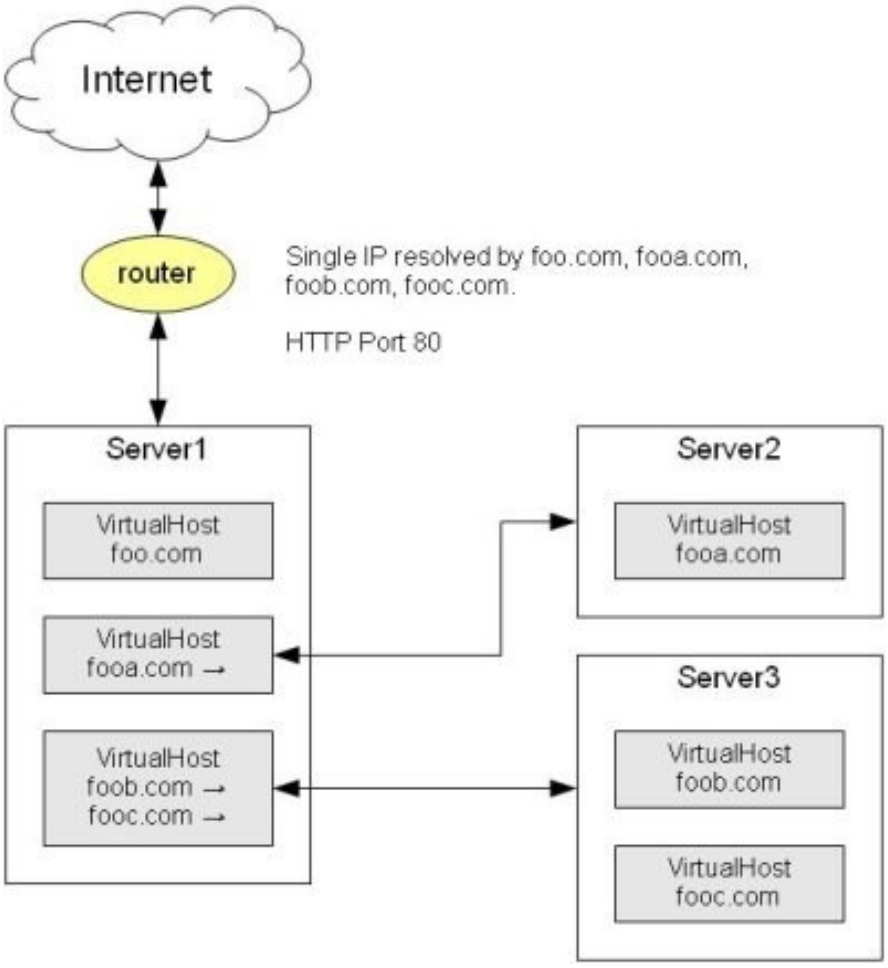
integratedwebsystems.com

http://www.integratedwebsystems.com/multiple-web-servers-over-a-single-ip-using-apache-as-a-reverse-proxy/

Multiple Web Servers over a Single IP, Using Apache as a Reverse Proxy

As a developer, I don't get to play with the IT side of things quite as much as I would like to. So I enjoy the little things, like learning how to use a reverse proxy on Apache. At home, I only have one IP coming in with my regular Internet connection, and I want the ability to run multiple servers inside my network on port 80. In the past, I've always just setup a new port and routed it to whichever server I wanted. 8080 here, 8081 there, etc. Well with Apache reverse proxy (mod_proxy), I found it was very easy to setup route requests to other internal servers.



This diagram shows the basic idea of what I'm doing. I have a "landing" server that takes all the traffic from the router. Using virtual hosts, I filter the web traffic by hostname and then provide mod_proxy directives to hand off the requests to other internal web servers. Options like *ProxyPreserveHost* allow you to hand off the original hostname in the request so you can further use hostname filtering on the secondary servers.

