DAWSON COLLEGE – Electronics Engineering Technology Department

Fall 2012

Fundamentals of Web Servers (243-544-DW)

Server Cluster

Submitted by: Matthew Masi, Erick Hernandez, Keinz Rajan

Date Submitted: November 3, 2012

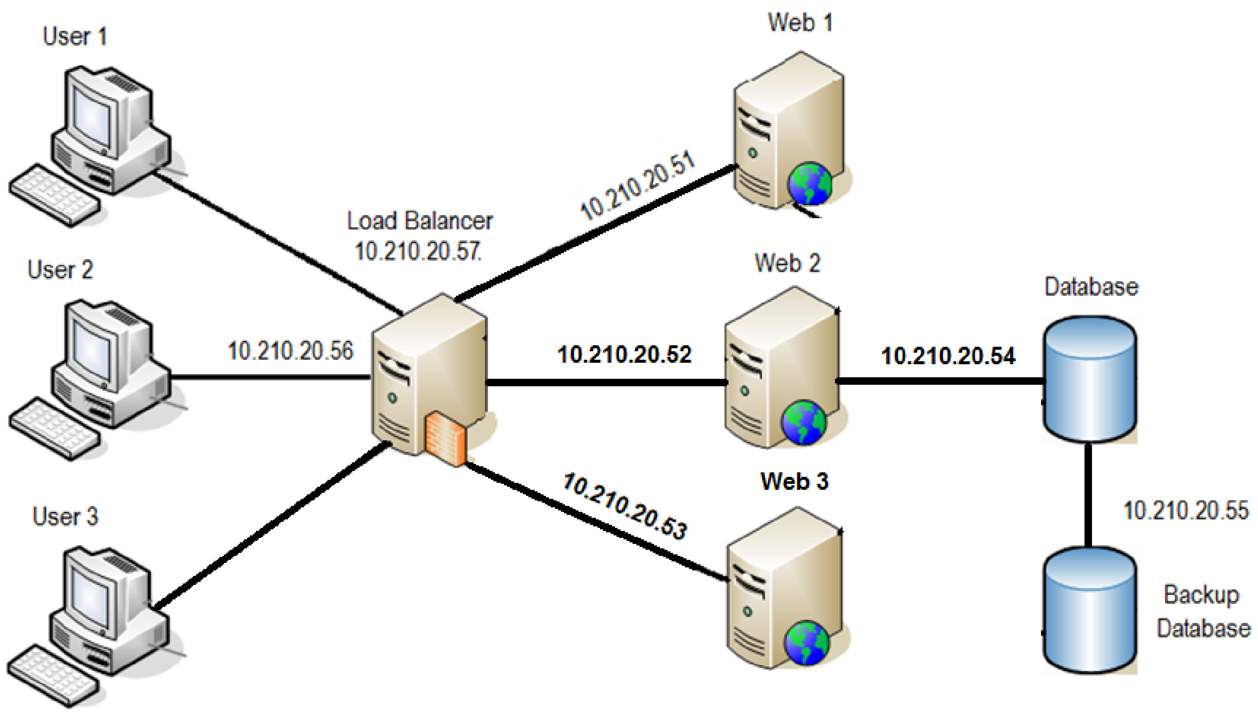
PURPOSE

1. Construct a server cluster to deliver web pages using 6 virtual Linux machines.
2. The web application will use Apache, PHP and MySQL.
3. The web application will have a backup database machine.
4. Multiple Apache/PHP machines will be managed using a load balancer
5. One of the machines will be used as a load balancer.

EQUIPMENT & SOFTWARE

- 6 Virtual servers running Centos 6.3
- Computer
- Putty SSH Client

NETWORK LAYOUT



Server	IP Address	Configuration
Web 1	10.210.20.51	Apache/PHP
Web 2	10.210.20.52	Apache/PHP
Web 3	10.210.20.53	Apache/PHP
Database	10.210.20.54	MySQL
Backup Database	10.210.20.55	MySQL
Load Balancer	10.210.20.57	Piranha

APACHE

Apache HTTP Web Server was installed and configured on three servers (10.210.20.51, 10.210.20.52 and 10.210.20.53). This was done as follows:

1. Install Apache using the following command:

   ```
   sudo yum install httpd mod_ssl
   ```

2. Apache does not start automatically after an install so the following command is used to start it:

   ```
   sudo /usr/sbin/apachectl start
   ```

3. After starting apache, the first thing that you will see is the following error:

   ```
   Starting httpd: httpd: Could not reliably determine the server's fully
   qualified domain name, using 127.0.0.1 for ServerName
   ```

   The address 127.0.0.1 is used by the server name by default. The server name can be set for the next time the server is started. This can be done by opening the Apache main "config":

   ```
   sudo nano /etc/httpd/conf/httpd.conf
   ```

   Towards the end of the httpd.conf file, there is a section that starts with ServerName and gives the example:

   ```
   #ServerName www.example.com:80
   ```

   The line should be uncommented and the IP address of the current server replaces "www.example.com". In this example, we are using Web 1 (10.210.20.51):

   ```
   ServerName 10.210.20.51:80
   ```

4. After editing the Apache main "config" file, restart Apache:

   ```
   sudo /usr/sbin/apachectl restart
   ```

   The warning should be gone after this.

5. By default, our version of Centos (Centos 6.3) has a firewall installed which will block access to port 80, which Apache runs on. The following command will open port 80:

   ```
   sudo iptables -I INPUT -p tcp --dport 80 -j ACCEPT
   ```

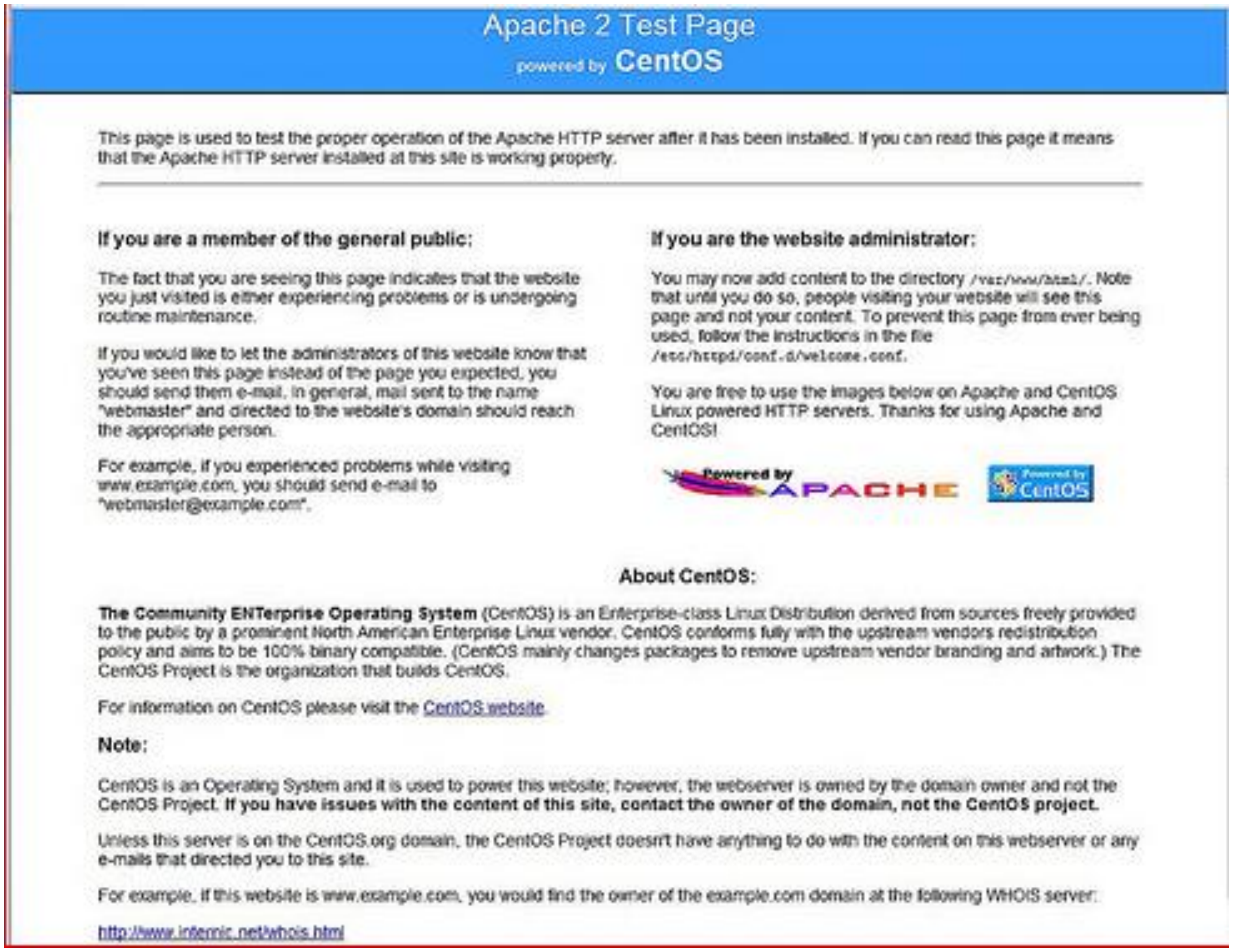
6. This firewall rule should be saved so that your web server will be accessible the next time you reboot. The command to do so is as follows:

```
sudo service iptables save
```

7. Reboot and navigate to:

```
http://10.210.20.51
```

You will see the following page:



Apache 2 Test Page
powered by CentOS

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache and CentOS Linux powered HTTP servers. Thanks for using Apache and CentOS!

