to distribute load among several servers. There are several advantages POUND gives to end user which are very convenient and does the job right.

- Supports virtual hosts.
- Configurable.
- When a backend server is failed or recovered from a failure, it detects it automatically and bases its load balancing decisions according to that.
- It rejects incorrect requests.
- No specified browser or webservers.

Let's have a look at how can get this hack done.

First of all you will need a scenario for better understanding about getting this done. So I will use a scenario where there are two webservers and one gateway server which needs to balance the requests comes to gateway server to webservers.

Pound Gateway Server : 172.16.1.222
Web Server 01 : 172.16.1.204
Web Server 02 : 192.168.1.161



Pound Web Server Load Balancer

Step1: Install Pound Load Balancer on Gateway Server

1. The easiest way to install **Pound** is using pre-compiled RPM packages, you can find RPMs for RedHat based distributions at:

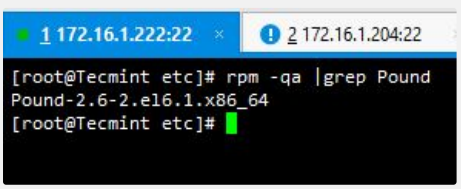
- http://www.invoca.ch/pub/packages/pound/

Alternatively, Pound can be easily installed from the EPEL repository as shown below.

```
# yum install epel-release
# yum install Pound
```

After Pound installed, you can verify whether it is installed by issuing this command.

```
# rpm –qa |grep Pound
```



Install Pound Load Balancer

2. Secondly, you need **two web-servers** to balance the load and make sure you have clear identifiers in order to test the pound configuration works fine.

Here I have two servers bearing IP addresses **172.16.1.204** and **192.168.1.161**.

For ease of use, I have created python SimpleHTTPServer to create an instant webserver on both servers. Read about python SimpleHTTPServer

In my scenario, I have my **webserver01** running on **172.16.1.204** through port **8888** and **webserver02** running on **192.168.1.161** through port **5555**.

```
[root@test webserver01]# python -m SimpleHTTPServer 8888
Serving HTTP on 0.0.0.0 port 8888 ...
```

Pound Webserver 1

```
[root@localhost webserver2]# python -m SimpleHTTPServer 5555
Serving HTTP on 0.0.0.0 port 5555 ...
```

Pound Webserver 2

Step 2: Configure Pound Load Balancer

3. Now it's time to make the configurations done. Once you have installed pound successfully, it creates the pound's config file in **/etc**, namely **pound.cfg**.

We have to edit the server and backend details in order to balance the load among the webservers. Go to **/etc** and open **pound.cfg** file for editing.

```
# vi /etc/pound.cfg
```

Make the changes as suggested below.

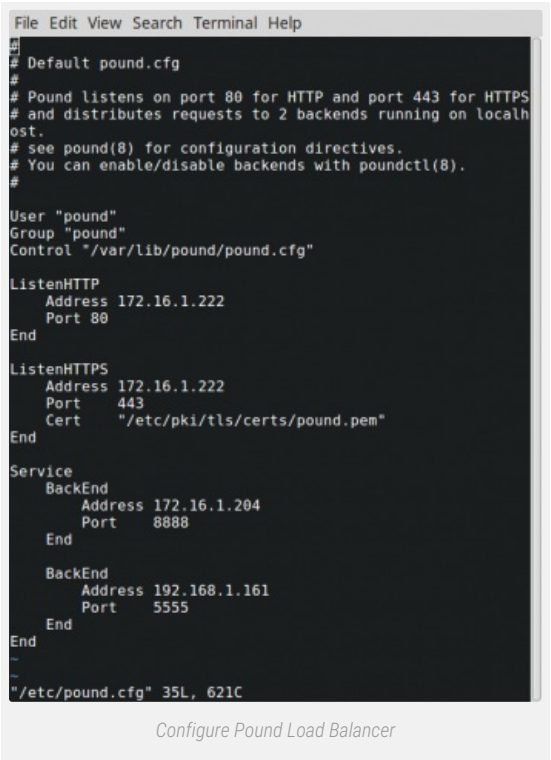
```
ListenHTTP
    Address 172.16.1.222
    Port 80
End

ListenHTTPS
    Address 172.16.1.222
    Port    443
    Cert    "/etc/pki/tls/certs/pound.pem"
End

Service
    BackEnd
        Address 172.16.1.204
        Port    8888
    End

    BackEnd
        Address 192.168.1.161
        Port    5555
    End
End
```

This is how my **pound.cfg** file looks like.