Here is an example of the landing server configuration.

```
NameVirtualHost *:80

<VirtualHost *:80>
  ServerName foo.com

  DocumentRoot /srv/www/default

  <Location "/">
    Order Deny, Allow
    Deny from all
    Allow from all
  </Location>
</VirtualHost>

<VirtualHost *:80>
  ServerName fooa.com

  ProxyPreserveHost on
  ProxyPass / http://server2/
  ProxyPassReverse / http://server2/
</VirtualHost>

<VirtualHost *:80>
  ServerName foob.com
  ServerAlias fooc.com

  ProxyPreserveHost on
  ProxyPass / http://server3/
  ProxyPassReverse / http://server3/
</VirtualHost>
```

You can see here, that the default site [foo.com] actually his hosted by the landing server. The rest of the hostnames are passed on to the other internal servers. You can also proxy just virtual directories as well. So if you want the requested path, “/something-foo” to go to another internal server, you can. You just use /something-foo as the first argument in ProxyPass and ProxyPassReverse.

The other web servers would have normal configurations like this one below. You can also proxy requests to other web servers like IIS. The proxy will handle basic authentication with an internal Windows IIS server.

```
NameVirtualHost *:80

<VirtualHost *:80>
  ServerName fooa.com

  DocumentRoot /srv/www/fooa.com

  <Location "/">
    Order Deny, Allow
    Deny from all
    Allow from all
  </Location>
</VirtualHost>
```

How To Use Apache HTTP Server As Reverse-Proxy Using mod_proxy Extension
https://www.digitalocean.com/community/tutorials/how-to-use-apache-http-server-as-reverse-proxy-using-mod_proxy-extension

Configure Apache as a Reverse Proxy Using mod_proxy on Ubuntu
https://devops.profitbricks.com/tutorials/configure-apache-as-a-reverse-proxy-using-mod_proxy-on-ubuntu/

Tutorial: Apache 2.4 as reverse proxy
https://www.leaseweb.com/labs/2014/12/tutorial-apache-2-4-transparent-reverse-proxy/

Installing Apache And mod_proxy

```
# -  - #
# - Update & install additional programs - #
sudo apt-get update;
sudo apt-get install libapache2-mod-proxy-html libxml2-dev;

# -  - #
# - Run the following command to get a list of available Apache modules - #
sudo a2enmod

# -  - #
# - Paste this list of modules - #
proxy proxy_ajp proxy_http rewrite deflate headers proxy_balancer proxy_connect proxy_html

# -  - #
# - All together in just one command - #
sudo a2enmod proxy proxy_ajp proxy_http rewrite deflate headers proxy_balancer proxy_connect
proxy_html
```

Modifying The Default Configuration

```
# -  - #
# - Edit 000-default.conf file - #
sudo vim /etc/apache2/sites-available/000-default.conf


# -  - #
# - Copy-and-paste the below block of configuration, amending it to suit your needs - #

<VirtualHost *:*>
    ProxyPreserveHost On

    # Servers to proxy the connection, or;
    # List of application servers:
    # Usage:
    # ProxyPass / http://[IP Addr.]:[port]/
    # ProxyPassReverse / http://[IP Addr.]:[port]/
    # Example:
    ProxyPass / http://0.0.0.0:8080/
    ProxyPassReverse / http://0.0.0.0:8080/

    ServerName localhost
</VirtualHost>
```

Enabling Load-Balancing

If you have multiple back-end servers, a good way to distribute the connection when proxying them is to use Apache's load balancing features.
Start editing the virtual-host settings like the previous step, but this time using the below configuration example:

```
<Proxy balancer://mycluster>
    # Define back-end servers:

    # Server 1
    BalancerMember http://0.0.0.0:8080/

    # Server 2
    BalancerMember http://0.0.0.0:8081/
</Proxy>

<VirtualHost *:*>
    # Apply VH settings as desired
    # However, configure ProxyPass argument to
    # use "mycluster" to balance the load

    ProxyPass / balancer://mycluster
</VirtualHost>
```