About CentOS:

The Community ENTerprise Operating System (CentOS) is an Enterprise-class Linux Distribution derived from sources freely provided to the public by a prominent North American Enterprise Linux vendor. CentOS conforms fully with the upstream vendors redistribution policy and aims to be 100% binary compatible. (CentOS mainly changes packages to remove upstream vendor branding and artwork.) The CentOS Project is the organization that builds CentOS.

For information on CentOS please visit the CentOS website.

Note:

CentOS is an Operating System and it is used to power this website; however, the webserver is owned by the domain owner and not the CentOS Project. **If you have issues with the content of this site, contact the owner of the domain, not the CentOS project.**

Unless this server is on the CentOS.org domain, the CentOS Project doesn't have anything to do with the content on this webserver or any e-mails that directed you to this site.

For example, if this website is www.example.com, you would find the owner of the example.com domain at the following WHOIS server:

http://www.internic.net/whois.html

8. Now that we have Apache installed and working properly, we need to make sure that it's set to start automatically when the web server is rebooted:

```
sudo /sbin/chkconfig httpd on
```

To check if this was done successfully:

```
sudo /sbin/chkconfig --list httpd
```

The output should be as follows if it is working:

```
       httpd           0:off        1:off   2:on    3:on    4:on    5:on
6:off
```

After we got Apache working properly on the web server, we created a file called index.html in the /var/www/html folder which would be the page that appears when navigating the IP address of the web server. The contents of the file are as follows:

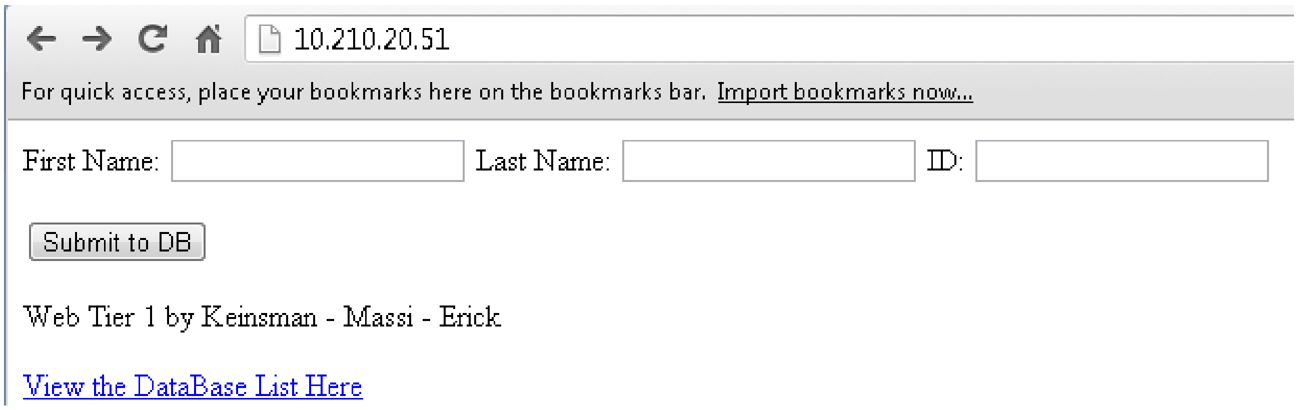
```
<html>
<body>

<form action="http://10.210.20.53/insert.php" method="post"
enctype="multipart/form-data">
            First Name: <input type="text" name="fname" />
            Last Name: <input type="text" name="lname" />
            ID: <input type="text" name="idnumz" />
<p>
<button>Submit to DB</button>
<p>
</form>

<p> Web Tier 1 by Keinsman - Massi - Erick </p>

<a href="http://10.210.20.53/display.php/">View the DataBase List Here</a>
</body>
</html>
```

After creating this file, when navigating to the web servers IP address, we got the following page:


10.210.20.51
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

First Name: Last Name: ID:

Submit to DB

Web Tier 1 by Keinsman - Massi - Erick

View the DataBase List Here

PHP5

After Apache was installed and configured, PHP5 was installed with a few common modules.

1. Install PHP:

```
sudo yum install php-mysql php-devel php-gd php-pecl-memcache php-
pspell php-snmp php-xmlrpc php-xml
```

2. After installation is complete, reload Apache:

```
  sudo /usr/sbin/apachectl restart
```

We used PHP scripts to send information to the MYSQL database that we have installed on machine 10.210.20.53. The following PHP script takes the data entered in the HTML page and sends it into our MYSQL database that was created using Webmin:

```
<!DOCTYPE HTML>
<html>
<?php

$con = mysql_connect("10.210.20.54","root","dawson2012");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }

mysql_select_db("student_db", $con);

$sql="INSERT INTO Stable (Firstname, Lastname, Idbatch )
VALUES
('$_POST[fname]','$_POST[lname]','$_POST[idnumz]','$filename')";

if (!mysql_query($sql,$con))
  {
  die('Error: ' . mysql_error());
  }
echo "1 record added";
mysql_close($con);
?>

<a href="http://10.210.20.54/display.php/">View the Database List Here</a>
</html>
```

Load Balancers & Clusters:

In this example website www.webservers.edu is hosted on ip address 131.12.14.12 which is a public IP address. The server is running multiple services such HTTP, PHP, MySQL, Video Streaming, News Feed, Forums and other services. The server taking into consideration that it can only serve up to 500 concurent users, is limited when it comes to processing capabilities. It becomes a burden for the server to be able to handle more than its capable of doing.

There are other factors that need to be taken in to consideration such as security, maintenance, and hardware failure. In terms of security the IP address of the server is exposed to the public if its not being hosted by a Web Hosting Company, in this case it becomes vulnerable to attacks by Hackers who might wish to take down the server or run DDOS packets to it.

Maintenance is also crucial to the company that is running this server, in a case where maintenance – need to be done on the server will have to be temporarily limited or shut down. This can result in unfavorable circumstances where company might have sales / customer loss.

Hardware is also an important factor. Servers are known to fail, it can range from burnt memories, storage failure, power failure and if upgrade needs to be performed or Server need to be replaced it places an overhead for the web administrator to bring up the services on time.

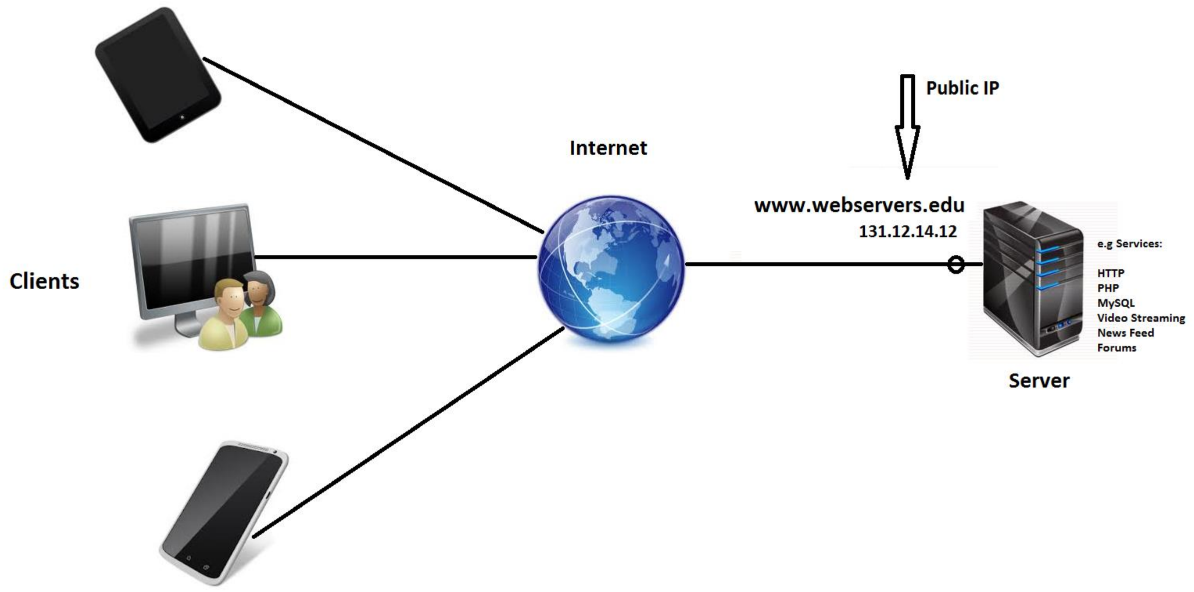


Figure 1: Web Server - without the use of Load Balancers

Internet is becoming a huge market for companies and organizations, and when it comes to serve applications with reliability and redundancy it’s always a good idea to consider Load Balancers. Load Balancing overcomes the overloading problems of the servers and can direct internet traffic in an efficient manner. Load Balancers come in three different flavors – hardware , software – cloud . They work by sending network traffic to a virtual IP address which is an address connected to the load balancer. Then load balancer depending on the type of architecture will request the response to the end user or the load balancer.

