

# 1. Configurar Apache como Reverse Proxy

## Configuración del servidor web s\_backend-1

Personalizar página index.html asociada al sitio web. Verificar acceso al sitio web

Hemos modificado el fichero /var/www/html/index.html para añadir un html de prueba.

Tras hacer curl en la propia máquina backend, nos devuelve el html correctamente:

```
root@s_backend-1:~# curl http://backend-1.srdsi.lab
<!DOCTYPE html>
<HTML>
<HEAD>
  <TITLE> BACKEND ONE </TITLE>
</HEAD>
<BODY>
  <H1> ***** BACKEND ONE ***** </H1>
</BODY>
</HTML>root@s_backend-1:~#
```

## Configuración del proxy proxyweb

### 4. Habilitar los módulos necesarios

```
root@proxyweb:~# a2enmod proxy proxy_http
Enabling module proxy.
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
To activate the new configuration, you need to run:
  service apache2 restart
```

## 5. Poner en marcha el sitio virtual (frontend.srdsilab) que actuará como reverse proxy [ver anexo]

Hemos añadido:

```
<VirtualHost *:80>
    ServerName frontend.srdsilab
    DocumentRoot /var/www/frontend
    ErrorLog ${APACHE_LOG_DIR}/frontend-error.log
    CustomLog ${APACHE_LOG_DIR}/frontend-access.log combined
</VirtualHost>
```

En /etc/apache2/sites-available/frontend.conf de la máquina proxy.

También hemos creado un index.html en /var/www/frontend:

```
<!DOCTYPE html>
<html>
<head>
    <title>Frontend Proxy</title>
</head>
<body>
    <h1>Frontend Proxy Server</h1>
    <p>Esta página será reemplazada por el contenido del backend</p>
</body>
</html>
```

## 6. Habilitar la funcionalidad de proxy en el sitio virtual web creado añadiendo

Hemos añadido lo indicado en el enunciado al fichero

/etc/apache2/sites-available/frontend.conf de la máquina proxy.

```
<VirtualHost *:80>
    ServerName frontend.srdsilab
    DocumentRoot /var/www/frontend

    # evita que el front-end sea utilizado como proxy forward
    ProxyRequests Off

    # el paso del proxyweb al servidor interno (s_backend-1) sea transparente p
ra el usuario
    ProxyPreserveHost Off

    # gestionan el salto y la vuelta del proxyweb al s_backend-1
    ProxyPass / http://backend-1.srdsilab/
    ProxyPassReverse / http://backend-1.srdsilab/

    ErrorLog ${APACHE_LOG_DIR}/frontend-error.log
    CustomLog ${APACHE_LOG_DIR}/frontend-access.log combined
</VirtualHost>
```

## 7. Habilitar el sitio virtual y reiniciar el servidor Apache

```
root@proxyweb:~# a2ensite frontend.conf
Enabling site frontend.
To activate the new configuration, you need to run:
  service apache2 reload
root@proxyweb:~# service apache2 reload
[ ok ] Reloading web server: apache2.
root@proxyweb:~#
```

## 8. Comprobar accesos desde un navegador

Hemos añadido en /etc/hosts del proxy:

```
127.0.0.1 frontend.srdsilab
```

a) En el servidor proxyweb (navegador lynx )

Al hacer lynx <http://frontend.srdsilab> desde el propio proxy:

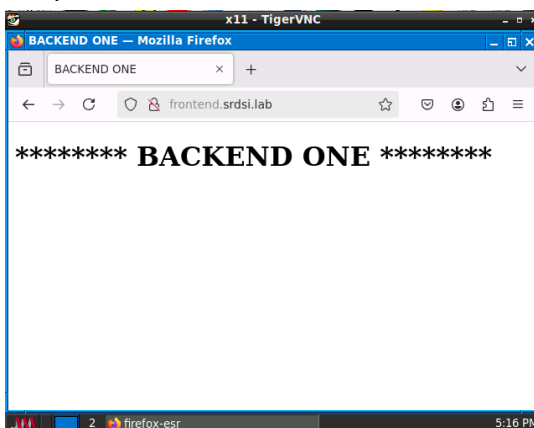
```
CTYPE html>
          Esta es la página personalizada de BACKEND-1

Si puedes ver este texto, la configuración está funcionando
correctamente.
```

Como se puede observar, nos devuelve el contenido del backend en lugar del index.html del frontend, ya que hemos indicado las directivas ProxyPass.

b) En los clientes. ¿Qué máquina responde a los clientes?