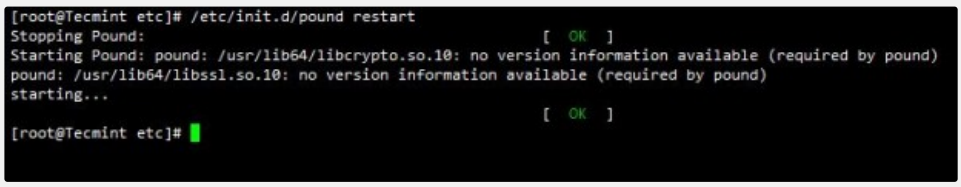
Configure Pound Load Balancer

Under the "**ListenHTTP**" and "**ListenHTTPS**" tags, you have to enter the **IP address** of the server you have installed POUND.

By default a server handles HTTP requests though port **80** and HTTPS requests through port **443**. Under the "**Service**" tag, you can add any amount of sub tags called "**BackEnd**". BackEnd tags bears the IP addresses and the port numbers which the webservers are running on.

Now save the file after editing it correctly and restart the **POUND** service by issuing one of below commands.

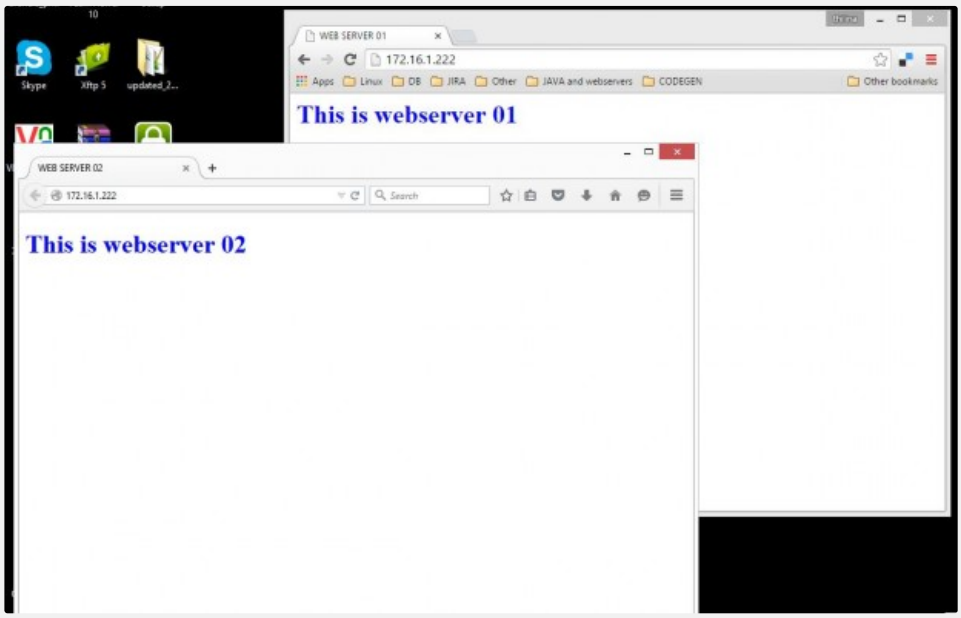
```
# /etc/init.d/pound restart
OR
# service pound restart
OR
# systemctl restart pound.service
```



Start Pound Load Balancer

4. Now it's time to check. Open two web browsers to check whether our configurations work fine. In the address bar type your **POUND** gateway's IP address and see what appears.

First request should load the first **webserver01** and second request from the other web browser should load the second **webserver02**.



