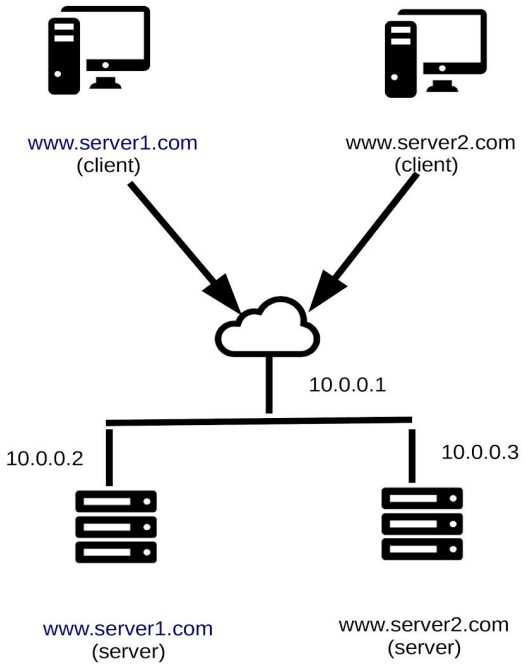
Multiple Servers Hosted On Private I.P Address And Accessed Through Single Public I.P. Address

-By Ravi Prakash

Introduction

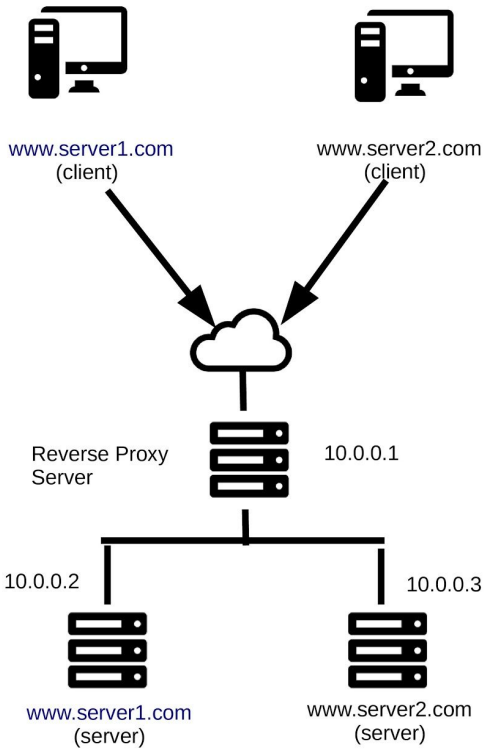
The topic is about hosting two servers e.g server1 and server2 on a single public I.P. address e.g. 10.0.0.1 . Let's assume that server1 and server2 are under same network system has private I.P. addresses 10.0.0.2 and 10.0.0.3 and domain names www.server1.com and www.server2.com respectively. So my next is when I search for www.server1.com and www.server2.com these two sites (i.e. HTTP or HTTPS request) should be accessed by the same I.P. Address i.e. 10.0.0.1 . It means that after resolving both names DNS should give same I.P. Address i.e. 10.0.0.1 . Now below is the diagram for above problem.



What will be our usual approach? NAT and PAT? But there are some drawbacks of that.You have to use ports(>1024) to forward traffic to specific server e.g. to access server1.com address will be 10.0.0.1:8080(Let's say) and same for server2.com address will be 10.0.0.1:8081. It means to access this server every time you have to use some non-standard ports which is quite hectic and makes url little bit messy like www.server1.com:8080 or www.server2.com:8081 .

Solution

Then, what will be the solution for the problem? Reverse Proxy. Yes, if we somehow construct a reverse proxy server using apache HTTP server which will handle the requests for server1 and server2 and can be accessed without PAT. Below is the diagram for the solution.



Now we can simply search for www.server1.com and www.server2.com without using any port and internally I.P. address or port no. can be changed which will not be reflected to the outside world and it will be safe and secure.

How To Configure The Whole System

We'll start with configuring a reverse proxy server to handle requests from clients.

Step 1: Create a ".conf" file in /etc/apache2/sites-available directory e.g. reverseProxy.conf on Reverse Proxy server.

Step 2: Write these configuration details into that.

```
1
2
3   <VirtualHost *:80>
4
5       ServerName www.server1.com
6       ServerAlias server1.com
7
8       ProxyPreserveHost On
9
10      ProxyPass / http://10.0.0.2/
11      ProxyPassReverse / http://10.0.0.2/
12
13
14
15      ErrorLog ${APACHE_LOG_DIR}/error.log
16      CustomLog ${APACHE_LOG_DIR}/access.log combined
17
18
19  </VirtualHost>
20
21  # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
22
23  <VirtualHost *:80>
24
25      ServerName www.server2.com
26      ServerAlias server2.com
27
28      ProxyPreserveHost On
29
30      ProxyPass / http://10.0.0.3/
31      ProxyPassReverse / http://10.0.0.3/
32
33      ErrorLog ${APACHE_LOG_DIR}/error.log
34      CustomLog ${APACHE_LOG_DIR}/access.log combined
35
36  </VirtualHost>
37
```

reverseProxy.conf

Above configuration is server name based classification.That means when traffic hits reverse proxy server it will be directed to required server.

Step 3: Create .conf file on each server e.g. server1.conf and server2.conf on server1 and server2 respectively.

```
<VirtualHost *:80>

    ServerName www.server1.com
    DocumentRoot /var/www/server1
    DirectoryIndex index.php

    <Directory />
        Options FollowSymLinks
        AllowOverride None
        Require all denied
    </Directory>
    <Directory /var/www/server1>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>



    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    |
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

Server1.conf

```
<VirtualHost *:80>

    ServerName www.server2.com
    DocumentRoot /var/www/server2
    DirectoryIndex index.php

    <Directory />
        Options FollowSymLinks
        AllowOverride None
        Require all denied
    </Directory>
    <Directory /var/www/server2>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>



    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined


</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

Server2.conf

Step 4: Now it's time for testing. Search for www.server1.com and www.server2.com from web browser.You'll find that your traffic will go through reverse proxy server whose IP address is 10.0.0.1 to 10.0.0.2 (for www.server1.com) or 10.0.0.3 (for www.server2.com) which is our destination I.P. address without any PAT and NAT.

That's all. Follow above steps and you are good to go.