Responde el backend1



¿no responde el proxy?

9. Cuando Apache actúa como reverse proxy, la directiva ProxyPass añade una serie de cabeceras para pasar información adicional al servidor web:

*X-Forwarded-For* dirección IP del cliente

*X-Forwarded-Host* URL original solicitada por el cliente

*X-Forwarded-Server* nombre del servidor proxy

a) Realizar una captura con wireshark, y comprobar que información pasa el servidor proxy al servidor backend en estas cabeceras cuando le reenvía la solicitud de un cliente.



Podemos ver que aparece la IP del cliente y que solicita la consulta al servidor frontend.srdsi.lab

b) ¿Y si cambiamos ProxyPreserveHost On ? ¿Cambia alguna cabecera? ¿Qué ocurre?

Editamos el fichero /etc/apache2/sites-available/frontend.conf en la máquina proxy, cambiando Off por On en la línea ProxyPreserveHost

```
user1@demo: ~/GNS3/projects x s_backend-1 x proxyweb
GNU nano 2.2.6 File: frontend.conf Modified
<VirtualHost *:80 >
  ServerName frontend.srdsilab
  DocumentRoot /var/www/frontend

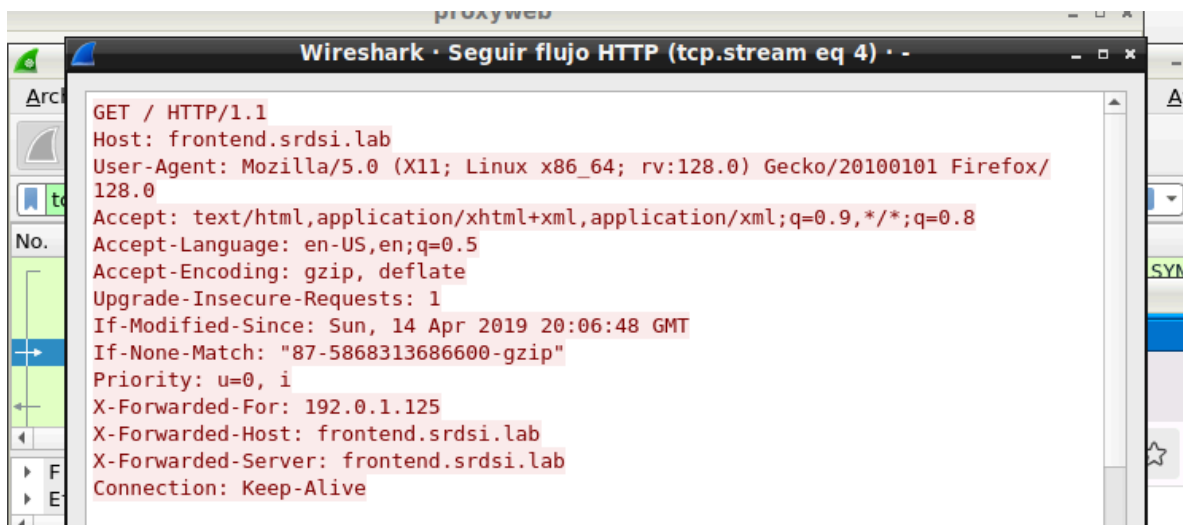
  ErrorLog "/var/log/apache2/sitioweb-error.log"
  CustomLog "/var/log/apache2/sitioweb-access.log" common

  ProxyRequests Off
  ProxyPreserveHost On

  ProxyPass / http://backend-1.srdsilab/
  ProxyPassReverse / http://backend-1.srdsilab/
</VirtualHost>
```

Recargamos la configuración:

```
root@proxyweb:/etc/apache2/sites-available# a2ensite frontend.conf
Site frontend already enabled
root@proxyweb:/etc/apache2/sites-available# service apache2 reload
[ ok ] Reloading web server: apache2.
root@proxyweb:/etc/apache2/sites-available#
```



En este caso la petición devolverá lo mismo, pero al servidor backend se le envía en la cabecera Host, el nombre del host por el que se ha hecho la petición (nombre del proxy)

