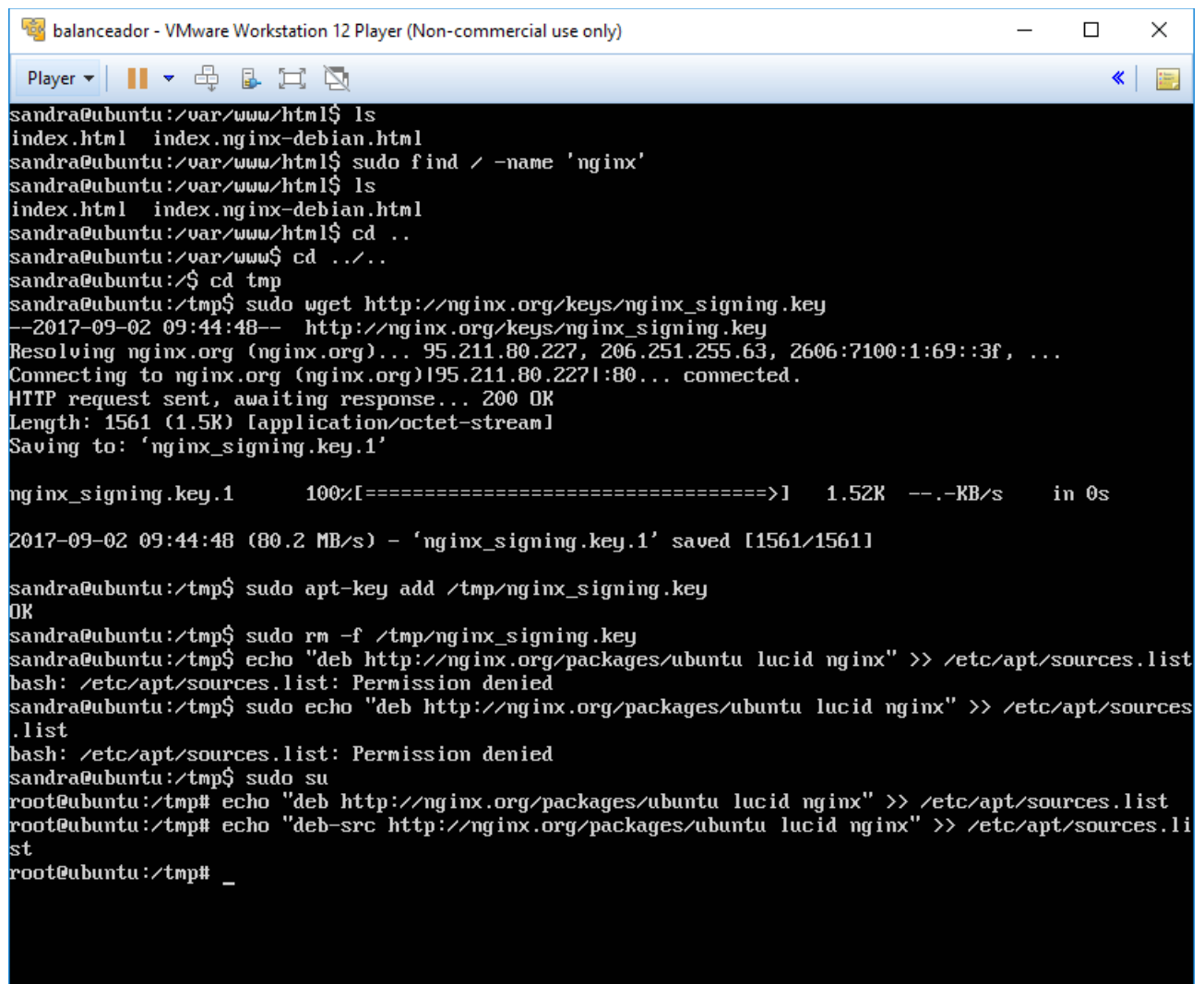


## Práctica 3

### NGINX

En primer lugar importamos la clave del repositorio:



```
balanceador - VMware Workstation 12 Player (Non-commercial use only)
Player
sandra@ubuntu:/var/www/html$ ls
index.html  index.nginx-debian.html
sandra@ubuntu:/var/www/html$ sudo find / -name 'nginx'
sandra@ubuntu:/var/www/html$ ls
index.html  index.nginx-debian.html
sandra@ubuntu:/var/www/html$ cd ..
sandra@ubuntu:/var/www$ cd ../../
sandra@ubuntu:/$ cd tmp
sandra@ubuntu:/tmp$ sudo wget http://nginx.org/keys/nginx_signing.key
--2017-09-02 09:44:48--  http://nginx.org/keys/nginx_signing.key
Resolving nginx.org (nginx.org)... 95.211.80.227, 206.251.255.63, 2606:7100:1:69::3f, ...
Connecting to nginx.org (nginx.org)|95.211.80.227|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1561 (1.5K) [application/octet-stream]
Saving to: 'nginx_signing.key.1'

nginx_signing.key.1      100%[=====>]  1.52K  --.-KB/s    in 0s