Enabling SSL Reverse-Proxy Support

If you are dealing with SSL connections and certificates, you will also need to enable a secondary virtual host with below settings.
Repeat the steps from the previous steps but using these configuration options:
Listen 443

```
NameVirtualHost *:443
<VirtualHost *:443>

    SSLEngine On

    # Set the path to SSL certificate
    # Usage: SSLCertificateFile /path/to/cert.pem
    SSLCertificateFile /etc/apache2/ssl/file.pem

    # Servers to proxy the connection, or;
    # List of application servers:
    # Usage:
    # ProxyPass / http://[IP Addr.]:[port]/
    # ProxyPassReverse / http://[IP Addr.]:[port]/
    # Example:
    ProxyPass / http://0.0.0.0:8080/
    ProxyPassReverse / http://0.0.0.0:8080/

    # Or, balance the load:
    # ProxyPass / balancer://balancer_cluster_name

</VirtualHost>



sudo service apache2 restart;
```

My sample with myaccounting.com.conf:

```
sudo vim /etc/apache2/sites-available/myaccounting.com.conf

# -  - #
# - The file will look something like this - #
<VirtualHost *:80>
        ServerAdmin admin@myaccounting.com
        ServerName myaccounting.com
        ServerAlias www.myaccounting.com
        DocumentRoot /home/AllDisks-Data/WebServer/www/WebSite/myaccounting.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>


# -  - #
# - Enable virtual-host files - #
sudo a2ensite 000-default.conf

sudo a2ensite myaccounting.com.conf
sudo a2ensite myinventory.com.conf
sudo a2ensite mysales.com.conf

# -  - #
# - Disable virtual-host files - #
sudo a2dissite myaccounting.com.conf
sudo a2dissite myinventory.com.conf
sudo a2dissite mysales.com.conf

sudo service apache2 restart;
```

 # - My sample with uServerOffice: - #
 # - Ubuntu Server in NJ Office: 72.76.115.67 - #

```
# -  - #
# - Put all the configuration in just the 000-default.conf file - #

# -  - #
# - Open file with vim - #
sudo vim /etc/apache2/sites-available/000-default.conf
```

 # - - #
 # - File content ofuServerOffice /etc/apache2/sites-available/000-default.conf - #

```
# - uServerOffice - #
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /home/AllDisks-Data/WebServer/www/WebSite
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

#       <Location "/">
#               Order Deny, Allow
#               Deny from all
#               Allow from all
#        </Location>
</VirtualHost>
```

```
<VirtualHost *:80>
        ServerAdmin admin@myaccounting.com
        ServerName myaccounting.com
        ServerAlias www.myaccounting.com
        DocumentRoot /home/AllDisks-Data/WebServer/www/WebSite/myaccounting.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

<VirtualHost *:80>
        ServerAdmin admin@myinventory.com
        ServerName myinventory.com
        ServerAlias www.myinventory.com
        DocumentRoot /home/AllDisks-Data/WebServer/www/WebSite/myinventory.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

<VirtualHost *:80>
        ServerAdmin admin@mysales.com
        ServerName mysales.com
        ServerAlias www.mysales.com
        DocumentRoot /home/AllDisks-Data/WebServer/www/WebSite/mysales.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

 # - - #
 # - File content continuation - #
 # - Accessing the other server in the office - #
 # - uServerOffice will redirect websites to uServer1710-VBox server on iMac - #

```
# - uServer1710-VBox - #
<VirtualHost *:80>
        ServerAdmin admin@userver-myaccounting.com
        ServerName userver-myaccounting.com
        ServerAlias www.userver-myaccounting.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        ProxyPreserveHost on
        ProxyPass / http://192.168.1.110/
        ProxyPassReverse / http://192.168.1.110/
</VirtualHost>

<VirtualHost *:80>
        ServerAdmin admin@userver-myinventory.com
        ServerName userver-myinventory.com
        ServerAlias www.userver-myinventory.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        ProxyPreserveHost on
        ProxyPass / http://192.168.1.110/
        ProxyPassReverse / http://192.168.1.110/
</VirtualHost>

<VirtualHost *:80>
        ServerAdmin admin@userver-mysales.com
        ServerName userver-mysales.com
        ServerAlias www.userver-mysales.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        ProxyPreserveHost on
        ProxyPass / http://192.168.1.110/
        ProxyPassReverse / http://192.168.1.110/
</VirtualHost>
```

 # - - #
 # - File content continuation - #
 # - Accessing the other server at home - #
 # - uServerOffice will redirect websites to uServerHome - #