## 10. Habilitar la conexión segura mediante TLS al reverse proxy.

a) Habilitar el módulo en Apache para TLS

```
user1@demo: ~/GNS3/projects x s_backend-1 x proxyweb x
root@proxyweb:/etc/apache2/sites-available# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
elf-signed certificates.
To activate the new configuration, you need to run:
service apache2 restart
root@proxyweb:/etc/apache2/sites-available# service apache2 restart
[ ok ] Restarting web server: apache2.
root@proxyweb:/etc/apache2/sites-available#
```

b) Modificar el sitio web

```
user1@demo: ~/GNS3/projects x s_backend-1 x proxyweb x
GNU nano 2.2.6 File: frontend.conf Modified
<VirtualHost *:80>
    ServerName frontend.srdsilab
    Redirect / https://frontend.srdsilab/
</VirtualHost>

<VirtualHost *:443 >
    ServerName frontend.srdsilab
    DocumentRoot /var/www/frontend

    ErrorLog "/var/log/apache2/sitioweb-error.log"
    CustomLog "/var/log/apache2/sitioweb-access.log" common

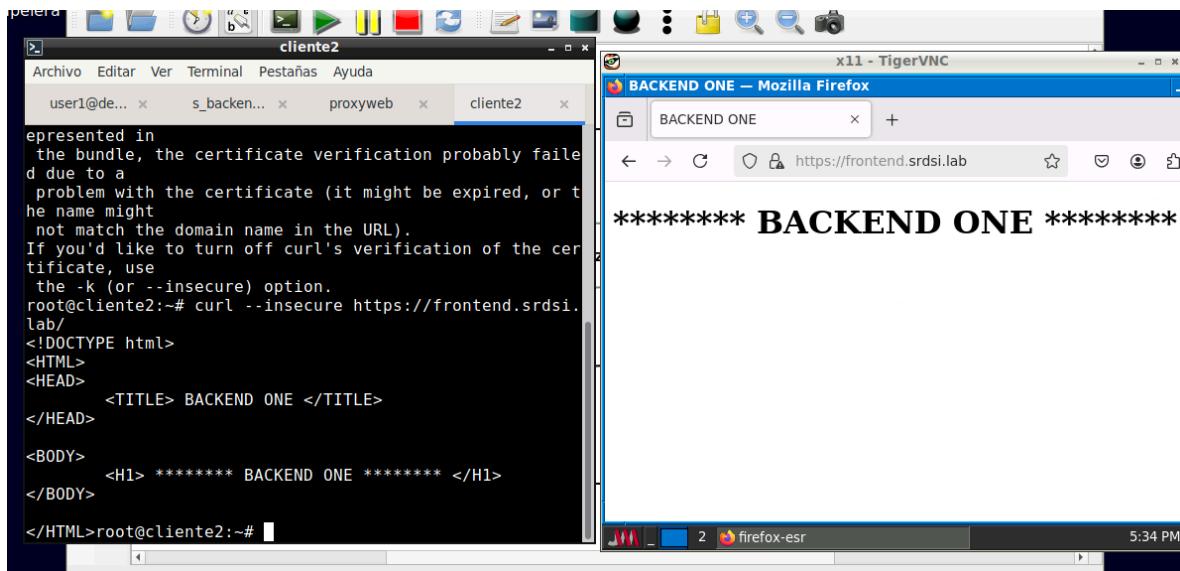
    ProxyRequests Off
    ProxyPreserveHost On

    ProxyPass / http://backend-1.srdsilab/
    ProxyPassReverse / http://backend-1.srdsilab/

    SSLEngine On
    SSLCertificateFile /etc/apache2/certificados/proxy.crt
    SSLCertificateKeyFile /etc/apache2/certificados/proxy.key
    SSLProxyEngine On

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

c) Comprobar que el acceso es seguro desde cliente1 y cliente2



Ambos clientes tienen acceso seguro, pero al usar certificados firmados por nosotros, debemos indicar en el navegador y con curl que nos fiamos.

## 2. Habilitar Balanceo de carga

Configuración de los servidores s\_backend-X

### 3. Personalizar página index.html asociada al sitio web.

Ambas máquinas ya cuentan con la página index.html personalizada, pudiendo identificarlas fácilmente.