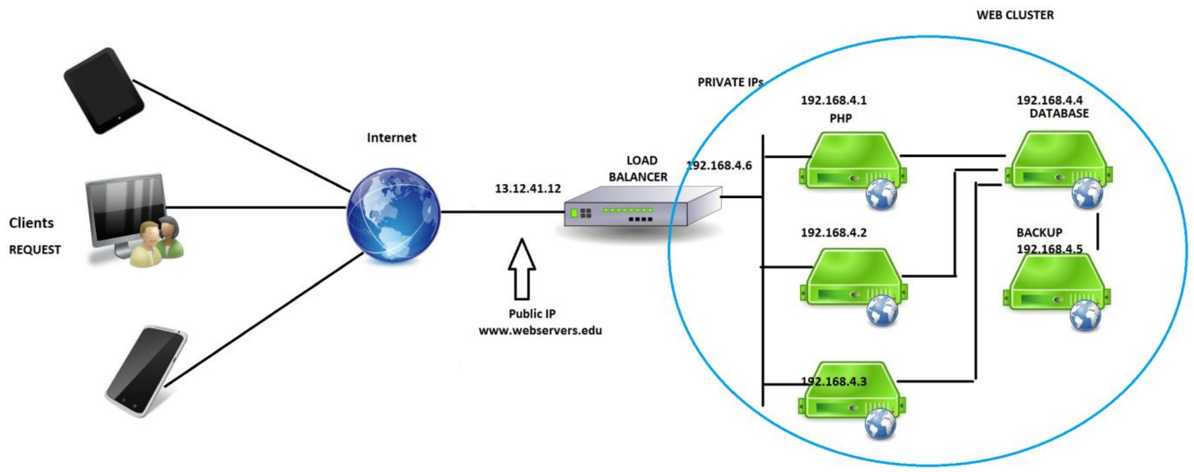


Figure 2: Web Cluster Load Balancer

The advantages of using Load Balancers are that companies can create clusters of servers for different applications. Doesn’t have to be specifically for Port:80 – HTTP, it can be used for SSL, IMAP, and other applications. In terms of security it is much more secure because only one IP address is known and whereas multiple servers can be within the organization and assigned a Private IP Address. Now using the preferred algorithm / architecture multiple servers can serve thousands of users depending on its processing capabilities but not having one server having the work load all by itself. The load balancer can even open ports for specific applications.

Maintenance and Hardware updates become such an ease because the website does not have to be taken down. Selected machines can undergo update and maintenance and the load balancer will still relay the active machines. In case a hardware failure with one of the server it will still sense for active connection and only response with the machines that are still up.

There are a wide range of LVS - Linux Virtual Servers which can perform load balancing in the Unix environment some of the known load balancers are nginx, pound, ultramonkey and piranha. Piranha is a good choice for the Red Hat operating system. Since we had CentOS 6.3 running for our servers it was efficient to use a mix of GUI and CLI configuration of Piranha.

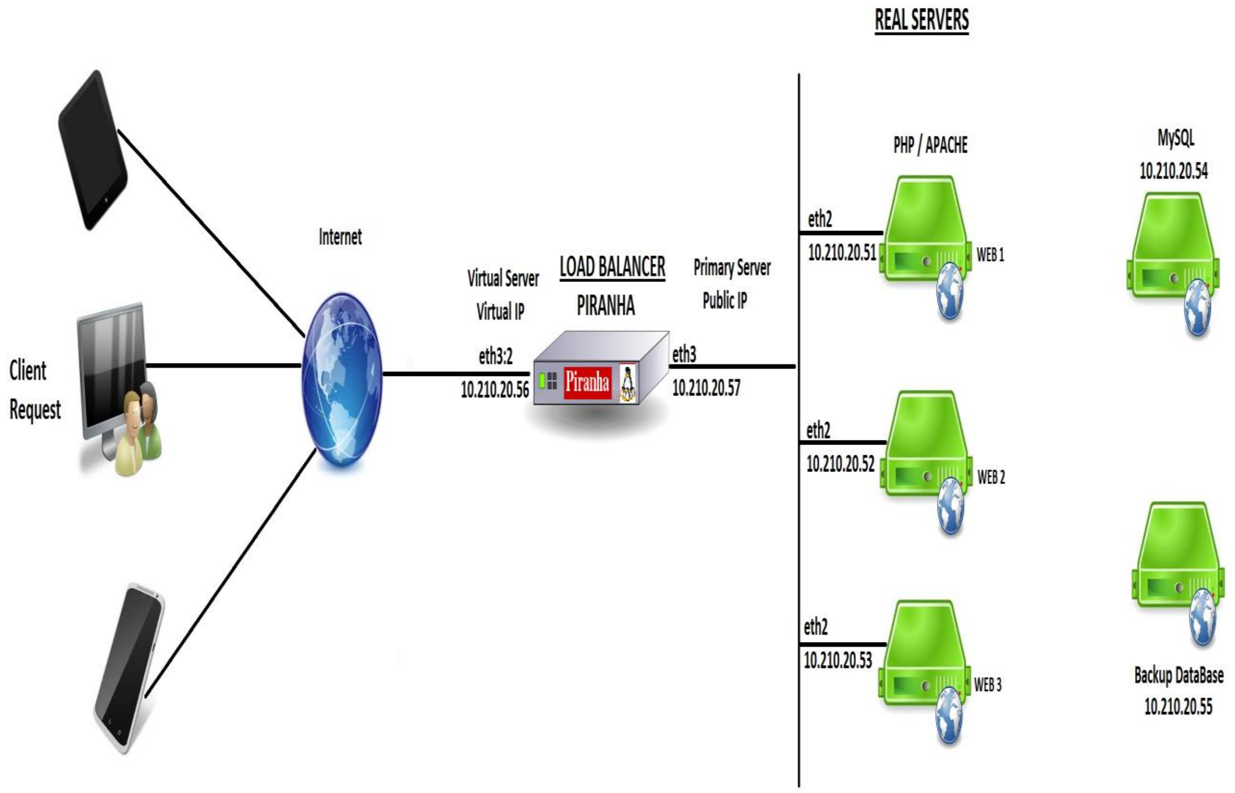
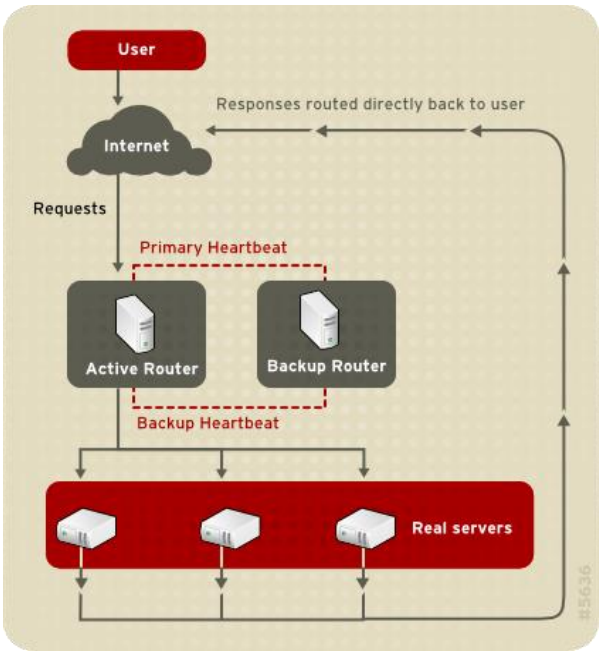


Figure 3: Group 1 - Web Topology

Direct Routing :

Direct Routing is the method where packets by the clients are directly forwarded to the real server. The real servers our are web application tier 10.210.20.51, 10.210.20.52, 10.210.20.53 – these must be configured to accept traffic from the virtual server's IP address, this is configured using arptables. The load balancer receives requests from the virtual IP and then based on the scheduling algorithm the real servers will respond back to the clients.