Check Pound Load Balancing

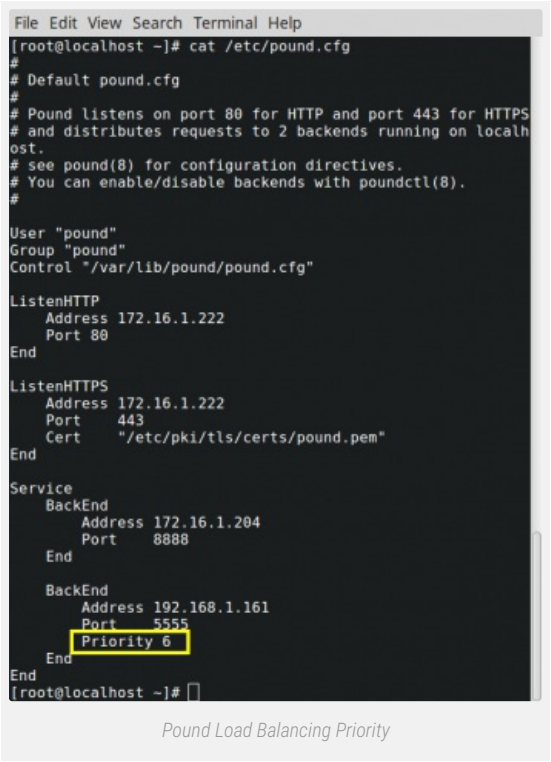
Furthermore, think of a scenario like if you have **two webservers** to load balance and one of the server's performance is good and other's performance is not so good.

So when load balancing among them, you will have to consider for which server you have to put more weight on. Obviously for the server with good performance specs.

To balance the load like that, you just have to add a single parameter inside the **pound.cfg** file. Let's have a look at it.

Think server **192.168.1.161:5555** is the better server. Then you need put more requests flow to that server. Under the "**BackEnd**" tag which is configured for **192.168.1.161** server, add the parameter "**Priority**" before the End tag.

Look at below example.



Pound Load Balancing Priority

The range we can use for the "**Priority**" parameter is between **1-9**. If we do not define it, default value of **5** will be assigned.

Then load will be balanced equally. If we define the Priority number, **POUND** will load the server with higher priority number more oftenly. So in this case, **192.168.1.161:5555** will be loaded more often than the server **172.16.1.204:8888**.

Step 3: Planning Emergency Breakdowns

Emergency Tag: This tag is used to load a server in case of all the back end servers are dead. You can add it before the last End tag of **pound.cfg** as follows.

```
"Emergency
        Address 192.168.5.10
        Port        8080
  End"
```

6. POUND always keep track of which backend servers are alive and which are not. We can define after how many seconds POUND should checkout the backend servers by adding "**Alive**" parameter in **pound.cfg**.

You can use the parameter as "**Alive 30**" for set it to **30** seconds. Pound will temporarily disable the backend servers which are not responding. When we say not responding server may be dead or cannot establish a connection at that moment.

POUND will check the disabled backend server after every time period you have defined in the **pound.cfg** file in case if the server could establish a connection, then POUND can get back to work with the server.

7. POUND daemon will be handled by **poundctl** command. By having that we don't need to edit the **pound.cfg** file and we can issue **Listner Server**, **BackEnd** servers and sessions etc. via a single command.

Syntax: poundctl -c /path/to/socket [-L/-l] [-S/-s] [-B/-b] [-N/-n] [-H] [-X]

- **-c** defines path to your socket.
- **-L** / **-l** defines the listener of your architecture.
- **-S** / **-s** defines the service.
- **-B** / **-b** defines the backend servers.

See **poundctl** man pages for more information.

Hope you enjoy this hack and discover more options regarding this. Feel free to comment below for any suggestions and ideas. Keep connected with Tecmint for handy and latest How To's.

Read Also: Installing XR Crossroads Load Balancer for Web Servers

Sharing is Caring...

Tags: Apache load balancer pound

Thilina Uvindasiri View all Posts

I'm a BSc Special (Hons) graduate in Information Technology and works as an Engineer-Systems. Love to work, explore and research Linux and play rugby as a hobby.

NEXT STORY
GNU/Linux Advanced System Administration Free eBook – Download Now

PREVIOUS STORY
How to Use and Execute PHP Codes in Linux Command Line – Part 1

YOU MAY ALSO LIKE...

