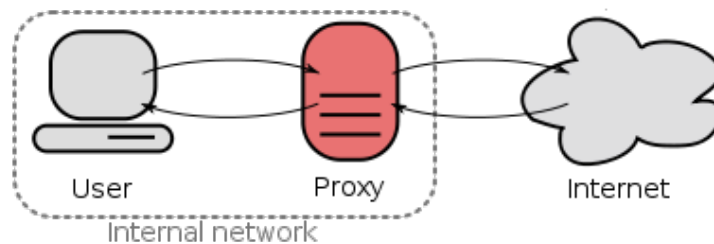


Actividad: Configuración de Apache como *proxy*

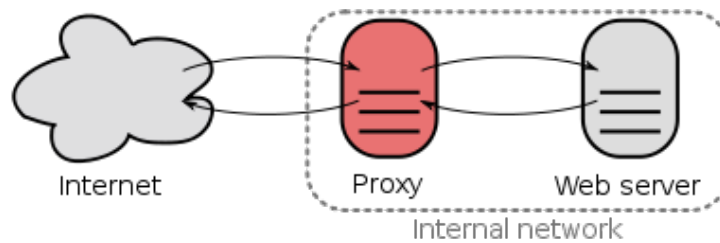
Un servicio proxy es un servicio que actúa como intermediario en una comunicación cliente-servidor.

Un servidor proxy (*Forward Proxy*) permite que no se contacte directamente un servidor, sino que es el proxy quien se conecta con dicho servidor, manteniendo al cliente en el anonimato; un proxy inverso (*Reverse Proxy*), mantiene al servidor en el anonimato para sus clientes.

Forward Proxy:



Reverse Proxy:



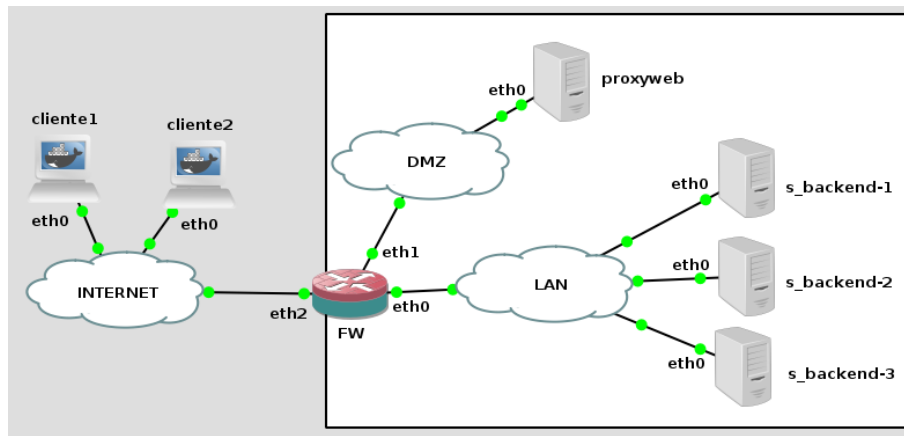
Documentación a entregar

Soluciones documentadas y comentadas de los ejercicios propuestos. Indicar tiempo dedicado a su realización.

Puesta en marcha: GNS3

- En la máquina virtual SRDSI-Debian11-GNS3¹:
 1. [Hecho] Descargar contenedor actualizado desde *egela* (`iptermlog.tar.gz`) y cargarlo en la máquina virtual (`docker load < iptermlog.tar.gz`)
 2. Descargar el proyecto `proxy-web-apache` de *egela*.
 3. Descomprimir el proyecto (`tar xvzf fichero.tar.gz -C path/destino`). Lanzar GNS3 (no actualizar), abrir el proyecto y poner en marcha todos los dispositivos.

Entorno de trabajo:



- *s_backend-X*, máquinas en la red interna que tienen réplicas del servidor web interno.
- *proxyweb*, máquina en la DMZ donde habrá que configurar un *reverse proxy* hacia el cluster de servidores web internos. Alojara los sitios web públicos.

¹Máquina virtual utilizada en la actividad sobre *firewalls*

1. Configurar Apache como *Reverse Proxy*

Para acceder a nuestros servidores web desde Internet los clientes solo podrán acceder a través de un sitio web público (`frontend.srdsi.lab`). Necesitamos disponer de un *reverse proxy*, en este caso utilizaremos el software Apache, donde habilitaremos una serie de módulos², que dependen del protocolo a través del cual se desea acceder al servidor de *backend*, en nuestro caso un servidor web.