2017-09-02 09:44:48 (80.2 MB/s) - 'nginx_signing.key.1' saved [1561/1561]

sandra@ubuntu:/tmp$ sudo apt-key add /tmp/nginx_signing.key
OK
sandra@ubuntu:/tmp$ sudo rm -f /tmp/nginx_signing.key
sandra@ubuntu:/tmp$ echo "deb http://nginx.org/packages/ubuntu lucid nginx" >> /etc/apt/sources.list
bash: /etc/apt/sources.list: Permission denied
sandra@ubuntu:/tmp$ sudo echo "deb http://nginx.org/packages/ubuntu lucid nginx" >> /etc/apt/sources
.list
bash: /etc/apt/sources.list: Permission denied
sandra@ubuntu:/tmp$ sudo su
root@ubuntu:/tmp# echo "deb http://nginx.org/packages/ubuntu lucid nginx" >> /etc/apt/sources.list
root@ubuntu:/tmp# echo "deb-src http://nginx.org/packages/ubuntu lucid nginx" >> /etc/apt/sources.li
st
root@ubuntu:/tmp# _
```

instalamos nginx en el balanceador con:

```
sudo apt-get update && sudo apt-get dist-upgrade && sudo apt-get autoremove
```

```
sudo apt-get install nginx
```

```
sudo systemctl start nginx
```