Apache Virtual Hosting: IP Based and Name Based Virtual Hosts in RHEL/CentOS/Fedora

6 JAN, 2014

How to Sync Two Apache Web Servers/Websites Using Rsync

11 NOV, 2013

9 RESPONSES

Comments 9 Pingbacks 0

Pankaj Gupta July 11, 2018 at 1:44 pm

Hi,

I have 4 Pound servers on Linux environment where Dynamic configuration is done. Is there any method to take backup of Dynamic Configuration. Cause when ever the pound is restarted the configuration is lost.

Please suggest
Best Regards
Pankaj

Reply

Surender August 2, 2016 at 12:47 pm

Dear Thilina Uvindasiri,
I want to learn apace load balancing or HA clustering concept without sharing storage.
So please help me to learn this concept.

Reply

Ravi Saive August 2, 2016 at 12:57 pm

@Surendra,

For Apache Load Balancing and HA clustering concept I suggest you to read our articles:

Setup High-Availability Load Balancer with HAProxy to Control Apache Traffic
Setup High-Performance HHVM with Apache

Reply

surender August 2, 2016 at 1:24 pm

Dear Saive,

if there any way to upload and update the document root of apache (centos) . it will reflect to fail over node or may another server.

Reply

Ravi Saive August 2, 2016 at 3:19 pm

@Surendra,

A better way to update or sync apache web server files using rsync, for more details read our article Sync Two Apache Web Server Files Using Rsync

Reply

Paresh July 15, 2015 at 3:35 pm

Hi
what kind of redundancy option available for gateway server?

Reply

Thilina Dhanushka July 16, 2015 at 12:33 pm

Hi Paresh,

Setting up a fail-over for the LB server would do that. Upcoming RedHat clustering for a failover scenario lessons will help you do that.

Reply

Thilina Uvindasiri July 13, 2015 at 11:46 am

Hi Konrad,

POUND is designed for load balancing of application traffic. To my knowledge, you can't use POUND for balancing domain traffic.

Reply

konrad July 10, 2015 at 7:11 pm

Hi,
can I configure POUND for openLDAP ?

Reply

GOT SOMETHING TO SAY? JOIN THE DISCUSSION.

Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Notify me of followup comments via e-mail. You can alsosubscribe without commenting.

Post Comment

How to Add Linux Host to Nagios Monitoring Server Using NRPE Plugin

How to Install Nagios 4.4.5 on RHEL/CentOS 8/7 and Fedora 30

Install Cacti (Network Monitoring) on RHEL/CentOS 8/7 and Fedora 30

How to Install Google Chrome 75 On RHEL/CentOS 7 and Fedora 30

How to Install Ubuntu Alongside With Windows 10 or 8 in Dual-Boot

LINUX EBOOKS

- Introducing Learn Linux In One Week and Go from Zero to Hero
- RedHat RHCE/RHCSA Certification Preparation Guide
- Linux Foundations LFCS/LFCE Certification Guide
- Postfix Mail Server Setup Guide for Linux
- Ansible Setup Guide for Linux
- Django Setup Guide for Linux
- Awk Getting Started Guide for Beginners
- Citrix XenServer Setup Guide for Linux

LINUX MONITORING TOOLS

Guider – A System Wide Linux Performance Analyzer

Conky – The Ultimate X Based System Monitor Application

GoAccess (A Real-Time Apache and Nginx) Web Server Log Analyzer

How to Create a Centralized Log Server with Rsyslog in CentOS/RHEL 7

4 Useful Commandline Tools to Monitor MySQL Performance in Linux

LINUX INTERVIEW QUESTIONS

10 Interview Questions and Answers on Various Commands in Linux

10 Useful 'ls' Command Interview Questions – Part 2

11 Basic Linux Interview Questions and Answers

15 Basic MySQL Interview Questions for Database Administrators

10 Advance VsFTP Interview Questions and Answers – Part II

OPEN SOURCE TOOLS

6 Best Email Clients for Linux Systems

My Favorite Command Line Editors for Linux – What's Your Editor?

13 Best Open Source Video Players For Linux in 2019

10 Tools to Take or Capture Desktop Screenshots in Linux

4 Best Linux Boot Loaders