We will be running two separate servers server1 and server2. Server1 is running Debian Linux with Nginx reverse proxy and an IP address 10.1.1.251. Server2 run basic website using Apache2 webserver on IP address 10.1.1.252. We assume that Server2 is up and running to serve a content on port 80:

$ lynx -dump http://10.1.1.252

Hello this is apache2 sitting on host 10.1.1.252

As a result our main focus will be dedicated to Server1 and configuration of Nginx reverse proxy to act as a intermediary between client and Server2.

**Minimal Reverse proxy configuration**

This section will show a minimalistic reverse proxy configuration which can be later improved to suit your environment. Let's start by nginx installation:

# apt-get install nginx

Next, we disable a default virtual host:

# unlink /etc/nginx/sites-enabled/default

Next, create a new file within /etc/nginx/sites-available directory to hold your reverse proxy configuration eg reverse-proxy with a following content:

server {

listen 80;

location / {

proxy\_pass http://10.1.1.252;

}

}

The above reverse proxy configuration is absolutely minimalistic and self explanatory. The main feature is the proxy\_pass directive which instructs nginx to proxy all requests communing on socket 10.1.1.251:80 to remote socket 10.1.1.252:80.Make sure that your nginx configuration does not contain any errors and restart nginx.

# nginx -t

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

nginx: configuration file /etc/nginx/nginx.conf test is successful

# service nginx restart

At this stage you should be able to query your new reverse proxy on ip address 10.1.1.251 and reach a content of apache2 running on ip address 10.1.1.251:

$ lynx -dump http://10.1.1.251

Hello this is apache2 sitting on host 10.1.1.252

Once the above basic reverse proxy configuration is working nginx contains number of additional [directives](http://nginx.org/en/docs/http/ngx_http_proxy_module.html) to improve your configuration.