En este primer ejercicio solo accediremos al servidor `s_backend-1` que aloja la página `backend-1.srdsi.lab`.

Configuración del servidor web *s_backend-1*

1. [Hecho] Asignar nombre a la máquina: `s_backend-1` y dominio: `srdsi.lab`
2. [Hecho] Poner en marcha un sitio web virtual `backend-1.srdsi.lab`
3. Personalizar página `index.html` asociada al sitio web. Verificar acceso al sitio web.

Configuración del proxy *proxyweb*

1. [Hecho] Asignar nombre a la máquina: `proxyweb` y dominio: `srdsi.lab`
2. [Hecho] Añadir en `/etc/hosts` la resolución de nombre para el sitio virtual web:

```
@IP_s_backend-1 backend-1.srdsi.lab
```
3. [Hecho] Instalar Apache (si no lo está)
4. Habilitar los módulos necesarios

```
$ a2enmod proxy proxy_http
```

 - `mod_proxy`, habilita a Apache para gestionar y redirigir conexiones
 - `mod_proxy_http`, habilita conexiones HTTP(S) entre el servidor proxy y el servidor web
5. Poner en marcha el sitio virtual (`frontend.srdsi.lab`) que actuará como *reverse proxy* [ver anexo]
6. Habilitar la funcionalidad de proxy en el sitio virtual web creado añadiendo las siguientes directivas:

```
<VirtualHost *:80>
...
# evita que el front-end sea utilizado como proxy forward
ProxyRequests Off
```

²http://httpd.apache.org/docs/trunk/mod/mod_proxy.html

el paso del *proxyweb* al servidor interno (*s_backend-1*) sea transparente para el usuario

```
ProxyPreserveHost Off
```

```
# gestionan el salto y la vuelta del proxyweb al s_backend-1.
```

```
ProxyPass / http://backend-1.srdsi.lab/
```

```
ProxyPassReverse / http://backend-1.srdsi.lab/
```

```
...
```

```
</VirtualHost>
```

7. Habilitar el sitio virtual y reiniciar el servidor Apache

8. Comprobar accesos desde un navegador

a) En el servidor *proxyweb* (navegador *lynx*)

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

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.

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

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

a) Habilitar el modulo en Apache para TLS

b) Modificar el sitio web

```
<VirtualHost *:80>
```

```
    ServerName frontend.srdsi.lab
```

```
    Redirect / https://frontend.srdsi.lab/
```

```
</VirtualHost>
```

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

```
    ...
```

```
    SSLEngine On
```

```
    SSLCertificateFile /etc/apache2/certificados/proxy.crt
```

```
    SSLCertificateKeyFile /etc/apache2/certificados/proxy.key
```

```
    SSLProxyEngine On
```

```
</VirtualHost>
```

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

2. Habilitar Balanceo de carga

Si tenemos múltiples servidores webs *backend*, podemos distribuir las conexiones balanceando la carga entre ellos configurando el proxy en Apache.

Configuración de los servidores *s_backend-X*

1. [Hecho] Asignar nombre a la máquina: *s_backend-X* y dominio: *srdsi.lab*
2. [Hecho] Poner en marcha un sitio web virtual *backend-X.srdsi.lab*
3. Personalizar página *index.html* asociada al sitio web.

Configuración del proxy *proxyweb*

1. Habilitar módulos

```
$ a2enmod proxy proxy_http proxy_balancer lbmethod_bybusyness  
lbmethod_bytraffic3
```

2. Poner en marcha un sitio virtual (*balanceo.srdsi.lab*) con la funcionalidad de balanceo de carga habilitada:

```
<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 ... ]  
        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>
```

³https://httpd.apache.org/docs/2.4/es/mod/mod_proxy_balancer.html

3. Habilitar el sitio virtual y reiniciar el servidor Apache
4. Comprobar que es correcta la resolución de nombres para los sitios web alojados en los servidores *s_backend-X*
5. Comprobar accesos desde el navegador del cliente y que se produce el balanceo entre servidores correctamente.