```
root@s_backend-3:~# curl http://localhost
<!DOCTYPE html>
<HTML>
<HEAD>
  <TITLE> BACKEND THREE </TITLE>
</HEAD>
<BODY>
  <H1> ***** BACKEND THREE ***** </H1>
</BODY>
</HTML>root@s_backend-3:~#
```

```
root@s_backend-2:~# curl http://localhost
<!DOCTYPE html>
<HTML>
<HEAD>
  <TITLE> BACKEND TWO </TITLE>
</HEAD>
<BODY>
  <H1> ***** BACKEND TWO ***** </H1>
</BODY>
</HTML>root@s_backend-2:~#
```

falta personalizar con grupo



# Configuración del proxy proxyweb

## 1. Habilitar módulos

```
root@proxyweb:/etc/apache2/sites-available# a2enmod proxy proxy_http proxy_balanc
cer lbmethod bybusyness
Module proxy already enabled
Considering dependency proxy for proxy_http:
Module proxy already enabled
Module proxy_http already enabled
Considering dependency proxy for proxy_balancer:
Module proxy already enabled
Considering dependency alias for proxy_balancer:
Module alias already enabled
Considering dependency slotmem_shm for proxy_balancer:
Enabling module slotmem_shm.
Enabling module proxy_balancer.
Considering dependency proxy_balancer for lbmethod_bybusyness:
Considering dependency proxy for proxy_balancer:
Module proxy already enabled
Considering dependency alias for proxy_balancer:
Module alias already enabled
Considering dependency slotmem_shm for proxy_balancer:
Module slotmem_shm already enabled
Module proxy_balancer already enabled
Enabling module lbmethod_bybusyness.
To activate the new configuration, you need to run:
    service apache2 restart
root@proxyweb:/etc/apache2/sites-available# lbmethod_bytraffic

root@proxyweb:/etc/apache2/sites-available# service apache2 restart
[ ok ] Restarting web server: apache2.
root@proxyweb:/etc/apache2/sites-available#
```

## 2. Poner en marcha un sitio virtual (balanceo.srdsilab) con la funcionalidad de balanceo de carga habilitada:

Crea el fichero /etc/apache2/sites-available/balanceo.conf:

```

GNU nano 2.2.6                               File: frontend.conf
VirtualHost *:80>
    <Proxy balancer://mycluster >
        # Indicar todos los sitios virtuales en servidores s_backend-X
        BalancerMember http://backend-1.srdsi.lab:80/ loadfactor=2
        BalancerMember http://backend-2.srdsi.lab:80/
        BalancerMember http://backend-3.srdsi.lab:80/
    ProxySet lbmethod=bybusyness
    </Proxy>
    <Location /gestor-balanceo >
        SetHandler balancer-manager
    </Location>
    ServerName balanceo.srdsi.lab
    ProxyRequests Off
    ProxyPreserveHost Off
    ProxyPass /gestor-balanceo !
    ProxyPass / balancer://mycluster
    ProxyPassReverse / balancer://mycluster
/VirtualHost>

```

[ Wrote 18 lines ]

### 3. Habilitar el sitio virtual y reiniciar el servidor Apache

```

root@proxyweb:/etc/apache2/sites-available# a2ensite balanceo.conf
Enabling site balanceo.
To activate the new configuration, you need to run:
    service apache2 reload
root@proxyweb:/etc/apache2/sites-available# service apache2 reload
[ ok ] Reloading web server: apache2.
root@proxyweb:/etc/apache2/sites-available#

```

### 4. Comprobar que es correcta la resolución de nombres para los sitios web alojados en los servidores s\_backend-X

Desde proxyweb:

```
root@proxyweb:/etc/apache2/sites-available# curl -I http://backend-1.srdsi.lab/
HTTP/1.1 200 OK
Date: Fri, 16 May 2025 17:54:52 GMT
Server: Apache/2.4.10 (Debian)
Last-Modified: Sun, 14 Apr 2019 20:06:48 GMT
ETag: "87-5868313686600"
Accept-Ranges: bytes
Content-Length: 135
Vary: Accept-Encoding
Content-Type: text/html

root@proxyweb:/etc/apache2/sites-available# curl -I http://backend-2.srdsi.lab/
HTTP/1.1 200 OK
Date: Fri, 16 May 2025 17:54:54 GMT
Server: Apache/2.4.10 (Debian)
Last-Modified: Sun, 14 Apr 2019 20:06:48 GMT
ETag: "87-5868313686600"
Accept-Ranges: bytes
Content-Length: 135
Vary: Accept-Encoding
Content-Type: text/html

root@proxyweb:/etc/apache2/sites-available# curl -I http://backend-3.srdsi.lab/
HTTP/1.1 200 OK
Date: Fri, 16 May 2025 17:54:57 GMT
Server: Apache/2.4.10 (Debian)
Last-Modified: Sun, 14 Apr 2019 20:06:48 GMT
ETag: "8b-5868313686600"
Accept-Ranges: bytes
Content-Length: 139
Vary: Accept-Encoding
Content-Type: text/html

root@proxyweb:/etc/apache2/sites-available#
```

Todos los backend son accesibles.

5. Comprobar accesos desde el navegador del cliente y que se produce el balanceo entre servidores correctamente

Recargando la página podremos ver que se balancea la carga entre los servidores.





6. Podemos comprobar el estado del balanceador de carga accediendo a <http://balanceo.srdsi.lab/gestor-balanceo>

Si pinchamos en los enlaces podemos cambiar características del sitio virtual; por ejemplo, cambiar factor, para que un virtual host se visite más que otro (suponiendo que se ejecuta en una máquina más potente) mostrar capturas

a) ¿Qué ocurre si un servidor backend deja de estar activo?

Para probar esto vamos a parar el servicio apache en el backend-1

```
root@_backend-1:~# service apache2 stop
[ ok ] Stopping web server: apache2.
root@_backend-1:~#
```

Si volvemos al administrador del balanceo y recargamos, veremos que en el estado del backend-1 nos marca "Internal Error". Si recargamos la página [balanceo.srdsi.lab](http://balanceo.srdsi.lab) varias veces veremos que ya no nos devuelve el resultado del backend-1.

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	Busy	Load	To	From
<a href="http://backend-1.srdsi.lab/">http://backend-1.srdsi.lab/</a>			2	0	Init Err	5	0	-2	2.0K	496
<a href="http://backend-2.srdsi.lab/">http://backend-2.srdsi.lab/</a>			1	0	Init Ok	5	0	1	2.6K	620
<a href="http://backend-3.srdsi.lab/">http://backend-3.srdsi.lab/</a>			1	0	Init Ok	5	0	1	2.6K	630

Vamos a recuperar el servicio y ver si se reincorpora:

```
root@backend-1:~# service apache2 start
[ ok ] Starting web server: apache2.
```

Accedemos al administrador del balanceo y veremos que continúa en estado de error.

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	Busy	Load	To	From
<a href="http://backend-1.srdsi.lab/">http://backend-1.srdsi.lab/</a>			2	0	Init Err	5	0	-2	2.0K	496
<a href="http://backend-2.srdsi.lab/">http://backend-2.srdsi.lab/</a>			1	0	Init Ok	6	0	0	3.1K	744
<a href="http://backend-3.srdsi.lab/">http://backend-3.srdsi.lab/</a>			1	0	Init Ok	5	0	2	2.6K	630

Para incorporar de nuevo el servicio podemos darle a hacerlo manualmente dándole al boton de submit en la configuración del worker para backed-1

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	Busy	Load
<a href="http://backend-1.srdsi.lab/">http://backend-1.srdsi.lab/</a>			2	0	Init Err	11	0	0
<a href="http://backend-2.srdsi.lab/">http://backend-2.srdsi.lab/</a>			1	0	Init Ok	23	0	2
<a href="http://backend-3.srdsi.lab/">http://backend-3.srdsi.lab/</a>			1	0	Init Ok	22	0	2

### Edit worker settings for <http://backend-1.srdsi.lab/>

Load factor:	<input type="text" value="2"/>												
LB Set:	<input type="text" value="0"/>												
Route:	<input type="text"/>												
Route Redirect:	<input type="text"/>												
Status:	<table border="1"> <tr> <th>Ignore Errors</th> <th>Draining Mode</th> <th>Disabled</th> <th>Hot Standby</th> </tr> <tr> <td>On <input type="radio"/></td> <td>On <input type="radio"/></td> <td>On <input type="radio"/></td> <td>On <input type="radio"/></td> </tr> <tr> <td>Off <input checked="" type="radio"/></td> <td>Off <input checked="" type="radio"/></td> <td>Off <input checked="" type="radio"/></td> <td>Off <input checked="" type="radio"/></td> </tr> </table>	Ignore Errors	Draining Mode	Disabled	Hot Standby	On <input type="radio"/>	On <input type="radio"/>	On <input type="radio"/>	On <input type="radio"/>	Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>
Ignore Errors	Draining Mode	Disabled	Hot Standby										
On <input type="radio"/>	On <input type="radio"/>	On <input type="radio"/>	On <input type="radio"/>										
Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>										
<input type="button" value="Submit"/>													

Ahora veremos que el servicio se ha recuperado:

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	Busy	Load	To	From
<a href="http://backend-1.srdsi.lab/">http://backend-1.srdsi.lab/</a>			2	0	Init Ok	12	0	2	4.5K	1.1K
<a href="http://backend-2.srdsi.lab/">http://backend-2.srdsi.lab/</a>			1	0	Init Ok	24	0	-1	12K	2.9K
<a href="http://backend-3.srdsi.lab/">http://backend-3.srdsi.lab/</a>			1	0	Init Ok	22	0	3	11K	2.7K

Si queremos permitir que el propio balanceador compruebe si el servicio se ha recuperado para re-incluirlo en la configuración del balanceador podemos añadir al **BalancerMember** un parámetro **retry**:

```
<Proxy balancer://mycluster >
# Indicar todos los sitios virtuales en servidores s backend-X
BalancerMember http://backend-1.srdsi.lab:80/ retry=30 loadfactor=1
BalancerMember http://backend-2.srdsi.lab:80/ retry=30 loadfactor=1
BalancerMember http://backend-3.srdsi.lab:80/ retry=30 loadfactor=1
ProxySet lbmethod=bybusyness
</Proxy>
```

b) Por seguridad, los usuarios no deben tener acceso al gestor ni realizar ningún tipo de modificaciones ¿Cómo se podría restringir el acceso al gestor de balanceo?

Para que solo administradores puedan ver o modificar el balancer-manager, añadiremos dentro del <VirtualHost> del sitio de balanceo estas directivas en el bloque de <Location /gestor-balanceo>:

```
<Location /gestor-balanceo >  
    SetHandler balancer-manager  
  
    # Opcion 1: restringir por rango de IP  
    Require ip 10.0.0.0/24  
  
    # Opcion 2: autenticacion basica  
    AuthType Basic  
    AuthName "Balancer Manager"  
    AuthUserFile /etc/apache2/.htpasswd  
    Require valid-user  
</Location>
```

Con Require ip solo podrán acceder hosts de la red de administración.

Con Basic Auth se pedirá usuario/contraseña previo acceso al gestor.

El fichero /etc/apache2/.htpasswd no existirá, así que vamos a crearlo con unas credenciales de ejemplo:

```
root@proxyweb:~# htpasswd -b -c /etc/apache2/.htpasswd test test  
Adding password for user test  
root@proxyweb:~#
```

Usuario test, Contraseña test

Reiniciamos el servidor:

```
root@proxyweb:~# service apache2 reload  
[ ok ] Reloading web server: apache2.  
root@proxyweb:~#
```

Y ahora desde el client1 podremos acceder al contenido en <http://balanceo.srdsi.lab/>, pero nos solicitará usuario y contraseña para el panel.



Balancer Manager BACKEND TWO

balanceo.srdsi.lab/gestor-balanceo

# Load Balancer Manager

Server Version: Apache/2.4.10 (Debian) OpenSSL/1.0.1t  
 Server Built: Sep 30 2019 19:32:08  
 Balancer changes will NOT be persisted on restart.  
 Balancers are inherited from main server.  
 ProxyPass settings are inherited from main server.

## LoadBalancer Status for balancer://

MaxMembers	StickySession	DisableFailover
3 [3 Used]	(None)	Off

Username  
 Password

Cancel Sign in

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	Busy	Load	To	From
<a href="http://backend-1.srdsi.lab/">http://backend-1.srdsi.lab/</a>			2	0	Init Ok	12	0	2	4.5K	1.1K
<a href="http://backend-2.srdsi.lab/">http://backend-2.srdsi.lab/</a>			1	0	Init Ok	24	0	-1	12K	2.9K
<a href="http://backend-3.srdsi.lab/">http://backend-3.srdsi.lab/</a>			1	0	Init Ok	22	0	3	11K	2.7K

Apache/2.4.10 (Debian) Server at balanceo.srdsi.lab Port 80

Una vez introducidas las credenciales podremos acceder sin problema.

Balancer Manager BACKEND TWO

balanceo.srdsi.lab/gestor-balanceo

# Load Balancer Manager for balanceo.srdsi.lab

Server Version: Apache/2.4.10 (Debian) OpenSSL/1.0.1t  
 Server Built: Sep 30 2019 19:32:08  
 Balancer changes will NOT be persisted on restart.  
 Balancers are inherited from main server.  
 ProxyPass settings are inherited from main server.

## LoadBalancer Status for balancer://mycluster [p1fa4a21e\_mycluster]

MaxMembers	StickySession	DisableFailover	Timeout	FailoverAttempts	Method	Path	Active
3 [3 Used]	(None)	Off	0	2	bybusyness	/	Yes

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	Busy	Load	To	From
<a href="http://backend-1.srdsi.lab/">http://backend-1.srdsi.lab/</a>			2	0	Init Ok	1	0	2	538	124
<a href="http://backend-2.srdsi.lab/">http://backend-2.srdsi.lab/</a>			1	0	Init Ok	1	0	-1	538	124
<a href="http://backend-3.srdsi.lab/">http://backend-3.srdsi.lab/</a>			1	0	Init Ok	1	0	-1	538	126

Apache/2.4.10 (Debian) Server at balanceo.srdsi.lab Port 80



### 3. Enmascarar una URL con un Reverse Proxy

1. Activamos los módulos:

```
user1@demo: /var x proxyweb
root@proxyweb:/etc/apache2/sites-available# a2enmod proxy proxy_http
Module proxy already enabled
Considering dependency proxy for proxy_http:
Module proxy already enabled
Module proxy http already enabled
root@proxyweb:/etc/apache2/sites-available#
```

2. En la máquina proxyweb, crear el sitio virtual web para el servicio:  
`http://www.srdsi.lab :8080`

Poner la siguiente configuración en `/etc/apache2/sites-available/www.conf`

```
user1@demo: /var x proxyweb
<VirtualHost *:8080>
    ServerName www.srdsi.lab
    DocumentRoot /var/www/www

    ErrorLog ${APACHE_LOG_DIR}/www-error.log
    CustomLog ${APACHE_LOG_DIR}/www-access.log combined
</VirtualHost>
```

Y añadir el siguiente `index.html` en `/var/www/www`:

```
user1@demo: /var x proxyweb
<!DOCTYPE html>
<html>
<head>
    <title>Servicio Web Original</title>
</head>
<body>
    <h1>Servicio Web en puerto 8080</h1>
    <p>Este es el sitio original que está funcionando en www.srdsi.lab:8080</p>
    <p>Fecha y hora: <?php echo date('Y-m-d H:i:s'); ?></p>
</body>
</html>
```

3. También crear el sitio virtual para <http://servicio.srdsi.lab>

En `/etc/apache2/sites-available/servicio.conf`:



```
user1@demo: /var x proxyweb
<VirtualHost *:80>
    ServerName servicio.srdsi.lab

    ProxyRequests Off
    ProxyPreserveHost Off
    ProxyPass / http://www.srdsi.lab:8080/
    ProxyPassReverse / http://www.srdsi.lab:8080/

    ErrorLog ${APACHE_LOG_DIR}/servicio-error.log
    CustomLog ${APACHE_LOG_DIR}/servicio-access.log combined
</VirtualHost>
~
~
```

4. Habilitar el puerto de escucha en /etc/apache2/ports.conf

```
Listen 80
Listen 8080
```

5. Habilitar los dos sitios virtuales y reiniciar el servidor Apache

En /etc/hosts:

```
127.0.0.1 www.srdsi.lab
127.0.0.1 servicio.srdsi.lab
```

```
user1@demo: /var x proxyweb
root@proxyweb:/etc/apache2# a2ensite www.conf
Site www already enabled
root@proxyweb:/etc/apache2# a2ensite servicio.conf
Site servicio already enabled
root@proxyweb:/etc/apache2# service apache2 reload
[ ok ] Reloading web server: apache2.
root@proxyweb:/etc/apache2#
```

Adicionalmente se han añadido las siguientes reglas iptable en el router para redirigir el tráfico del puerto 8080:

```
root@FW:~# iptables -t nat -A PREROUTING -p tcp -d 203.0.113.24 --dport 8080 -j DNAT --to-destination 10.0.0.100:8080
root@FW:~# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
root@FW:~# iptables -A FORWARD -p tcp -d 10.0.0.100 --dport 8080 -j ACCEPT
root@FW:~# iptables -A FORWARD -p tcp -s 10.0.0.100 --sport 8080 -j ACCEPT
```

6. Realizar capturas con Wireshark y comprobar accesos desde el navegador

de un cliente.

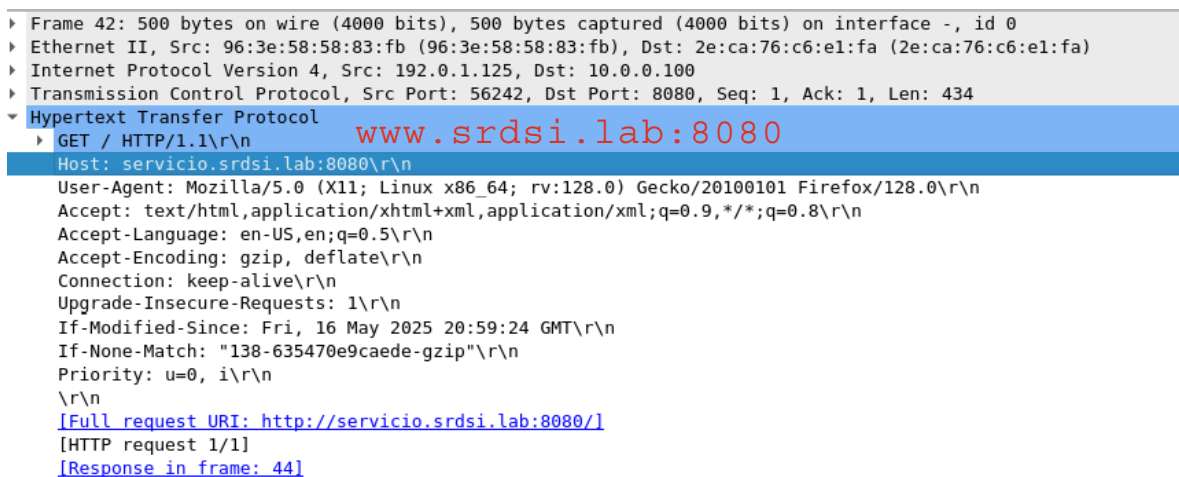
a) Acceder a <http://www.srdsi.lab:8080>



b) Acceder a <http://servicio.srdsi.lab>



Al realizar la conexión al puerto 8080 y analizar un paquete get entre el firewall y el proxy, podemos ver que la cabecera host contiene el puerto 8080:



Al realizar la misma captura indicando el dominio sin el puerto, la cabecera host no tiene el puerto:

```

> Frame 54: 495 bytes on wire (3960 bits), 495 bytes captured (3960 bits) on interface -, id 0
> Ethernet II, Src: 96:3e:58:58:83:fb (96:3e:58:58:83:fb), Dst: 2e:ca:76:c6:e1:fa (2e:ca:76:c6:e1:fa)
> Internet Protocol Version 4, Src: 192.0.1.125, Dst: 10.0.0.100
> Transmission Control Protocol, Src Port: 53302, Dst Port: 80, Seq: 1, Ack: 1, Len: 429
< Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: servicio.srdsi.lab\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    If-Modified-Since: Fri, 16 May 2025 20:59:24 GMT\r\n
    If-None-Match: "138-635470e9caede-gzip"\r\n
    Priority: u=0, i\r\n
    \r\n
    [Full request URI: http://servicio.srdsi.lab/]
    [HTTP request 1/1]
    [Response in frame: 56]
```