```
# - uServerHome - #
<VirtualHost *:80>
        ServerAdmin admin@userverhome-myaccounting.com
        ServerName userverhome-myaccounting.com
        ServerAlias www.userverhome-myaccounting.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        ProxyPreserveHost on
        ProxyPass / http://69.123.114.8/
        ProxyPassReverse / http://69.123.114.8/
</VirtualHost>

<VirtualHost *:80>
        ServerAdmin admin@userverhome-myinventory.com
        ServerName userverhome-myinventory.com
        ServerAlias www.userverhome-myinventory.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        ProxyPreserveHost on
        ProxyPass / http://69.123.114.8/
        ProxyPassReverse / http://69.123.114.8/
</VirtualHost>

<VirtualHost *:80>
        ServerAdmin admin@userverhome-mysales.com
        ServerName userverhome-mysales.com
        ServerAlias www.userverhome-mysales.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        ProxyPreserveHost on
        ProxyPass / http://69.123.114.8/
        ProxyPassReverse / http://69.123.114.8/
</VirtualHost>
```

```
# -  - #
# - My sample with uServerOffice: - #
# - Configuration of the hosts file in the iMac computer - #

# -  - #
# - Edit the file with vim - #
sudo vim /etc/hosts
sudo gedit /etc/hosts


##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting.  Do not change this entry.
##
127.0.0.1       localhost
255.255.255.255 broadcasthost
::1             localhost



# -  - #
# - Public IP for uServerOffice in New Jersey - #
# - 72.76.115.67 - #

# -  - #
# - Public IP  WebSite directed to - #

# - uServerOffice in Ney Jersey - #
72.76.115.67 myaccounting.com
72.76.115.67 myinventory.com
72.76.115.67 mysales.com

72.76.115.67 www.myaccounting.com
72.76.115.67 www.myinventory.com
72.76.115.67 www.mysales.com


# - uServer1710-VBox on iMac in New Jersey - #
72.76.115.67 userver-myaccounting.com
72.76.115.67 userver-myinventory.com
72.76.115.67 userver-mysales.com

72.76.115.67 www.userver-myaccounting.com
72.76.115.67 www.userver-myinventory.com
72.76.115.67 www.userver-mysales.com


# - uServerHome in Connecticut - #
72.76.115.67 userverhome-myaccounting.com
72.76.115.67 userverhome-myinventory.com
72.76.115.67 userverhome-mysales.com

72.76.115.67 www.userverhome-myaccounting.com
72.76.115.67 www.userverhome-myinventory.com
72.76.115.67 www.userverhome-mysales.com
```

 # - - #
 # - Configuration of uServer1710-Vbox Office in NJ: - #

 # - 000-default.conf - #
sudo vim /etc/apache2/sites-available/000-default.conf

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /home/AllDisks-Data/WebServer/www/WebSite
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

 # - userver-myaccounting.com.conf - #
sudo vim /etc/apache2/sites-available/userver-myaccounting.com.conf

```
<VirtualHost *:80>
        ServerAdmin admin@userver-myaccounting.com
        DocumentRoot /home/AllDisks-Data/WebServer/www/WebSite/uServer-myaccounting.com
        ServerName userver-myaccounting.com
        ServerAlias www.userver-myaccounting.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

 # - userver-myinventory.com.conf - #
sudo vim /etc/apache2/sites-available/userver-myinventory.com.conf

```
<VirtualHost *:80>
        ServerAdmin admin@userver-myinventory.com
        DocumentRoot /home/AllDisks-Data/WebServer/www/WebSite/uServer-myinventory.com
        ServerName userver-myinventory.com
        ServerAlias www.userver-myinventory.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

 # - userver-mysales.com.conf - #
sudo vim /etc/apache2/sites-available/userver-mysales.com.conf