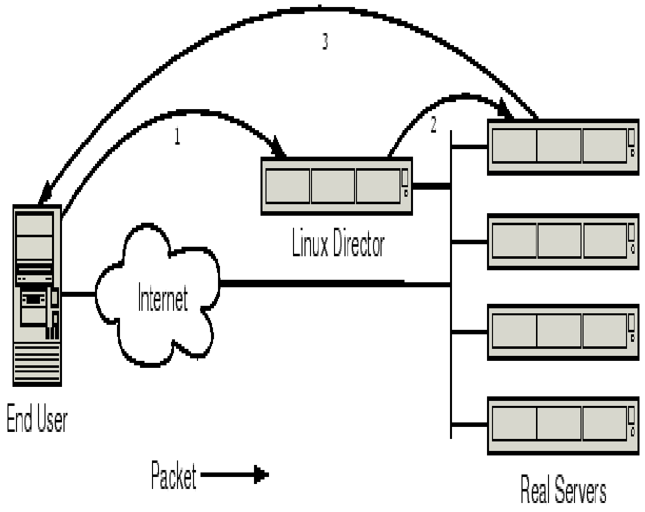


Figure 4: Direct Routing - Source:
http://www.centos.org/docs/5/html/Virtual_Server_Administration/s2-lvs-directrouting-VSA.html
www.ultramonkey.org

ROUND ROBIN:

There are multiple scheduling algorithms, the one which would be useful for us the Round-Robin Scheduling . Round Robin is the simplest algorithm to implement on the load balancer.This algorithm distributes each request sequentially to multiple servers, so if client x, access 10.210.20.56, they will see web 1, and client 2 will be web 2, and so forth.

HEARTBEAT:

In load balancing it is necessary to use a heartbeat service to monitor the status of the Real Server. It works by sending a heartbeat message periodically to each server. If no heartbeat message is received the load balancer considers that particular server to have failed. The heartbeat is running on port : 539 so its important to open this on all real servers and the load balancer.

ARPTABLES_JF:

ARPTABLES_JF is necessary to be configured on all real servers, this application is used so that each IP address on the real servers can directly route packets. Instead of relaying back the ip address of the real server, the ARP packets are sent “mangled” containing the Virtual IP of the Load Balancer.

PIRANHA CONFIGURATION:

The load balancer will be on machine 10.210.20.56, in order for this to work we configured statically the second interface to be 10.210.20.57 by using VSphere Client on VMware with the assistance of the instructor. The issues we overcome while configuring the second interface is that eth3 did not restart with 10.210.20.57, what we did was reboot the 10.210.20.56.

To make sure that our machine is up to date with the latest updates:

```
# yum update
```

Install Piranha and Ipsvadm

```
# yum install piranha ipvsadm –y
```

Open the following ports

♦ Piranha: 3636

♦ HTTP: 80

♦ Heartbeat: 539

```
# sudo iptables –I INPUT  -p tcp --dport 3636 -j ACCEPT
# sudo iptables –I INPUT  -p tcp --dport 80 -j ACCEPT
# sudo iptables –I INPUT  -p tcp --dport 539 -j ACCEPT
```

Save the firewall rules so that next time upon reboot the web server will be accessible with these ports.

```
# sudo service iptables save
```

At this point it would be ideal to reboot the server.

```
# sudo reboot
```

Start the GUI service

```
# service piranha-gui start
```

Enter the following commands so that upon reboot the web server will start with GUI and Hearbeat PULSE

```
# chkconfig piranha-gui on
# chkconfig pulse on
```

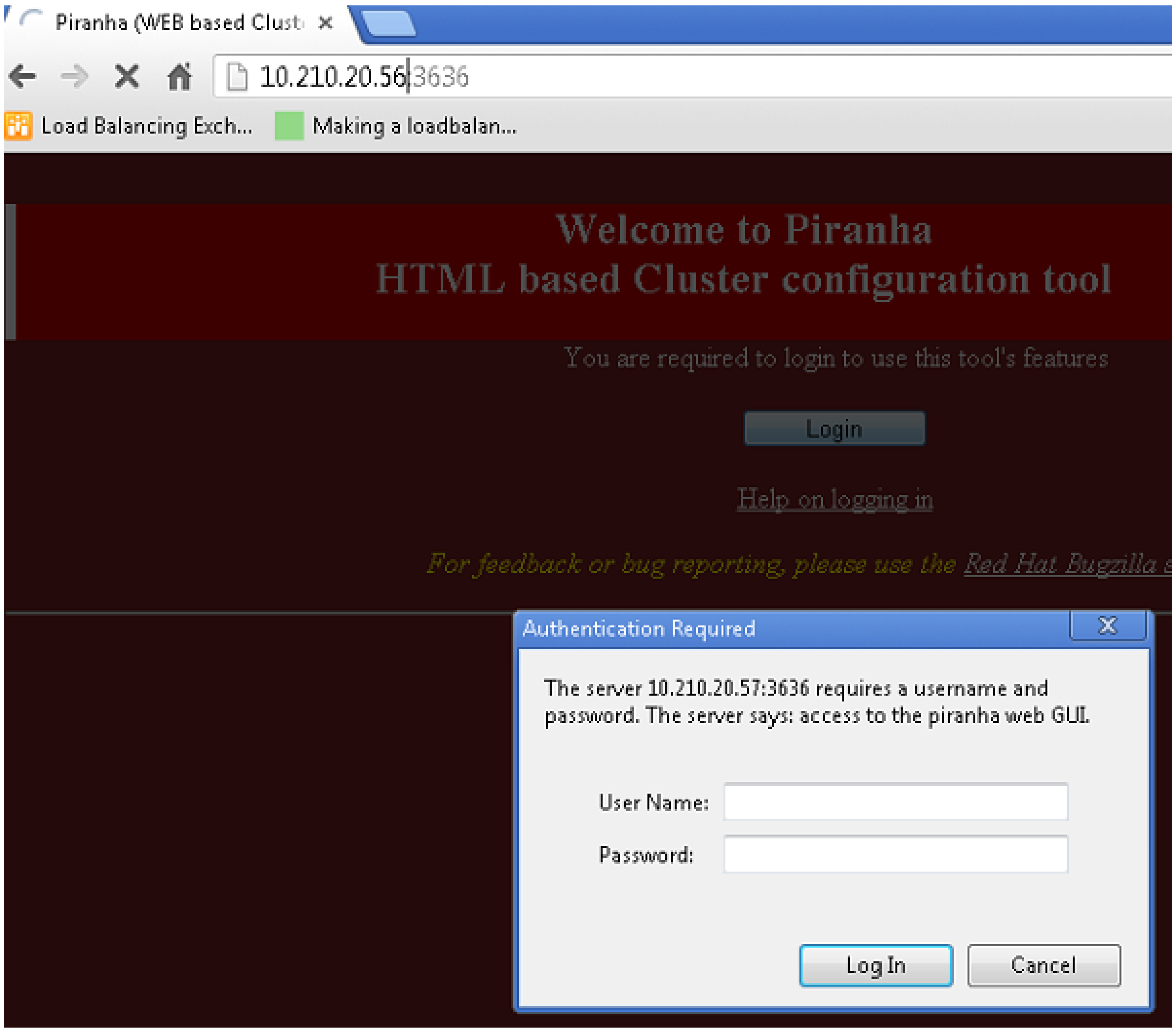
By default the GUI is accessible with the User Name & Password: **piranha** to change the password enter the command

