

Chapter 4 - Modules

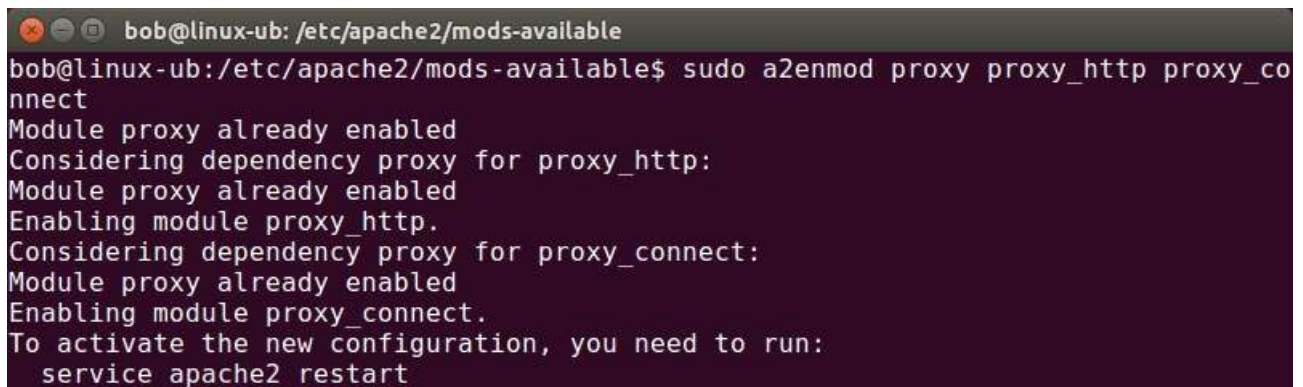
In this chapter we will describe some basic Apache modules. We will explain how you can configura Apache as a forward and reverse proxy.

Configure Apache as a forward proxy

Apache can be configured as both a forward and a reverse proxy. An ordinary proxy (also called a **forward proxy**) is an intermediate server that sits between the client and the origin server. The client is configured to use the forward proxy to access other sites. When a client want to get the content from the origin server, it sends a request to the proxy naming the origin server as the target. The proxy then requests the content from the origin server and returns it to the client.

Here is how we can configure Apache as a forward proxy:

First, we need to enable the **proxy**, **proxy_http**, and **proxy_connect** modules. We can do that using the **a2enmod** command:

A terminal window with a dark background and light text. The prompt is 'bob@linux-ub: /etc/apache2/mods-available'. The command entered is 'sudo a2enmod proxy proxy_http proxy_connect'. The output shows that the 'proxy' module is already enabled, and dependencies for 'proxy_http' and 'proxy_connect' are also handled. The final instruction is to run 'service apache2 restart' to activate the changes.

```
bob@linux-ub: /etc/apache2/mods-available
bob@linux-ub:/etc/apache2/mods-available$ sudo a2enmod proxy proxy_http proxy_co
nnect
Module proxy already enabled
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
Considering dependency proxy for proxy_connect:
Module proxy already enabled
Enabling module proxy_connect.
To activate the new configuration, you need to run:
  service apache2 restart
```

Next, go to the **/etc/apache2/mods-enabled** directory and open the file **proxy.conf** in a text editor of your choice. Uncomment the **#ProxyRequests On** line and the **<Proxy *>** block:

```

bob@linux-ub: /etc/apache2/mods-enabled
bob@linux-ub:/etc/apache2/mods-enabled$ cat proxy.conf
<IfModule mod_proxy.c>

    # If you want to use apache2 as a forward proxy, uncomment the
    # 'ProxyRequests On' line and the <Proxy *> block below.
    # WARNING: Be careful to restrict access inside the <Proxy *> block.
    # Open proxy servers are dangerous both to your network and to the
    # Internet at large.
    #
    # If you only want to use apache2 as a reverse proxy/gateway in
    # front of some web application server, you DON'T need
    # 'ProxyRequests On'.

    ProxyRequests On
    <Proxy *>
        AddDefaultCharset off
        Require all denied
        Require local
    </Proxy>

    # Enable/disable the handling of HTTP/1.1 "Via:" headers.
    # ("Full" adds the server version; "Block" removes all outgoing Via: hea
ders)
    # Set to one of: Off | On | Full | Block

```

Now, create a new file in the **/etc/apache2/sites-available** directory. We will call our file **forward_proxy.conf**. This is the configuration of the file:

```

<VirtualHost *:8080>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ProxyRequests On
    ProxyVia On

    <Proxy "">
        Require ip 192.168
    </Proxy>

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error_forward_proxy.log

```

CustomLog \${APACHE_LOG_DIR}/access_forward_proxy.log combined

```
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
```

</VirtualHost>

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

Here is a description of the lines in the file:

<VirtualHost *:8080> - specifies the port that will be used for this virtual host.