```
<VirtualHost *:80>
        ServerAdmin admin@userver-mysales.com
        DocumentRoot /home/AllDisks-Data/WebServer/www/WebSite/uServer-mysales.com
        ServerName userver-mysales.com
        ServerAlias www.userver-mysales.com
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

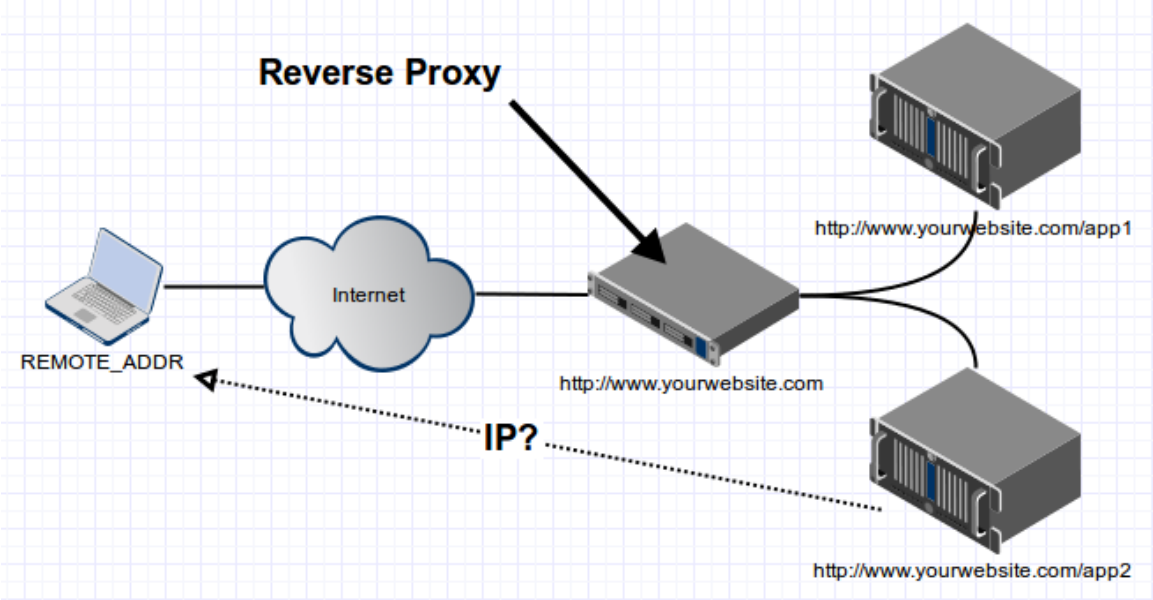
More to read:

http://www.microhowto.info/howto/configure_apache_as_a_reverse_proxy.html
http://www.microhowto.info/howto/redirect_http_requests_from_one_domain_to_another_using_apache.html
http://httpd.apache.org/docs/2.4/mod/mod_proxy.html

https://www.leaseweb.com/labs/2014/12/tutorial-apache-2-4-transparent-reverse-proxy/

Tutorial: Apache 2.4 as reverse proxy

This post explains how to configure Apache 2.4 (the version that comes with Ubuntu 14.04) as a fully transparent reverse proxy. If you have a single website that has multiple paths that are actually run by different web applications then this tutorial may be for you.



The proxy will serve both web applications from their own virtual host configuration. These may be on the same machine as shown below using the loop-back addresses 127.0.0.1 and 127.0.0.2 or on different machines if you use their (internal) IP addresses.

```
1   Site: http://www.yourwebsite.com/
2   App1: http://www.yourwebsite.com/app1 = http://127.0.0.1/app1
3   App2: http://www.yourwebsite.com/app2 = http://127.0.0.2/app2
```

This is the directory structure in which I want to load the various web apps:

```
1   maurits@nuc:/var/www/html$ ll
2   total 28
3   drwxr-xr-x 4 root root  4096 Dec  1 21:43 ./
4   drwxr-xr-x 3 root root  4096 Apr 21  2014 ../
5   -rw-r--r-- 1 root root 11510 Apr 21  2014 index.html
6   drwxr-xr-x 2 root root  4096 Dec  1 21:45 app1/
7   drwxr-xr-x 2 root root  4096 Dec  1 21:45 app2/
```

In this tutorial we run the web applications on the same paths as on the proxy. This means that the web apps run in a subdirectory, even on the machines behind the proxy. This avoids the need of rewriting and thus keeps this setup simple and easy to debug.