y configuramos nginx para que redirija la carga a dos servidores (máquina 1 y máquina 2).

```
balanceador - VMware Workstation 12 Player (Non-commercial use only)
Player
root@ubuntu:/etc/nginx/sites-enabled# cd ..
root@ubuntu:/etc/nginx# cd conf.d/
root@ubuntu:/etc/nginx/conf.d# cat default.conf
upstream apaches{
    server 192.168.109.131;
    server 192.168.109.132;
}
server {
    listen 80;
    server_name balanceador;

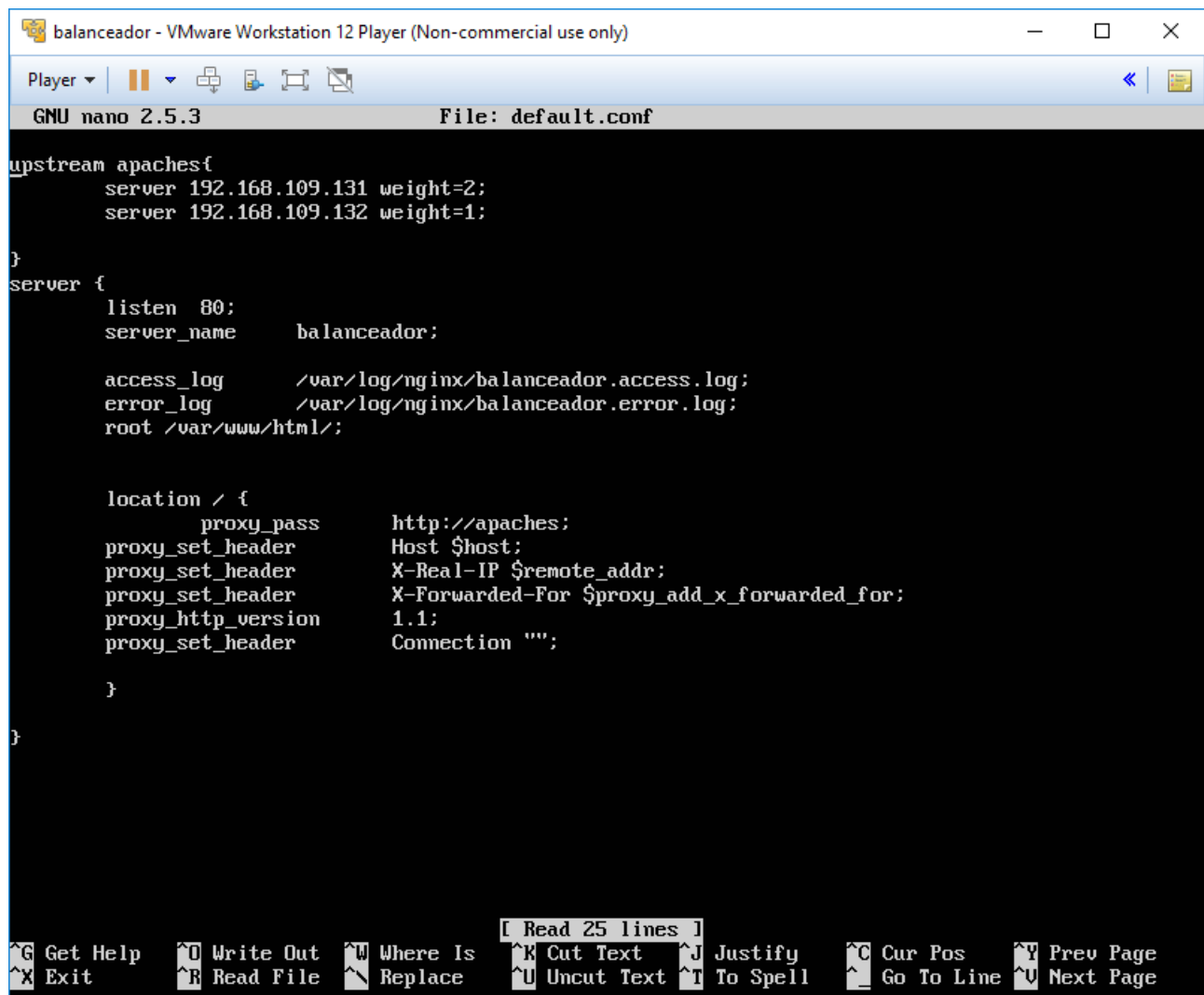
    access_log /var/log/nginx/balanceador.access.log;
    error_log /var/log/nginx/balanceador.error.log;
    root /var/www/html/;

    location / {
        proxy_pass http://apaches;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
root@ubuntu:/etc/nginx/conf.d#
```

Ahora reiniciamos el servicio y comprobamos que funciona el balanceo:

```
balanceador - VMware Workstation 12 Player (Non-commercial use only)
Player
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 2)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 1)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 2)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 1)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 2)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 1)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# _
```