ProxyRequests On, ProxyVia On - enables the proxy.

<Proxy "">Require ip 192.168</Proxy> - determines the range of IP addresses that will be allowed to use the proxy. In our case, the range of allowed hosts is **192.168.0.0 - 192.168.255.255**.

ErrorLog \${APACHE_LOG_DIR}/error_forward_proxy.log, CustomLog \${APACHE_LOG_DIR}/access_forward_proxy.log combined - specifies the log files location.

Next, open the **/etc/apache2/ports.conf** file and add the **Listen 8080** line:

```
bob@linux-ub:/etc/apache2$ ls
apache2.conf  envvars          mods-enabled  sites-available
conf-available  magic           ports.conf   sites-enabled
conf-enabled   mods-available  proxy.conf
bob@linux-ub:/etc/apache2$ cat ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
Listen 8080

<IfModule ssl_module>
    Listen 443
</IfModule>

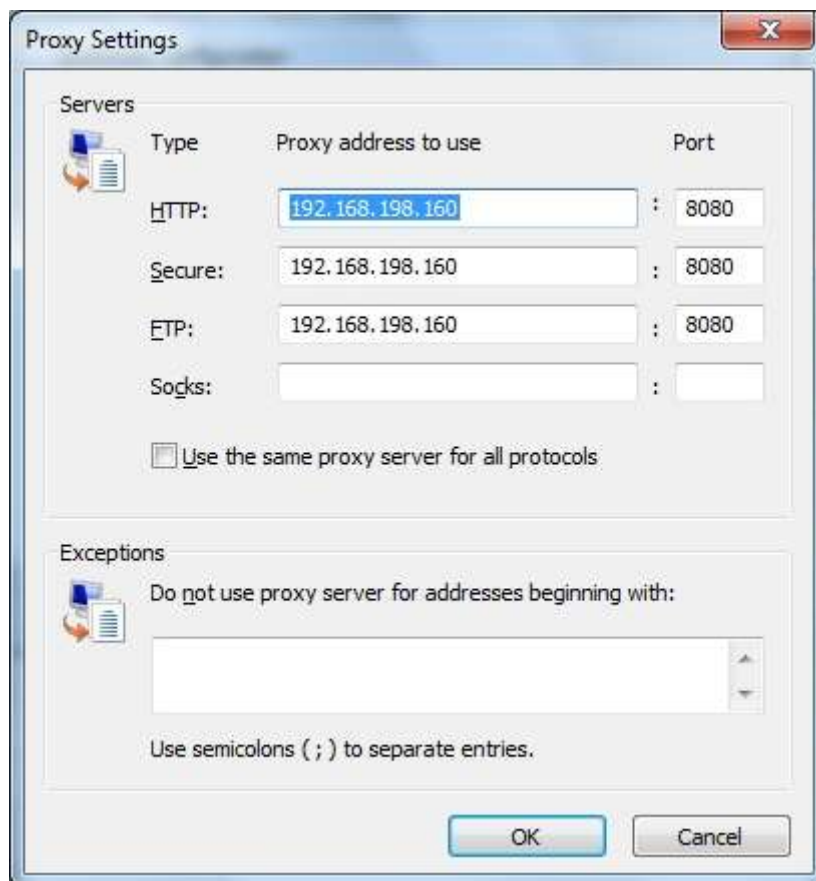
<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
bob@linux-ub:/etc/apache2$
```

Enable the site using the **a2ensite** command:

```
bob@linux-ub: /etc/apache2
bob@linux-ub:/etc/apache2$ sudo a2ensite forward_proxy.conf
Enabling site forward_proxy.
To activate the new configuration, you need to run:
service apache2 reload
```

Restart Apache in order for the changes to take effect. Your web clients need to be configured to use the proxy for outside connections. Here is a proxy configuration window from Windows:



Configure Apache as a reverse proxy

Apache can also be configured to serve as a **reverse proxy**. A reverse proxy appears to the client just like an ordinary web server and no special configuration on the client is necessary. The client makes ordinary requests for content. The reverse proxy then decides where to send those requests and returns the content as if it were itself the origin. Reverse proxies are usually used to provide Internet users access to a server that is behind a firewall or to balance load among several back-end servers.