Setting up the reverse proxy in Apache 2.4

What we are going to do is setup a reverse proxy. First we load the “proxy_http” module in Apache 2.4 using:

```
1     sudo a2enmod proxy_http
2     sudo service apache2 restart
```

Let’s setup the reverse proxy virtual host configuration in “/etc/apache2/sites-available/yourwebsite-proxy.conf” like this:

```
1     <VirtualHost *:80>
2     ServerName www.yourwebsite.com
3     DocumentRoot /var/www/html
4     ProxyPreserveHost On
5     ProxyPass /app1 http://127.0.0.1/app1
6     ProxyPass /app2 http://127.0.0.2/app2
7     </VirtualHost>
```

The virtual host configuration of app1 in “/etc/apache2/sites-available/yourwebsite-app1.conf” looks like this:

```
1     <VirtualHost 127.0.0.1:80>
2     ServerName www.yourwebsite.com
3     DocumentRoot /var/www/html
4     ...
5     </VirtualHost>
```

And the virtual host configuration of app2 in “/etc/apache2/sites-available/yourwebsite-app2.conf” looks like this:

```
1     <VirtualHost 127.0.0.2:80>
2     ServerName www.yourwebsite.com
3     DocumentRoot /var/www/html
4     ...
5     </VirtualHost>
```

Lets enable all sites and reload Apache using:

```
1     sudo a2ensite yourwebsite-proxy yourwebsite-app1 yourwebsite-app2
2     sudo service apache2 reload
```

Note that this works as the virtual host configurations with a specified IP address will be matched first. The “ProxyPreserveHost” will make sure the “Host” header in the request is not rewritten. The lack of a “ProxyPassReverse” will make sure that there is no rewriting done on the response.

Showing the correct remote IP address

It is important to understand that in the above setup, the proxied web application will only see a different “REMOTE_ADDR” environment variable, since there is absolutely no rewriting going on.

The real visitor address is passed along in “X-Forwarded-For” header. This is a comma separated list and the last entry holds the real client IP address.

If you are on Apache 2.4, like in Ubuntu 14.04, you can correct the reported remote address by loading the “remoteip” module like this:

```
1       sudo a2enmod remoteip
2       sudo service apache2 restart
```

Add the “RemoteIPHeader” and “RemoteIPInternalProxy” directives to the virtual host configurations:

```
1       <VirtualHost 127.0.0.1:80>
2       ServerName www.yourwebsite.com
3       DocumentRoot /var/www/html
4       RemoteIPHeader X-Forwarded-For
5       RemoteIPInternalProxy 127.0.0.0/8
6       ...
7       </VirtualHost>
```

Note that the “RemoteIPInternalProxy” you must specify the internal IP address of the proxy. To test if you did it right you can run a PHP script that calls “phpinfo()”. If you see that the “REMOTE_ADDR” value is not set to the proxy, then it is working.

Adding headers to the upstream request

We want to make Apache2 add upstream headers and therefor we need to load the “headers” module in Apache 2.4 using:

```
1       sudo a2enmod headers
2       sudo service apache2 restart
```

Next, we have to adjust the reverse proxy virtual host configuration in “/etc/apache2/sites-available/yourwebsite-proxy.conf” like this:

```
1 <VirtualHost *:80>
2 ServerName www.yourwebsite.com
3 DocumentRoot /var/www/html
4 ProxyPreserveHost On
5 RewriteEngine On
6 RequestHeader add X-SSL off
7 RewriteRule ^/app1/(.*) http://127.0.0.1/app1/$1 [P,L]
8 RewriteRule ^/app2/(.*) http://127.0.0.2/app2/$1 [P,L]
9 </VirtualHost>
```

In this example we add a “X-SSL” header with the value “off” to the proxied request. If you want to add headers to the response you can use the “Header” directive.

If you have any questions, please use the comments below.