

Trabajos prácticos

Aquí se proponen ejercicios para implementar algunos de los puntos abordados en el capítulo. En cada uno de ellos se da un ejemplo comentado de la realización del ejercicio, que deberá adaptar a la configuración de sus sistemas.

1. Servidor HTTP Apache con dos hosts virtuales

Configuramos un servidor HTTP Apache en una distribución de tipo Red Hat. Apache debe administrar dos hosts virtuales, diferenciados gracias a su nombre de host:

- ✓ `www`: este host virtual está accesible para todos los clientes HTTP.
- ✓ `rh`: este host virtual solamente está accesible para los usuarios que tengan una cuenta creada localmente en el servidor HTTP Apache.

Los nombres de host pueden ser definidos como alias DNS en la máquina host, o declarados en los archivos `hosts` del servidor y de los clientes que se utilizarán para hacer las pruebas.

Comandos y archivos útiles

- ✓ `rndc`
- ✓ `/etc/hosts`
- ✓ `host`
- ✓ `/etc/httpd/conf/httpd.conf`
- ✓ `http`
- ✓ `systemctl`
- ✓ `wget`
- ✓ `htpasswd`
- ✓ `firefox`

Etapas

1. Compruebe que el paquete de software del servidor HTTP Apache está instalado. Para simplificar las pruebas, desactive temporalmente el firewall del servidor.
2. Declare los dos nombres de host `www` y `rh` (por DNS o en el archivo `/etc/hosts`).
3. Configure el servidor HTTP Apache con el host virtual público `www`.
4. Cree el directorio de datos del host virtual, con una página HTML de prueba.
5. Compruebe y cargue la configuración.
6. Compruebe el acceso al host virtual `www`, desde la línea de comandos o un navegador local.
7. Compruebe el acceso al host virtual `www`, desde un navegador remoto.
8. Cree una base de cuentas local para el control de acceso al host virtual `rh`.
9. Configure el servidor HTTP Apache con el host virtual público `rh`, accesible solamente para las cuentas declaradas en la base de cuentas locales.
10. Cree el directorio de datos del host virtual, con una página HTML de prueba.
11. Compruebe y cargue la configuración.
12. Compruebe el control de acceso al host virtual `rh`, desde la línea de comandos o en un navegador local.
13. Compruebe el acceso al host virtual `rh`, desde un navegador remoto.

Resumen de comandos y resultado en pantalla

1. Compruebe que el paquete de software del servidor HTTP Apache está instalado. para simplificar las pruebas, desactive temporalmente el firewall del servidor.

En las distribuciones de tipo Red Hat, el paquete de software se llama `httpd`:

```
[root@centos8 ~]# yum list httpd
Última comprobación de caducidad de metadatos hecha hace 0:54:54 el sab
23 mayo 2020 11:34:12 CEST.
Paquetes disponibles
httpd.x86_64    2.4.37-16.module_el8.1.0+256+ae790463 @AppStream
```

El paquete está instalado.

Desactivamos el firewall del servidor:

```
[root@centos8 ~]# systemctl stop firewalld
```

2. Declare los dos nombres de hosts `www` y `rh` (por DNS o en el archivo `/etc/hosts`).

Como hemos configurado un servidor DNS durante los trabajos prácticos anteriores, vamos a usarlo para declarar dos alias en la zona `midns.es`:

```
[root@centos8 ~]# vi /var/named/db.midns.es
; Archivo de zona midns.es.
$TTL 1D ; Duración de vida por defecto 1 día
; Registro de declaración de la zona:
@ IN SOA centos8.midns.es. admin.midns.es. (
2020102301; serial
6H; refresh
1H; retry
```

```

2D; expire
1H); mínimo
; Servidores DNS:
@      IN    NS    centos8.midns.es.
@      IN    NS    debian10.midns.es.
; Direcciones (IPv4):
centos8    IN    A    192.168.1.60
debian10   IN    A    192.168.1.70
puesto     IN    A    192.168.1.24
puesto 1   IN    A    192.168.1.25
puesto 2   IN    A    192.168.1.26
www61      IN    A    192.168.1.61
; Alias:
www        IN    CNAME centos8
rh         IN    CNAME centos8
ftp        IN    CNAME debian10

```

Declaramos dos registros de tipo `CNAME`, alias del host `centos8.midns.es.`, y aumentamos el número de serie del archivo de zona.