# piranha-passwd
# dawson2012

To enabled IP Forwading enter
# sysctl -w net.ipv4.ip_forward=1

To activate IP Forwarding enter
# systl –p

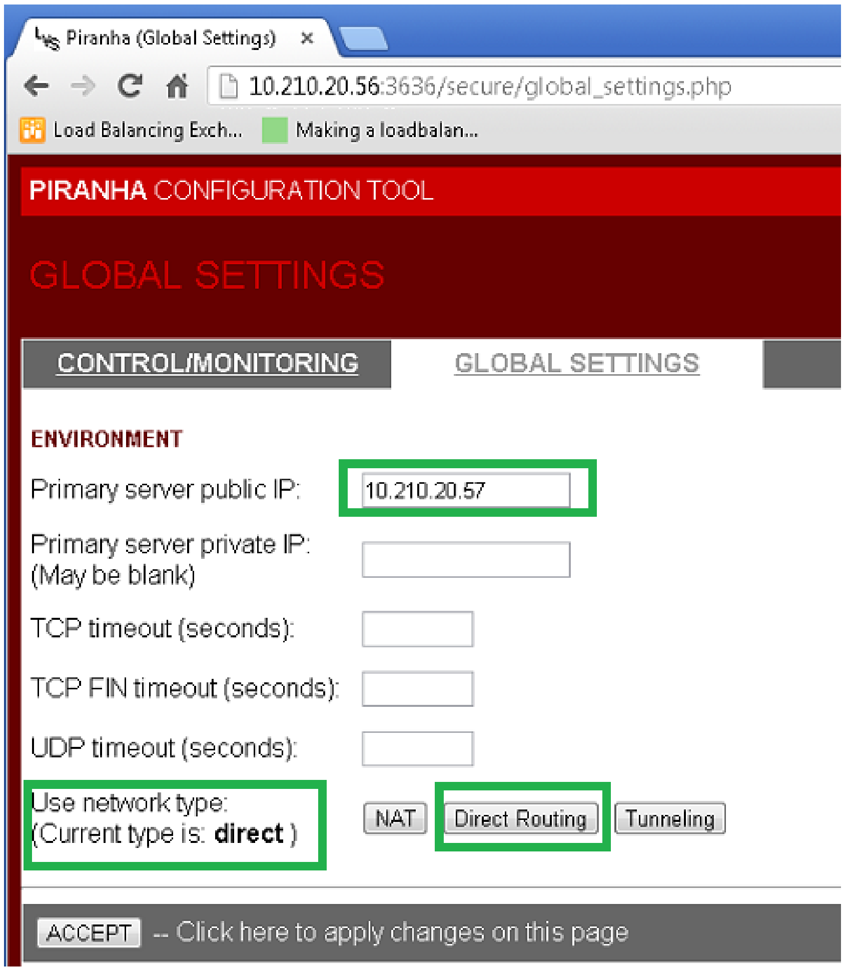
To access the Web-Based Configuration tool open a web browser – preferably chrome at **http://10.210.20.56:3636** and login with Username : **piranha** Password: **dawson2012**



The server 10.210.20.57:3636 requires a username and password. The server says: access to the piranha web GUI.

Figure 5: Piranha Login

Click on the Global Settings Tab and enter the Primary Server public IP at : **10.210.20.57** and make sure the network type is configure to **Direct Routing**.



Primary server public IP: 10.210.20.57

Primary server private IP: (May be blank)

TCP timeout (seconds):

TCP FIN timeout (seconds):

UDP timeout (seconds):

Use network type: (Current type is: **direct**)

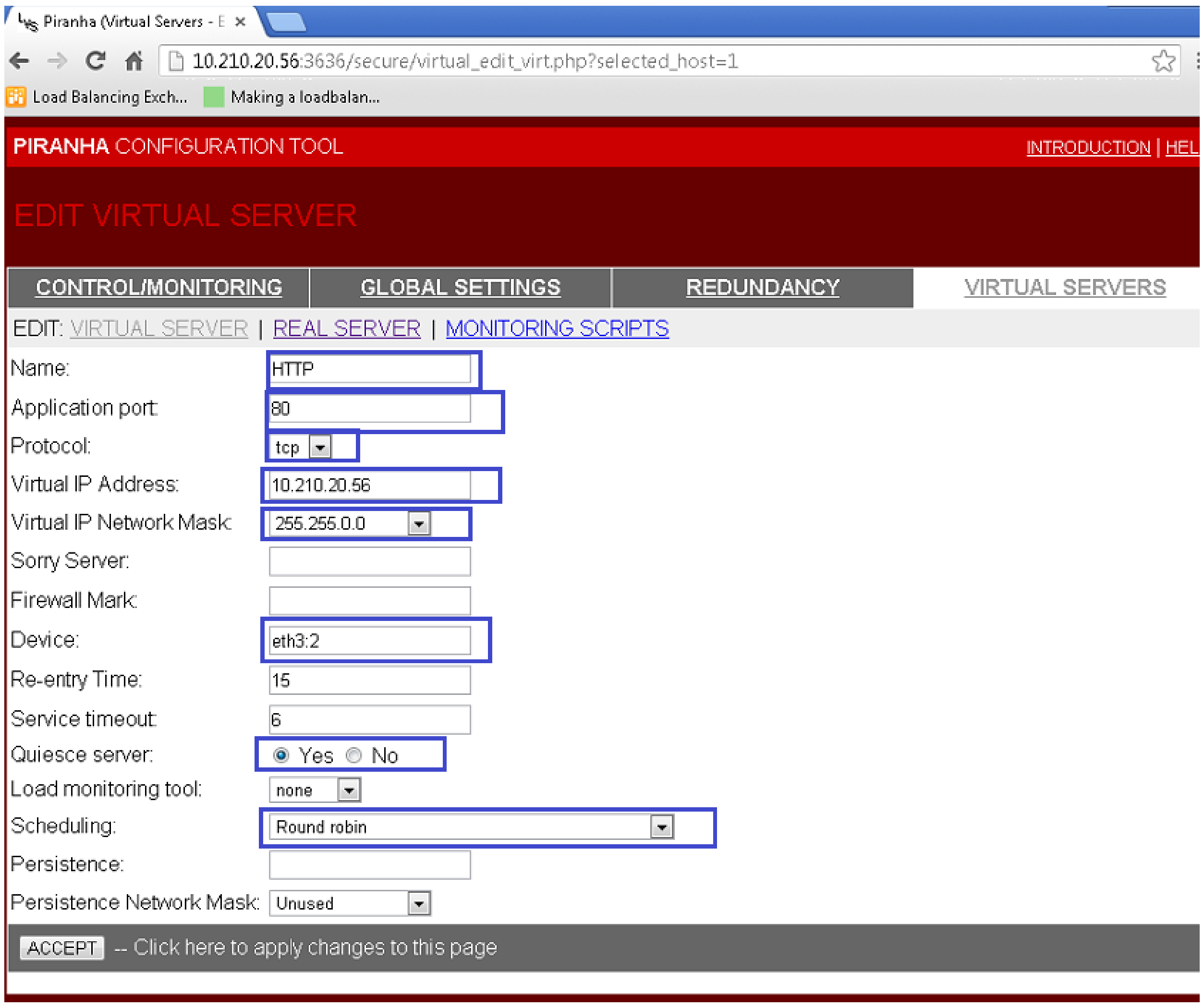
-- Click here to apply changes on this page

Figure 6: Global Settings

Next click on the Virtual Servers tab and add the following parameters Name: <**any**> , Application Port: **80**, Protocol: **TCP**, Virtual IP Address: **10.210.20.56**, Virtual IP Network Mask**: 255.255.0.0**, Device: **eth3:2**, Quiesce server: **Yes**, Scheduling: **Round Robin**. Click Accept and Activate the LVS.

Note: By entering eth3:2, we create a virtual IP address , a.k.a (VIP) this will allow to use multiple IPs on a single network interface. This allows webservers to host multiple SSL encrypted web sites on a single web server or allow cluster to communicate with it. This can be done entering networking-scripts and configuring it on eth3, but in our case Piranha will take of that.

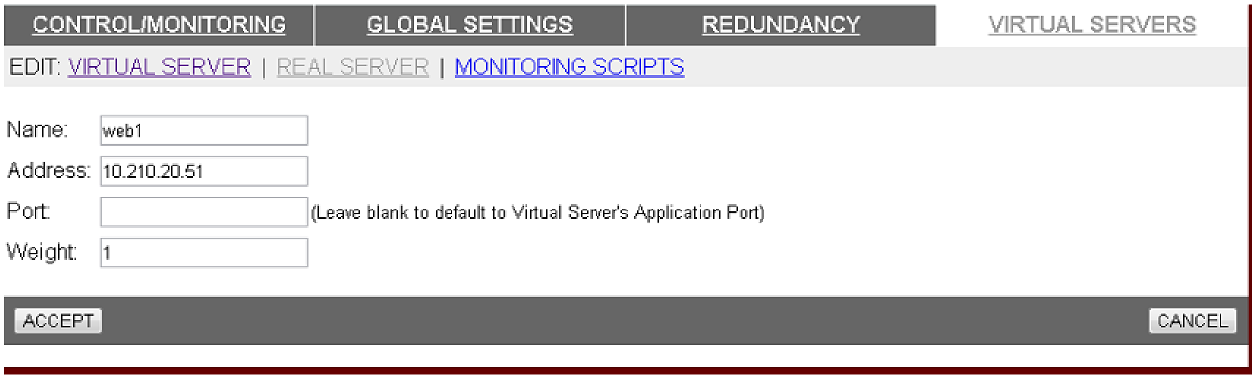
The scheduling algorithm will be selected here as Round Robin.