Here is how we can configure Apache as a reverse proxy:

First, we need to enable the **proxy**, **proxy_http**, and **proxy_connect** modules. We can do

that using the **a2enmod** command:

```
bob@linux-ub: /etc/apache2/mods-available
bob@linux-ub:/etc/apache2/mods-available$ sudo a2enmod proxy proxy_http proxy_connect
Module proxy already enabled
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
Considering dependency proxy for proxy_connect:
Module proxy already enabled
Enabling module proxy_connect.
To activate the new configuration, you need to run:
  service apache2 restart
```

Next, go to the **/etc/apache2/mods-enabled** directory and open the file **proxy.conf** in a text editor of your choice. Uncomment the **ProxyRequests On** line, the **<Proxy *>** block, and the **ProxyVia Off** line. Change the **ProxyRequests** to **Off** and **ProxyVia** to **On**:

```
bob@linux-ub: /etc/apache2/mods-available
GNU nano 2.2.6          File: proxy.conf          Modified

# 'ProxyRequests On'.

ProxyRequests Off
<Proxy *>
  AddDefaultCharset off
  Require all denied
  Require local
</Proxy>

# Enable/disable the handling of HTTP/1.1 "Via:" headers.
# ("Full" adds the server version; "Block" removes all outgoing Via: headers)
# Set to one of: Off | On | Full | Block
ProxyVia On

</IfModule>

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

^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Now, create a new file in the **/etc/apache2/sites-available** directory. We will call our file **reverse_proxy.conf**. This is the configuration of the file:

```
<VirtualHost *:80>
  ServerName msn.local

  ProxyPass / http://www.msn.com

  <Proxy "">
```

```
Require ip 192.168  
</Proxy>
```

```
</VirtualHost>
```

Enable the website using the **sudo a2ensite reverse_proxy.conf** command and restart Apache. When the internal client requests the website **msn.local**, he or she will be redirected to **www.msn.com**, as specified by the **ProxyPass** directive.

Display server statistics

You can use the Apache's **mod_status** module to display a web page containing statistics about the web server's current state. Some of the information included in the report are:

- active connections.
- the number of worker serving requests.
- the number of idle worker.
- the status of each worker, the number of requests that worker has performed and the total number of bytes served by the worker.
- the total number of accesses and byte count served.
- the time the server was started/restarted and the time it has been running for.

The **mod_status** module is usually enabled by default. If not, enable it using the **sudo a2enmod status** command.

To enable access to the server status page, you need to add a **Location** directive entry within the **VirtualHost** section in the **/etc/apache2/sites-available/000-default.conf** file:

```
bob@linux-ub: /etc/apache2/sites-available
GNU nano 2.2.6 File: 000-default.conf Modified

<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

<Location "/server-status">
SetHandler server-status
Require ip 192.168
</Location>

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

The **Location** directive listed above specifies that the server statistics page will be displayed when you browse to the **/server-status** URL. The **Require** directive specifies the hosts that will be allowed to access the webpage (in this case, all hosts from the **192.168.0.0 - 192.168.255.255** range).

We can get the information by browsing to **http://URL/server-status**:

Apache Status - Windows Internet Explorer
 http://192.168.198.160/server-status

Apache Status

Apache Server Status for 192.168.198.160 (via 192.168.198.160)

Server Version: Apache/2.4.7 (Ubuntu) OpenSSL/1.0.1f
 Server MPM: event
 Server Built: Jan 14 2016 17:45:23

Current Time: Friday, 01-Apr-2016 17:35:53 CEST
 Restart Time: Friday, 01-Apr-2016 17:35:47 CEST
 Parent Server Config. Generation: 1
 Parent Server MPM Generation: 0
 Server uptime: 6 seconds
 Server load: 0.14 0.15 0.11
 Total accesses: 0 - Total Traffic: 0 kB
 CPU Usage: u0 s0 cu0 cs0
 0 requests/sec - 0 B/second -
 1 requests currently being processed, 49 idle workers

PID	Connections		Threads		Async connections		
	total	accepting	busy	idle	writing	keep-alive	closing
13810	0	yes	1	24	0	0	0
13811	0	yes	0	25	0	0	0
Sum	0		1	49	0	0	0