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)

- a) ¿Qué ocurre si un servidor backend deja de estar activo?
 - ¿Qué información muestra el gestor?
 - ¿Se incorpora al gestor de balanceo una vez recuperado?
 - ¿Existe alguna configuración que permita el control de esta situación? ⁴
- 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?

⁴https://httpd.apache.org/docs/2.4/mod/mod_proxy.html#proxypass

3. Enmascarar una URL con un *Reverse Proxy*

Tenemos instalado un servicio web accesible a través de `http://www.srdsi.lab:8080`. Pero queremos acceder a dicho servicio mediante `http://servicio.srdsi.lab` porque es más fácil de recordar por los usuarios. Además, queremos que se mantenga oculta la URL real a los usuarios, y no dejar accesible el puerto 8080 desde el exterior.

Solución: enmascarar la verdadera URL con un *Reverse Proxy*

Los pasos a seguir son:

1. Activamos los módulos:

```
$ a2enmod proxy proxy_http
```
2. En la máquina proxyweb, crear el sitio virtual web para el servicio: `http://www.srdsi.lab:8080`
3. También crear el sitio virtual para `http://servicio.srdsi.lab`

```
<VirtualHost *:80>
    ServerName servicio.srdsi.lab
    [ProxyPreserveHost Off]
    ProxyRequests Off
    ProxyPass / http://www.srdsi.lab:8080/
    ProxyPassReverse / http://www.srdsi.lab:8080/
    ...
</VirtualHost>
```
4. Habilitar el puerto de escucha en `/etc/apache2/ports.conf`

```
Listen 8080
```
5. Habilitar los dos sitios virtuales y reiniciar el servidor Apache
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`

Anexos

- Asignar nombre y dominio a una máquina
 - Cambiar nombre

```
$ hostnamectl set-hostname nombremáquina
```
 - Indicar el dominio en `/etc/hosts`

```
127.0.0.1 nombremáquina.nombredominio nombremáquina
```
 - Al ejecutar: `$ hostname --fqdn`
debemos obtener: `nombremáquina.nombredominio`
- Poner en marcha un sitio virtual web en el servidor Apache2
 1. Añadir la resolución de nombres para el sitio web en `/etc/hosts`

```
@IP_maq_aloja_sitioweb (127.0.0.1) sitioweb.nombredominio
```
 2. Configurar sitio web en Apache `/etc/apache2/sites-available/sitioweb.conf`

```
<VirtualHost *:puerto>
    ServerName sitioweb.nombredominio
    DocumentRoot /var/www/sitioweb
    ...
    ErrorLog "/var/log/apache2/sitioweb-error.log"
    CustomLog "/var/log/apache2/sitioweb-access.log" common
    ...
</VirtualHost>
```
 3. Crear el directorio raíz para el nuevo sitio web (`/var/www/sitioweb`).
Añadir el fichero `/var/www/sitioweb/index.html` que muestre una información distintiva para el nuevo sitio web `sitioweb`.
 4. Habilitar sitio web y reiniciar servicio web:

```
$ a2ensite sitioweb
$ service apache2 restart [apache2ctl graceful]
```
 5. Comprobar accesos desde el navegador `http://sitioweb.nombredominio`

Otras funcionalidades

- Web caché (necesita `mod_cache`)
`ProxyRequests on`
`CacheRoot /var/spool/httpd/proxy`
- Filtrar el contenido “*proxied*”, antes de enviar al usuario (necesita `mod_ext_filter`)
`ExtFilterDefine naughtywords mode=output intype=text/html cmd="/bin/sed s/darned/ blasted/g"`
`<Proxy *>`
`SetOutputFilter naughtywords`
`</Proxy>`
- Autenticación para acceder al proxy
`ProxyPass "/secretserver/" "http://127.0.0.1:8080"`
`<Directory "proxy:http://127.0.0.1:8080/">`
`AuthName SecretServer`
`AuthType Basic`
`AuthUserFile /path/to/secretserver.htpasswd`
`Require valid-user`
`</Directory>`
- Asegurar el servidor proxy, controlar accesos
`<Proxy *>`
`Require ip @IP-redlocal`
`</Proxy>`
- Bloquear peticiones proxy a ciertos sitios
`ProxyBlock forbiddensite.com www.competitor.com monster.com`