Se comprueba el archivo de zona:

```

[root@centos8 ~]# named-checkzone midns.es /var/named/db.midns.es
zone midns.es/IN: loaded serial 2020102301
OK

```

Cargamos la zona:

```

[root@centos8 ~]# rndc reload
server reload successful

```

Comprobamos la resolución de nombres:

```

[root@centos8 ~]# host www.midns.es
www.midns.es is an alias for centos8.midns.es.
centos8.midns.es has address 192.168.1.60
[root@centos8 ~]# host rh.midns.es

```

rh.midns.es is an alias for centos8.midns.es.
centos8.midns.es has address 192.168.1.60

3. Configure el servidor HTTP Apache con el host virtual público `www`.

Vamos a declarar una sección `VirtualHost`, asociada al nombre de host `www.midns.es` y al directorio `/var/www/html/www`.

```
[root@centos8 ~]# vi /etc/httpd/conf/httpd.conf
[...]  
# Host virtual por nombre de host  
<VirtualHost www.midns.es>  
  ServerName www.midns.es  
  DocumentRoot /var/www/html/www  
  ErrorLog /var/log/httpd/www-err.log  
  TransferLog /var/log/httpd/www-acc.log  
</VirtualHost>  
[...]
```

4. Cree el directorio de datos del host virtual, con una página HTML de prueba.

Creamos el directorio de datos del host virtual, con una página HTML de prueba:

```
[root@centos8 ~]# mkdir /var/www/html/www  
[root@centos8 ~]# vi /var/www/html/www/index.html  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html>  
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>  
  <title>Servidor Web www</title>  
</head>  
<body>
```

```
<h1>Bienvenido al sitio www.midns.es</h1>
</body>
</html>
```

5. Compruebe y cargue la configuración.

Comprobamos la configuración del servidor HTTP Apache:

```
[root@centos8 ~]# httpd -t
Syntax OK
```

Recargamos la configuración:

```
[root@centos8 ~]# systemctl reload httpd
```

6. Compruebe el acceso al host virtual `www`, desde la línea de comandos o un navegador local.

Usamos el comando `wget` para comprobar el acceso al host virtual. La opción `-O -` fuerza la visualización de la respuesta del servidor en lugar de almacenarla en un archivo:

```
[root@centos8 ~]# wget -O - www.midns.es
--2020-05-23 15:13:47-- http://www.midns.es/
Resolviendo de www.midns.es (www.midns.es)... 192.168.1.60
Conectando con www.midns.es (www.midns.es)[192.168.1.60]:80... conectado.
Petición HTTP enviada, esperando respuesta ... 200 OK
Longitud:305 [text/html]
Grabando en:« STDOUT »

-          0%[          ] 0
--.-KB/s  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Servidor Web www</title>
</head>
<body>
<h1>Bienvenido al sitio www.midns.es</h1>
</body>
</html>
-      100%[=====] 305
--.-KB/s  ds 0s

2020-05-23 15:13:47 (36,4 MB/s) — escritos a stdout [305/305]

[root@centos8 ~]#

```

7. Compruebe el acceso al host virtual `www`, desde un navegador remoto.

Vamos a usar el navegador Firefox en la máquina de la red local.

Debe ser cliente DNS de 1 de los servidores DNS de la zona `midns.es`.

```

[root@centos7 ~]# cat /etc/resolv.conf
; generated by /usr/sbin/dhclient-script
nameserver 192.168.1.60
nameserver 192.168.1.254
[root@centos7 ~]#

```

Comprobamos la resolución de nombres:

```

[root@centos7 ~]# host www.midns.es
www.midns.es is an alias for centos8.midns.es.
centos8.midns.es has address 192.168.1.60

```

Desde el navegador Firefox, solicitamos la URL `www.midns.es` :



El host virtual está operativo.