Piranha (Virtual Servers - E
10.210.20.56:3636/secure/virtual_edit_virt.php?selected_host=1
Load Balancing Exch...
Making a loadbalan...
PIRANHA CONFIGURATION TOOL
INTRODUCTION | HEL
EDIT VIRTUAL SERVER
CONTROL/MONITORING
GLOBAL SETTINGS
REDUNDANCY
VIRTUAL SERVERS
EDIT: VIRTUAL SERVER | REAL SERVER | MONITORING SCRIPTS
ACCEPT
-- Click here to apply changes to this page

Name:	HTTP
Application port:	80
Protocol:	tcp
Virtual IP Address:	10.210.20.56
Virtual IP Network Mask:	255.255.0.0
Sorry Server:	
Firewall Mark:	
Device:	eth3:2
Re-entry Time:	15
Service timeout:	6
Quiesce server:	[✓ Yes] No
Load monitoring tool:	none
Scheduling:	Round robin
Persistence:	
Persistence Network Mask:	Unused

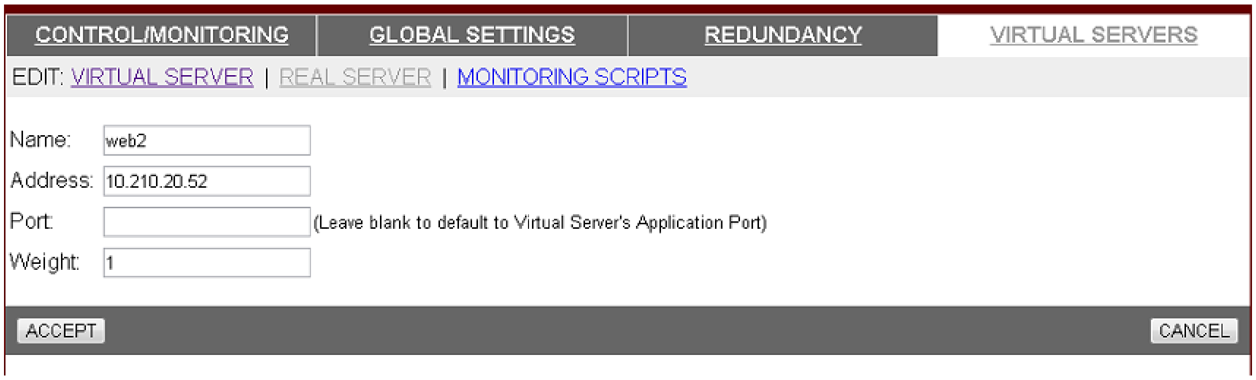
Figure 7: Virtual Server Configuration

We then proceed by configuring the IP address of the real servers in our case 10.210.20.51, 10.210.20.52, and 10.210.20.53. The default port is 80, so since we configured that for the Virtual Server it's not required. The weight is priority, but in this case it's not necessary since Round Robin will equally distribute the load. We left it at 1. Make sure you activate the Real Servers.



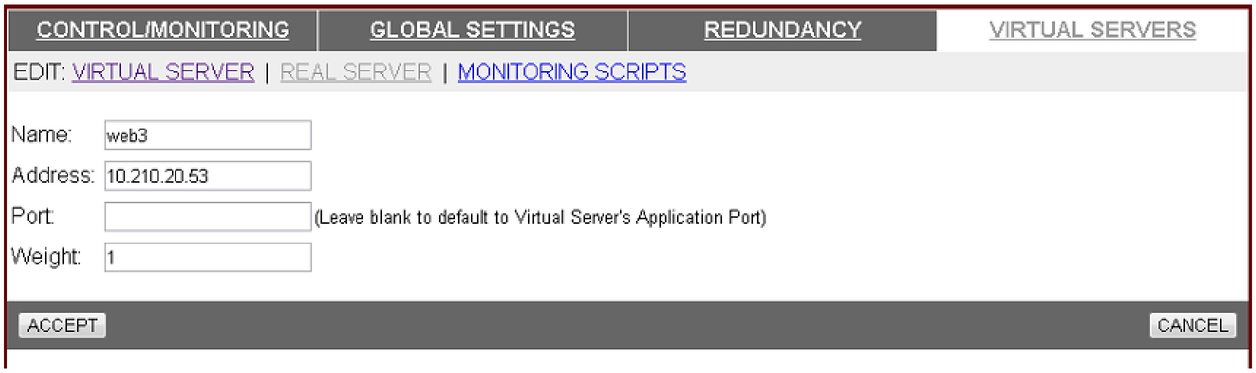
Name: web1
Address: 10.210.20.51
Port: (Leave blank to default to Virtual Server's Application Port)
Weight: 1

Figure 8: Real Server Web1 – 10.210.20.51



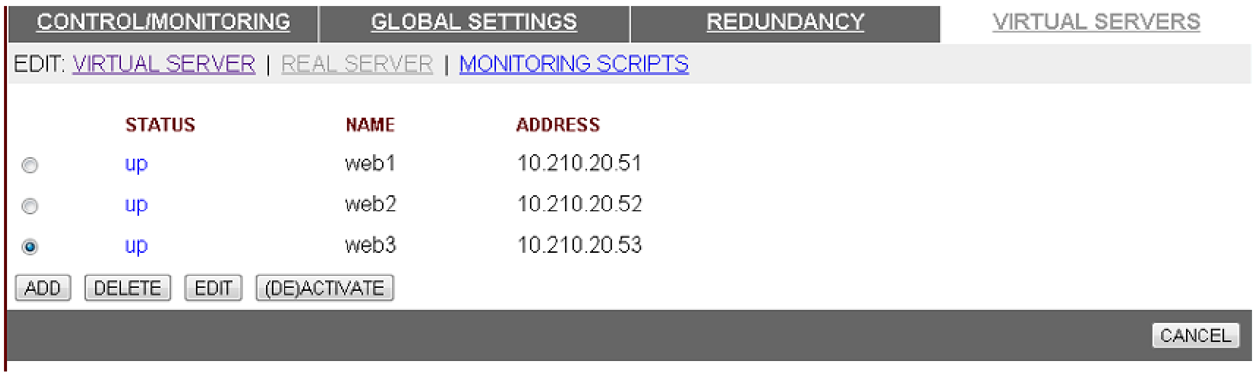
Name: web2
Address: 10.210.20.52
Port: (Leave blank to default to Virtual Server's Application Port)
Weight: 1

Figure 9: Real Server Web2 – 10.210.20.52



Name: web3
Address: 10.210.20.53
Port: (Leave blank to default to Virtual Server's Application Port)
Weight: 1

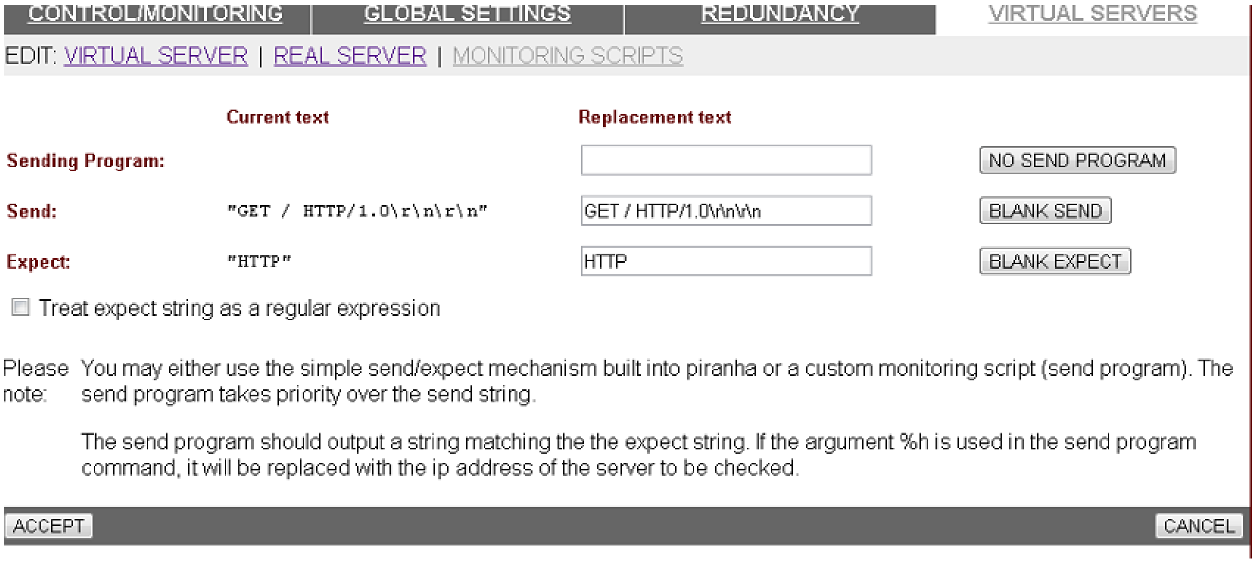
Figure 10: Real Server Web3 – 10.210.20.53



	STATUS	NAME	ADDRESS
○	up	web1	10.210.20.51
○	up	web2	10.210.20.52
●	up	web3	10.210.20.53

Figure 11: Activate All Real Server

Leave the Monitoring Scripts by default.