Ahora le asignamos un peso a cada máquina (peso 2 a la máquina 1 y peso 1 a la máquina 2):



```
balanceador - VMware Workstation 12 Player (Non-commercial use only)
Player
GNU nano 2.5.3 File: default.conf

upstream apaches{
    server 192.168.109.131 weight=2;
    server 192.168.109.132 weight=1;
}

server {
    listen 80;
    server_name balanceador;

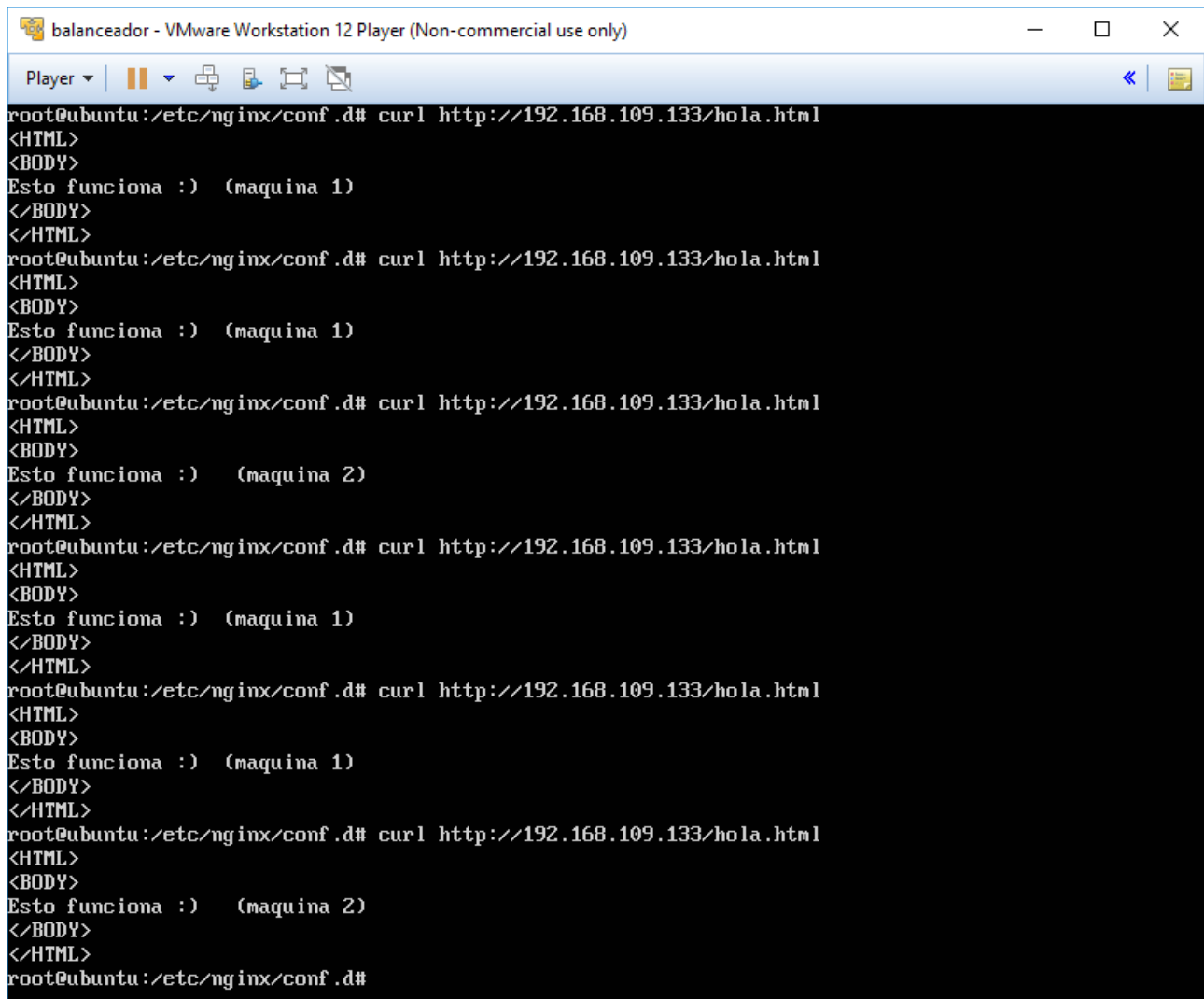
    access_log /var/log/nginx/balanceador.access.log;
    error_log /var/log/nginx/balanceador.error.log;
    root /var/www/html/;

    location / {
        proxy_pass http://apaches;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

[ Read 25 lines ]

Get Help Write Out Where Is Cut Text Justify Cur Pos Prev Page  
Exit Read File Replace Uncut Text To Spell Go To Line Next Page

reiniciamos el servicio y comprobamos que se han modificado correctamente las cargas de cada máquina:



```
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 1)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 1)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 2)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 1)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 1)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 2)
</BODY>
</HTML>
root@ubuntu:/etc/nginx/conf.d#
```

## HAPROXY

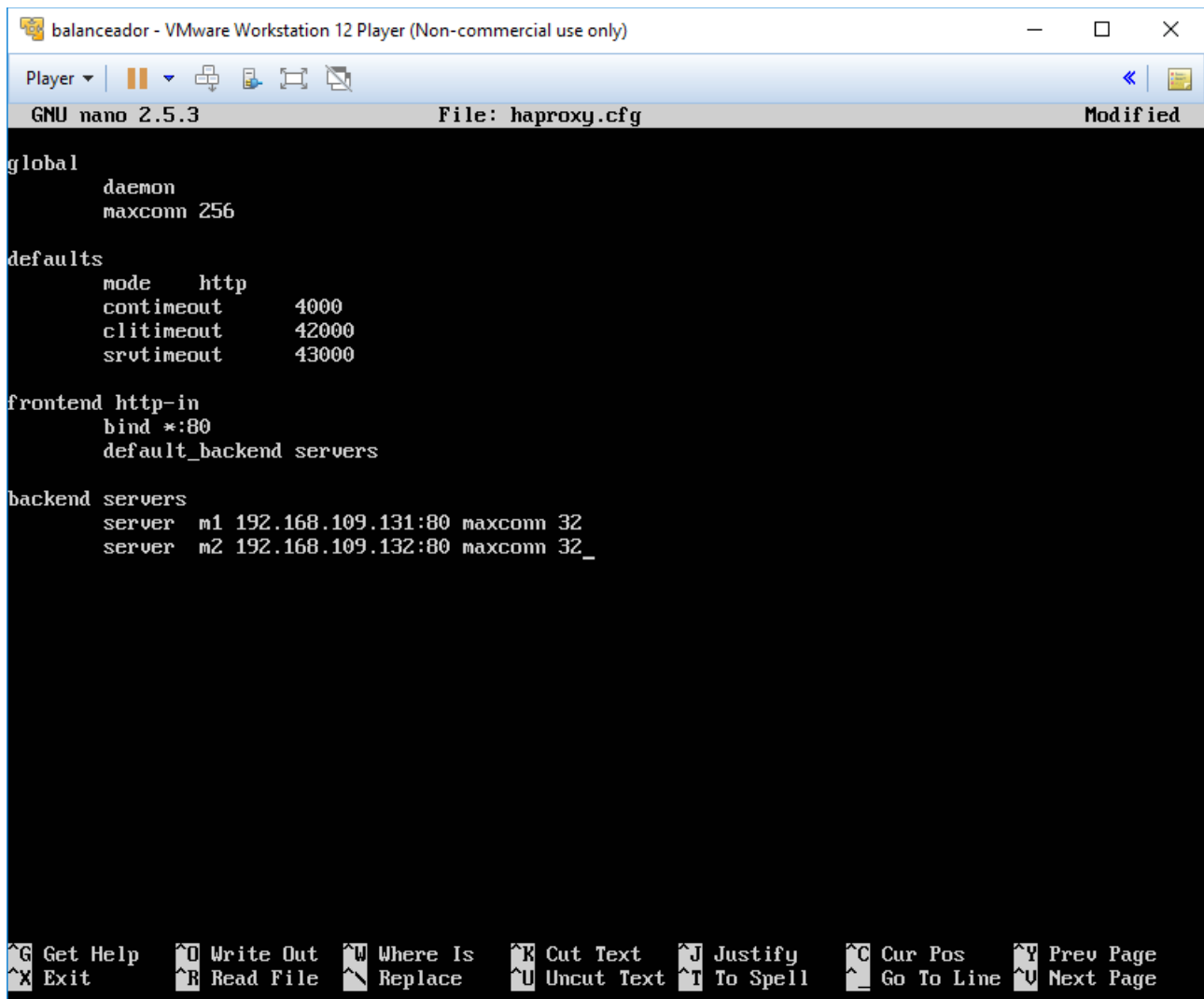
En primer lugar detenemos nginx con:

```
sudo service nginx stop
```

Y procedemos a instalar haproxy con:

```
sudo apt-get install haproxy
```

Ahora modificamos el archivo de configuración de haproxy:



The screenshot shows a VMware Workstation 12 Player window titled "balanceador - VMware Workstation 12 Player (Non-commercial use only)". The window contains a terminal window running the GNU nano 2.5.3 text editor, editing the file "haproxy.cfg". The editor's status bar at the top indicates "File: haproxy.cfg" and "Modified". The content of the file is as follows:

```
global
    daemon
    maxconn 256

defaults
    mode http
    timeout 4000
    clitimeout 42000
    srvttimeout 43000

frontend http-in
    bind *:80
    default_backend servers

backend servers
    server m1 192.168.109.131:80 maxconn 32
    server m2 192.168.109.132:80 maxconn 32_
```

The bottom of the terminal window displays a series of keyboard shortcuts for nano editor functions:

Get Help	Write Out	Where Is	Cut Text	Justify	Cur Pos	Prev Page
Exit	Read File	Replace	Uncut Text	To Spell	Go To Line	Next Page

Comprobamos que el servicio funciona y hacemos varias peticiones para ver que la carga se reparte:

```
root@ubuntu:/etc/haproxy# sudo /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg
root@ubuntu:/etc/haproxy# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 1)
</BODY>
</HTML>
root@ubuntu:/etc/haproxy# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 1)
</BODY>
</HTML>
root@ubuntu:/etc/haproxy# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 2)
</BODY>
</HTML>
root@ubuntu:/etc/haproxy# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 2)
</BODY>
</HTML>
root@ubuntu:/etc/haproxy# curl http://192.168.109.133/hola.html
<HTML>
<BODY>
Esto funciona :) (maquina 1)
</BODY>
</HTML>
root@ubuntu:/etc/haproxy#
```

## Alta carga con nginx

Instalamos apache benchmark con:

```
sudo apt-get install apache2-utils
```

Para probar la granja web ejecutamos:

```
ab -n 10000 -c 100 http://192.168.109.133/hola.html
```

```
balanceador - VMware Workstation 12 Player (Non-commercial use only)
Player
Server Software:      nginx/1.10.3
Server Hostname:      192.168.109.133
Server Port:          80

Document Path:        /hola.html
Document Length:      60 bytes

Concurrency Level:    100
Time taken for tests:  3.475 seconds
Complete requests:    10000
Failed requests:      3333
  (Connect: 0, Receive: 0, Length: 3333, Exceptions: 0)
Total transferred:    3053333 bytes
HTML transferred:     603333 bytes
Requests per second:  2877.57 [#/sec] (mean)
Time per request:     34.752 [ms] (mean)
Time per request:     0.348 [ms] (mean, across all concurrent requests)
Transfer rate:        858.03 [Kbytes/sec] received

Connection Times (ms)
      min    mean[+/-sdl]  median    max
Connect:    0      0   0.5      0      6
Processing:  3     34   2.4     35     41
Waiting:    1     34   2.4     35     41
Total:       7     35   2.3     35     41

Percentage of the requests served within a certain time (ms)
 50%    35
 66%    36
 75%    36
 80%    36
 90%    37
 95%    38
 98%    39
 99%    39
100%    41 (longest request)
sandra@ubuntu:/$ _
```

Como podemos ver ha tardado 3.475 segundos en ejecutar 10000 peticiones de 100 en 100.

## Alta carga con haproxy

Ahora detenemos nginx, iniciamos haproxy y volvemos a ejecutar:

```
ab -n 10000 -c 100 http://192.168.109.133/hola.html
```

```
balanceador - VMware Workstation 12 Player (Non-commercial use only)
Player
Server Software:      Apache/2.4.18
Server Hostname:      192.168.109.133
Server Port:          80

Document Path:        /hola.html
Document Length:       60 bytes

Concurrency Level:     100
Time taken for tests:   2.250 seconds
Complete requests:      10000
Failed requests:        5000
    (Connect: 0, Receive: 0, Length: 5000, Exceptions: 0)
Total transferred:      3065000 bytes
HTML transferred:       605000 bytes
Requests per second:    4444.76 [#/sec] (mean)
Time per request:       22.498 [ms] (mean)
Time per request:       0.225 [ms] (mean, across all concurrent requests)
Transfer rate:          1330.39 [Kbytes/sec] received

Connection Times (ms)
      min    mean[+/-sd] median    max
Connect:    0      0   0.5      0      6
Processing:  2     22   3.9     23     41
Waiting:    1     22   3.9     23     41
Total:       7     22   3.9     23     41

Percentage of the requests served within a certain time (ms)
 50%    23
 66%    24
 75%    24
 80%    25
 90%    28
 95%    29
 98%    33
 99%    35
100%    41 (longest request)
sandra@ubuntu:~$ _
```

Como podemos ver ha tardado 2,250 segundos en ejecutar 10000 peticiones de 100 en 100.

En comparación haproxy da mejores resultados que nginx, ya que el tiempo total de respuesta es mejor, pero también lo son el número de peticiones por segundo y el tiempo por petición.