8. Cree una base de cuentas local para el acceso al host virtual `rh`.

Vamos a crear un archivo de base de cuentas local, para un solo usuario `drh`, con el comando `htpasswd`. Usamos la opción `-c` para crear el archivo:

```
[root@centos8 ~]# htpasswd -c /etc/httpd/passwd.rh drh
New password: XXX
Re-type new password: XXX
Adding password for user drh
[root@centos8 ~]# cat /etc/httpd/passwd.rh
drh:$apr1$d8epvDhr$mqet8Vy6qh/D50nmKY13v/
```

9. Configure el servidor HTTP Apache con el host virtual público `rh`, accesible solamente para las cuentas declaradas en la base de cuentas locales.

Vamos a declarar una sección `VirtualHost`, asociada al nombre de host `rh.midns.es` y al directorio `/var/www/html/rh`. También es necesario crear una sección `Directory`, de acceso protegido, autorizada solamente para las cuentas de usuarios de la base de cuentas local creada anteriormente:


```
[...]
# Host virtual por nombre de host
<VirtualHost rh.midns.es>
  ServerName rh.midns.es
  DocumentRoot /var/www/html/rh
  ErrorLog /var/log/httpd/rh-err.log
  TransferLog /var/log/httpd/rh-acc.log
</VirtualHost>
# Directorio de acceso protegido
<Directory /var/www/html/rh>
  AuthType basic
  AuthUserFile /etc/httpd/passwd.rh
  AuthName "Identificación obligatoria"
  Require valid-user
</Directory>
[...]
```

Comprobamos que los módulos necesarios para la autenticación con base de cuentas local están cargados:

```
[root@centos8 ~]# httpd -M | grep auth_basic_module
auth_basic_module (shared)
[root@centos8 ~]# httpd -M | grep authn_file_module
authn_file_module (shared)
[root@centos8 ~]# httpd -M | grep authz_user_module
authz_user_module (shared)
```

10. Cree el directorio de datos del host virtual, con una página HTML de prueba.

Creamos el directorio de datos del host virtual, con una página HTML de test:

```
[root@centos8 ~]# mkdir /var/www/html/rh
[root@centos8 ~]# vi /var/www/html/rh/index.html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
```

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Intranet DRH</title>
</head>
<body>
<h1>Bienvenido a la intranet de recursos humanos </h1>
</body>
</html>
```

11. Compruebe y cargue la configuración.

Comprobamos la configuración del servidor HTTP Apache:

```
[root@centos8 ~]# httpd -t
Syntax OK
```

Recargamos la configuración:

```
[root@centos8 ~]# systemctl reload httpd
```

12. Compruebe el control de acceso al host virtual `rh`, desde la línea de comandos o en un navegador local.

Usamos el comando `wget` para comprobar el acceso al host virtual:

```
[root@centos8 ~]# wget -O - rh.midns.es
--2020-05-23 15:46:59-- http://rh.midns.es/
Resolviendo rh.midns.es (rh.midns.es)... 192.168.1.60
Conectando con rh.midns.es (rh.midns.es)[192.168.1.60]:80... conectado.
Petición HTTP enviada, en espera de la respuesta... 401 Unauthorized
```

La autenticación usuario/contraseña falló.

El acceso ha sido denegado, porque no hemos proporcionado una cuenta de usuario válida.

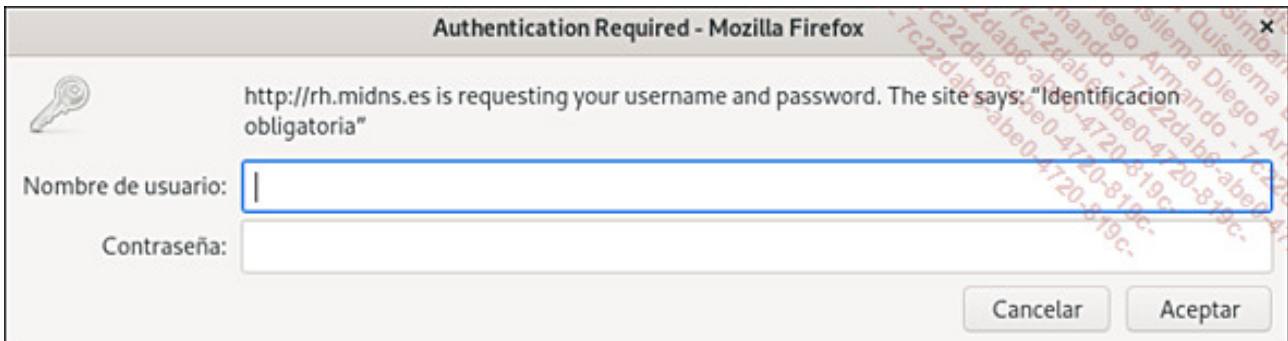
Comprobamos En el archivo de registro del acceso del host virtual, `/var/log/httpd/rh-acc.log` :

```
[root@centos8 ~]# vi /var/log/httpd/rh-acc.log
192.168.1.60 - - [23/May/2020:15:46:59 +0200] "GET / HTTP/1.1" 401 381
```

La petición del cliente ha sido denegada con un código `401` (acceso no autorizado).

13. Compruebe el acceso al host virtual `rh`, desde un navegador remoto.

Desde el navegador Firefox de la máquina cliente que ya se utilizó, solicitamos la URL `rh.midns.es` :



Una cuenta de usuario y su contraseña son solicitados.



El host virtual está operativo.

2. Servidor proxy HTTP Squid

Para proteger las máquinas de la intranet cuando navegan en Internet y optimizar los tiempos de acceso, se decide implementar un servidor proxy HTTP Squid en una distribución de tipo Debian. Este servidor solamente será accesible desde las redes en las que el mismo participe directamente.

Comandos y archivos útiles

- ✓ `dpkg-query`
- ✓ `/etc/squid/squid.conf`
- ✓ `systemctl`
- ✓ `wget`
- ✓ `firefox`

Etapas

1. Compruebe que el paquete de software del servidor Squid está instalado.
2. Configure el servidor Squid como servidor proxy y de caché. Inicie o reinicie el servidor.
3. Compruebe el acceso a un sitio de Internet, desde la línea de comandos o con un navegador local configurado para usar el servidor proxy.
4. Compruebe el acceso a un sitio de Internet, desde un navegador remoto configurado para usar el servidor. Pare el servidor proxy Squid y compruebe de nuevo.

Resumen de comandos y resultado en pantalla

1. Compruebe que el paquete del servidor Squid está instalado.

```

root@debian10:~# dpkg-query -f='${Package}
Deseado=desconocido(U)/Instalar/eliminaR/Purgar/retener(H)
| Estado=No/Inst/ficheros-Conf/desempaquetado/medio-conf/medio-inst(H)/
espera-disparo(W)/pendiente-disparo
|/ Err?=(ninguno)/requiere-Reinst (Estado,Err: mayúsc.=malo)
||/ Nombre      Versión      Arquitectura Descripción
+++-----
ii squid        4.6-1+deb10u6 amd64    Full featured Web Proxy cache
(HTTP proxy)

```

El paquete está instalado.

Comprobemos si el servidor está activo:

```

root@debian10:~# systemctl status squid
squid.service - Squid Web Proxy Server

```

```
Loaded: loaded (/lib/systemd/system/squid.service; enabled; vendor preset:
enabled)
Active: active (running) since Fri 2020-05-22 11:00:57 CEST; 1 day 5h ago
Docs: man:squid(8)
[...]
```

El servidor Squid se encuentra en arranque automático y está activo.

2. Configure el servidor Squid como servidor proxy y de caché. Inicie o reinicie el servidor Squid.

Se tiene que modificar la configuración por defecto para autorizar a los clientes de las redes locales:

```
root@debian10:~# vi /etc/squid/squid.conf
acl localnet src 0.0.0.1-0.255.255.255 # RFC 1122 "this" network (LAN)
acl localnet src 10.0.0.0/8          # RFC 1918 local private network (LAN)
acl localnet src 100.64.0.0/10       # RFC 6598 shared address space (CGN)
acl localnet src 169.254.0.0/16      # RFC 3927 link-local (directly plugged) machines
acl localnet src 172.16.0.0/12       # RFC 1918 local private network (LAN)
acl localnet src 192.168.1.0/16      # RFC 1918 local private network (LAN)
acl localnet src fc00::/7           # RFC 4193 local private network range
acl localnet src fe80::/10          # RFC 4291 link-local (directly plugged) machines
acl SSL_ports port 443
acl Safe_ports port 80              # http
acl Safe_ports port 21              # ftp
acl Safe_ports port 443             # https
acl Safe_ports port 70              # gopher
acl Safe_ports port 210             # wais
acl Safe_ports port 1025-65535      # unregistered ports
acl Safe_ports port 280             # http-mgmt
acl Safe_ports port 488             # gss-http
acl Safe_ports port 591             # filemaker
acl Safe_ports port 777             # multiling http
acl CONNECT method CONNECT
http_access deny !Safe_ports
```

```

http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
include /etc/squid/conf.d/*
http_access allow localhost
http_access allow localnet
http_access deny all
http_port 3128
[...]

```

Recargamos la configuración:

```
root@debian10:~# systemctl reload squid
```

3. Compruebe el acceso a un sitio de Internet, desde la línea de comandos o con un navegador local configurado para usar el servidor proxy.

Usamos el comando `wget` para comprobar un acceso a Internet, a través del servidor proxy Squid. El comando lee el nombre del proxy en la variable de entorno `http_proxy`:

```

root@debian10:~# http_proxy=localhost:3128; export http_proxy
root@debian10:~# wget -O - www.debian.org |more
Resolviendo localhost (localhost)... ::1, 127.0.0.1
Conectando con localhost (localhost)[::1]:3128... conectado.
Petición Proxy enviada, esperando respuesta... 302 Found
Localización: https://www.debian.org/ [siguiendo]
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 15111 (15K) [text/html]
Grabando a: "STDOUT"