	Current text	Replacement text	
Sending Program:			NO SEND PROGRAM
Send:	"GET / HTTP/1.0\r\n\r\n"	GET / HTTP/1.0\r\n\r\n	BLANK SEND
Expect:	"HTTP"	HTTP	BLANK EXPECT

☐ Treat expect string as a regular expression

Please note: You may either use the simple send/expect mechanism built into piranha or a custom monitoring script (send program). The send program takes priority over the send string.

The send program should output a string matching the the expect string. If the argument %h is used in the send program command, it will be replaced with the ip address of the server to be checked.

Figure 12: Monitoring Scripts

Restart the Pulse to apply the new configuration:

# service pulse restart

REAL SERVERS – WEB SERVERS:

Since we are using Direct Routing and want to forward the Real Servers with the Virtual Servers IP we need to install **arptables_jf** on all web servers.

Install arptables_jf on 10.210.20.51, 10.210.20.52, and 10.210.20.53

#yum install arptables_jf –y

Configure arptables_jf by entering the following commands

Web Server 1:

# arptables –A IN –d 10.210.20.56 –j DROP
# arptables _A IN –d 10.210.20.56 –j mangle --mangle-ip-s 10.210.20.51

Web Server 2:

# arptables –A IN –d 10.210.20.56 –j DROP
# arptables _A IN –d 10.210.20.56 –j mangle --mangle-ip-s 10.210.20.52

Web Server 3:

# arptables –A IN –d 10.210.20.56 –j DROP
# arptables _A IN –d 10.210.20.56 –j mangle --mangle-ip-s 10.210.20.53

Save the arptables rules
# service arptables_jf save

To permit arptables to start automatically upon boot enter:
# chkconfig arptables_jf on

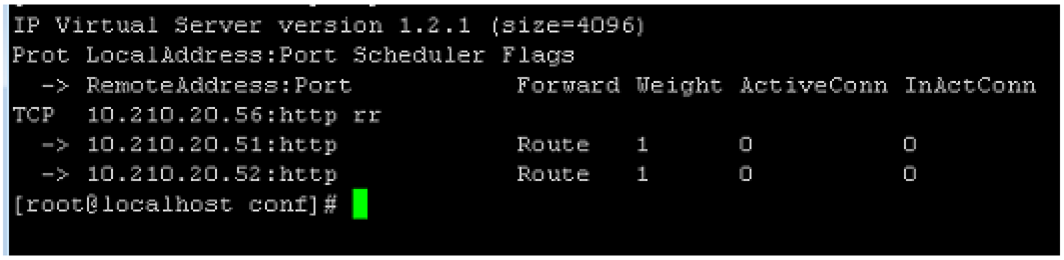
Add the Virtual IP address of the Load Balancer on all real servers by:
# ip addr add 10.210.20.56 dev eth2

In order to start the IP upon boot time, we open **/etc/rc.local** and add the following line:
/sbin/ip addr add 10.210.20.56 dev eth2

We can monitor the status of the Real Servers in CLI or GUI

For CLI

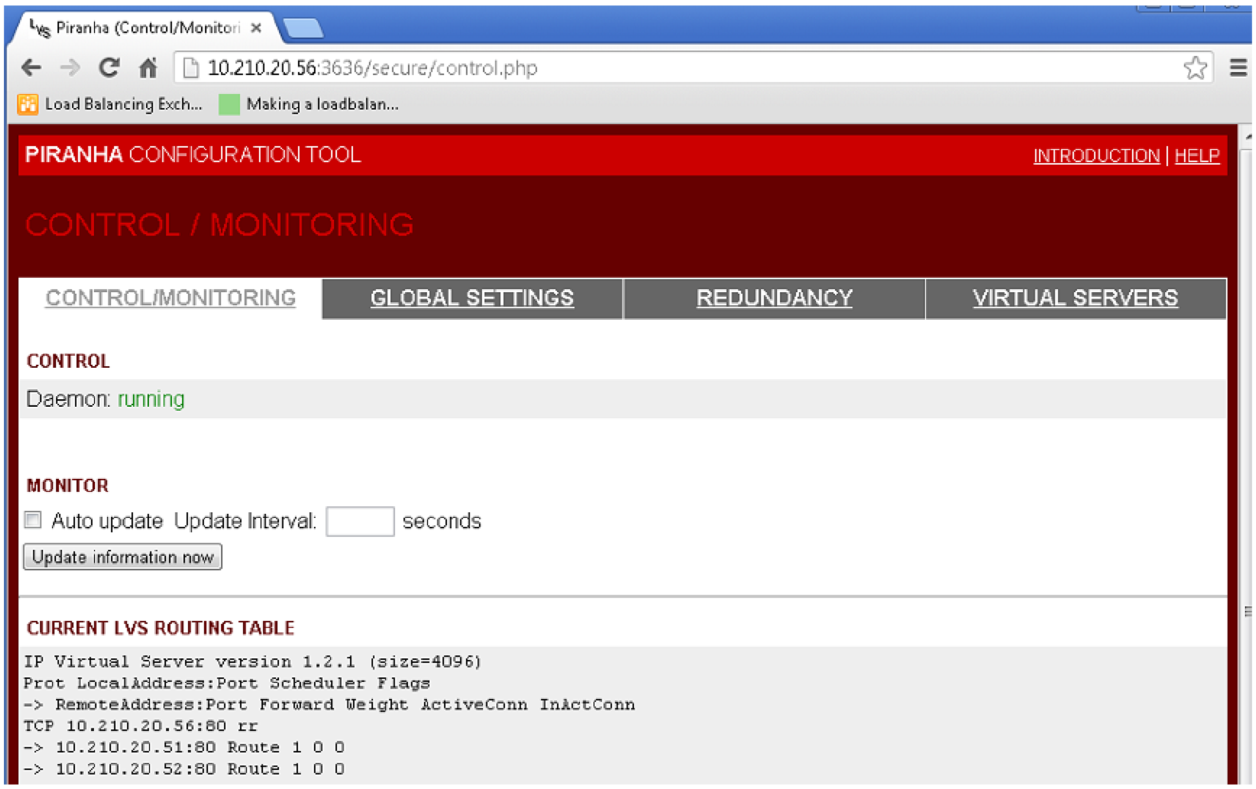
#ipvadm –L



```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.210.20.56:http rr
  -> 10.210.20.51:http            Route   1      0          0
  -> 10.210.20.52:http            Route   1      0          0
[root@localhost conf]#
```

Figure 13: ipvasm -L - from SSH

We can also view the status of each server in the Piranha Configuration Tool at **http:10.210.20.56:3636** under the **CONTROL/MONITORING** tab.



CONTROL

Daemon: running

MONITOR

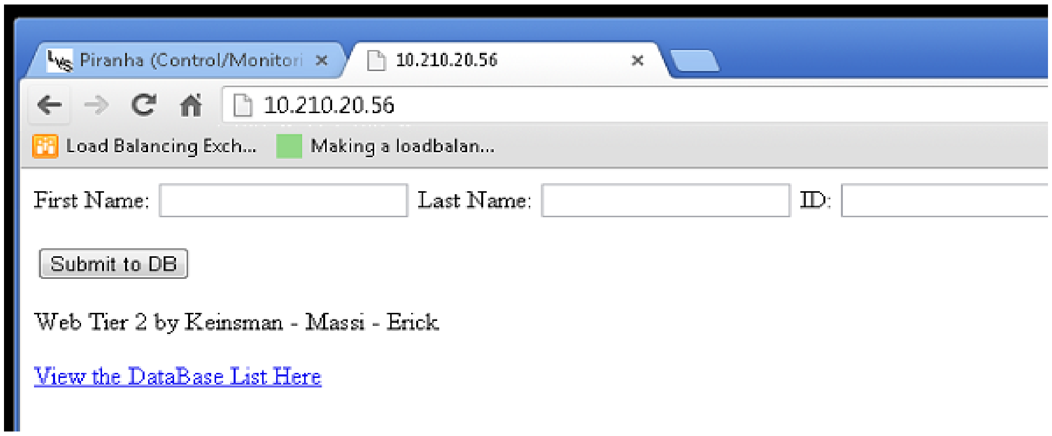
Auto update Update Interval: seconds

CURRENT LVS ROUTING TABLE

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
-> RemoteAddress:Port Forward Weight ActiveConn InActConn
TCP 10.210.20.56:80 rr
-> 10.210.20.51:80 Route 1 0 0
-> 10.210.20.52:80 Route 1 0 0
```

Proof Of Concept:

So from any devices / computers that are under the Electrotech Vlan, you can access the web servers at **10.210.20.56**



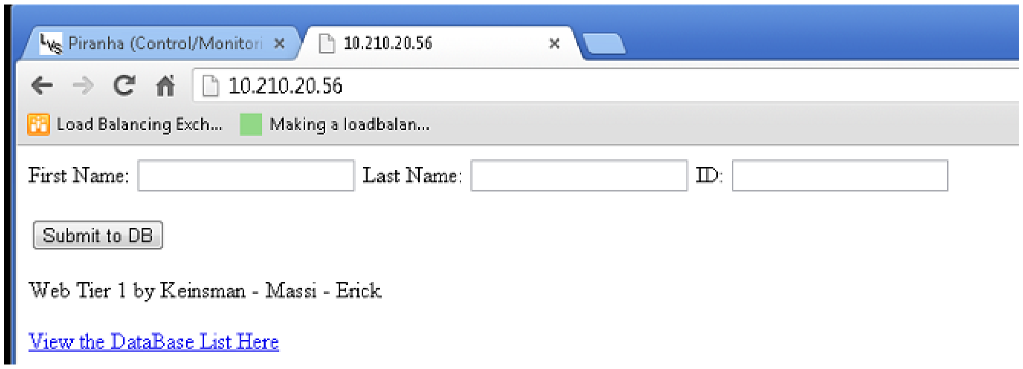
First Name: Last Name: ID:

Submit to DB

Web Tier 2 by Keinsman - Massi - Erick

View the DataBase List Here

Figure 14: Web Tier 2 - 10.210.20.52



First Name: Last Name: ID:

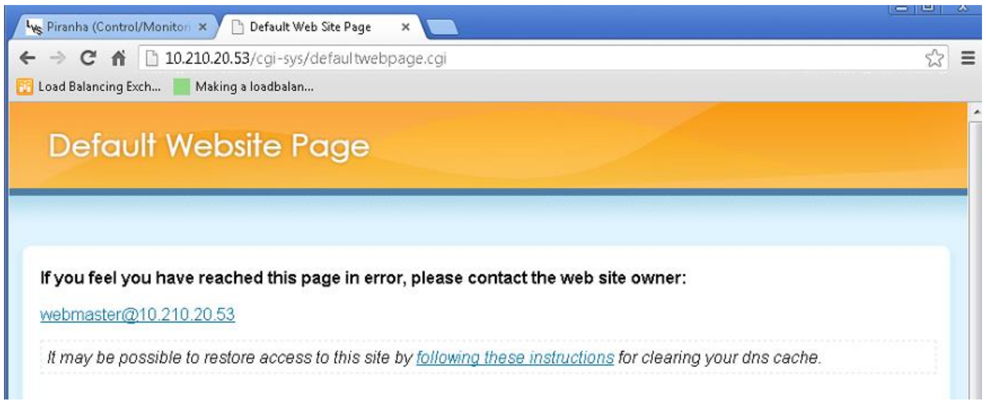
Submit to DB

Web Tier 1 by Keinsman - Massi - Erick

View the DataBase List Here

Figure 15: Web Tier 1 - 10.210.20.51

Since we were facing a couple of issues facing Cpanel , we were not able to demonstrate in this PBL report 10.210.20.56 pick up 10.210.20.53


Piranha (Control/Monitori
Default Web Site Page
10.210.20.53/cgi-sys/defaultwebpage.cgi
Load Balancing Exch...
Making a loadbalan...

Default Website Page

If you feel you have reached this page in error, please contact the web site owner:

webmaster@10.210.20.53

It may be possible to restore access to this site by following these instructions for clearing your dns cache.

Figure 16: CPanel

INSTALLING AND CONFIGURING MYSQL

1. Install MySQL
   yum install mysql-server mysql php-mysql

2. Set the MySQL service to start on boot
   chkconfig --levels 235 mysqld on

3. Start the MySQL service
   service mysqld start

4. Log into MySQL
   mysql -u root

5. Set the root user password for all local domains
   SET PASSWORD FOR 'root'@'localhost' = PASSWORD('*dawson2012*');
   SET PASSWORD FOR 'root'@'localhost.localdomain' = PASSWORD(' *dawson2012*');
   SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD(' *dawson2012*');

6. Exit MySQL
   exit

INSTALLING WEBMIN

1. download the Webmin rpm package
   wget http://prdownloads.sourceforge.net/webadmin/webmin-1.610-1.noarch.rpm

2. Install webmin
   rpm -U webmin-1.610-1.noarch.rpm

3. Access webmin
   http://server-ip:10000

Generating rsa keys and

ssh-keygen -t rsa

putting them in the other server

ssh-copy-id -i ~/.ssh/id_rsa.pub 10.210.20.75

Script on the database server

```
#!/bin/bash
mysqldump -u root pbl_db > /etc/pbl_db.backup
scp /etc/pbl_db.backup root@10.210.20.55:/etc
```

Script on the backup server

```
#!/bin/bash
mysql -u root -pdawson2012 pbl_db < /etc/pbl_db.backup
```

added one line to /etc/crontab

```
SHELL=/bin/bash

PATH=/sbin:/bin:/usr/sbin:/usr/bin

MAILTO=root

HOME=/

0-59 * * * * root /etc/scripts/script.sh
```

Minutes and scribe notes

Meeting 1:

Discussion of problem:

- create a cluster

- 6 virtual servers

- Apache, PHP, and MySQL must be installed

- 1 server is a load balancer

- 1 server is a database backup

We will be running Centos 6.3 on all our servers. Our load balancer server will have piranha installed. Apache will be installed on the three web servers along with PHP and MySQL will be installed on both the database and backup database. Network diagram:

Meeting 2:

We are given 6 IP addresses to use in our cluster. The IP assignment is as follows:

10.210.20.51 --> Web 1

10.210.20.52 --> Web 2

10.210.20.53 --> Web 3

10.210.20.54 --> MysQL database

10.210.20.55 --> backup database

10.210.20.56 --> load balancer

Installing Apache and PHP on the three web servers --> link:

- install Apache (yum install httpd mod_ssl)

- start Apache

- modify httpd.conf (set server name as the ip of the server)

- allow port 80 through the firewall

- restart apache and reboot server

- install PHP with a few given modules

- restart Apache and reboot

Meeting 3:

Installing and configuring piranha on load balancer (10.210.20.56):

We need to open up a second ethernet interface so that we have a public ip and a virtual ip. The virtual ip will be set to 10.210.20.57 and the public ip to 10.210.20.56. This will all be done through the piranha gui which can be accessed by allowing port 3636 through the firewall and typing in the following socket in the address bar: 10.210.20.57:3636. Arp tables must also be configured between the three web servers that will exchange during the load balancing process. Round robin option of selecting which web server to access.

Meeting 4:

Installing MySQL database and backup database:

- Run cronjob scripts on both the database and backup database to back up the database.

Source:

To accomplish this PBL, we had to refer back and forth on a couple of websites and understand arptables, direct routing, heartbeat, round robin, and firewall ports.

• Centos: Configure Piranha as Load Balancer (Direct Routing Method)
http://blog.secaserver.com/2012/07/centos-configure-piranha-load-balancer-direct-routing-method/

• Centos – Linux Virtual Server Administration – Piranha
http://www.centos.org/docs/5/html/5.1/pdf/Virtual_Server_Administration.pdf

• Linux Virtual Server Tutorial – Direct Routing, Heartbeat, Round Robin
http://www.ultramonkey.org/papers/lvs_tutorial/html/

• Implementing Virtual Servers and Load Balancing Cluster System with Linux
http://www.firewall.cx/linux-knowledgebase-tutorials/system-and-network-services/855-linux-services-virtual-servers.html