-                                0%[
]  0 --.-KB/s               <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>

```

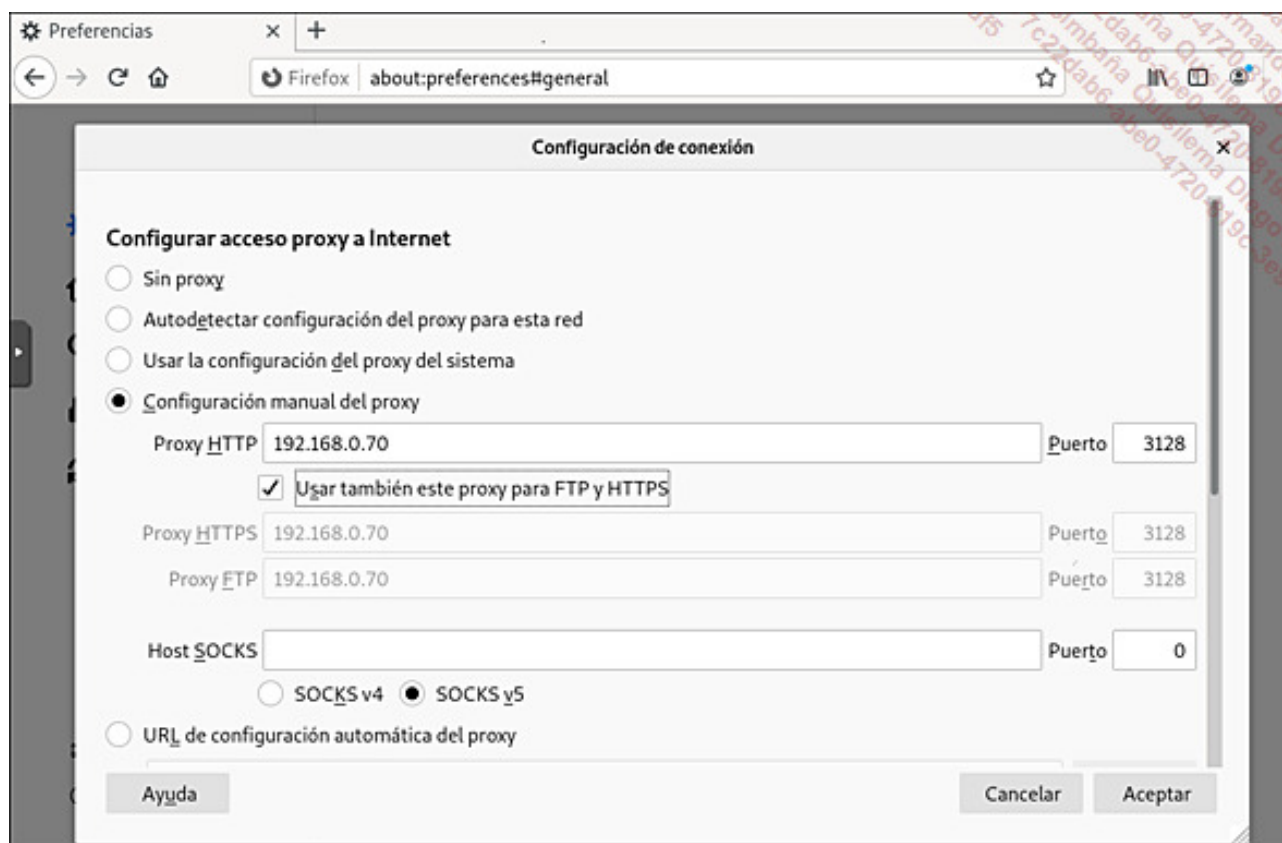
```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Debian -- The Universal Operating System </title>
[...]
```

El comando ha sido ejecutado correctamente por el servidor proxy para obtener la página de inicio del sitio `www.debian.org`. Podemos comprobarlo en el archivo de registro del servidor Squid:

```
root@debian10:~# vi /var/log/squid/access.log
[...]
1590245004.562 66::1 TCP_MISS/302 685 GET http://www.debian.org/ -
HIER_DIRECT/2001:67c:2564:a119::77 text/html
1590245022.021 155::1 TCP_TUNNEL/200 19875 CONNECT www.debian.org:443 -
HIER_DIRECT/2001:67c:2564:a119::77 -
1590245050.668 153::1 TCP_TUNNEL/200 19875 CONNECT www.debian.org:443 -
HIER_DIRECT/2001:67c:2564:a119::77 -
```

4. Compruebe el acceso a un sitio de Internet, desde un navegador remoto configurado para usar el servidor. Pare el servidor proxy Squid y compruebe de nuevo.

Configuremos el navegador Firefox de una máquina de la red local, para usar el servidor proxy Squid (preste atención al número de puerto, por defecto Firefox propone 8080):



A continuación podemos usar normalmente el navegador:



Podemos comprobar la entrada en el archivo de registro del servidor Squid:

```

root@debian10:~# vi /var/log/squid/access.log
[...]
1590246115.509 223 192.168.1.5 TCP_TUNNEL/200 14157 CONNECT
www.debian.org:443 - HIER_DIRECT/2001:67c:2564:a119::77 -
[...]

```

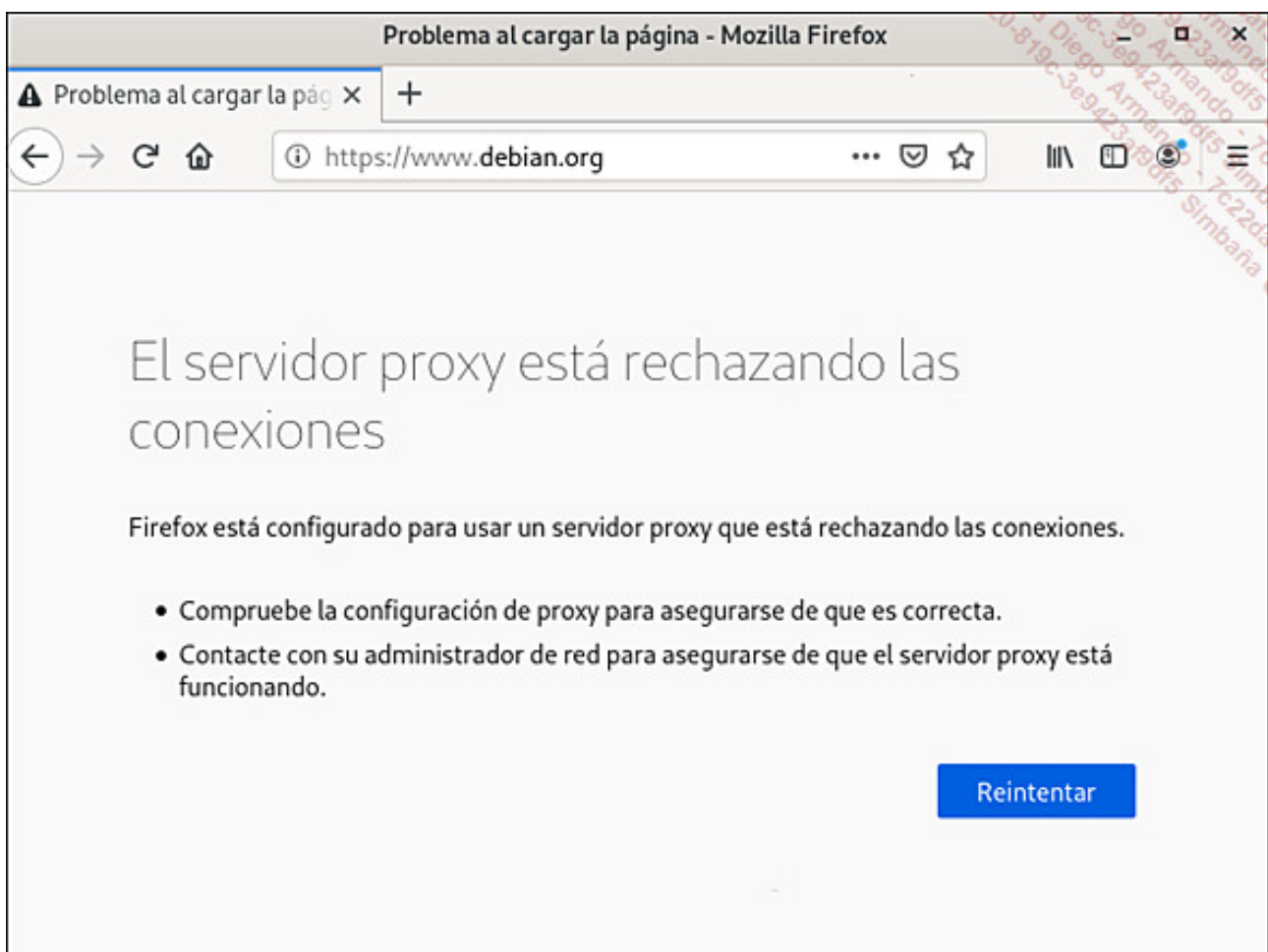
Paramos el servidor Squid:

```

root@debian10:~# systemctl stop squid
root@debian10:~# ps -ef | grep squid
root  7027  6693  0 17:05 pts/0    00:00:00 grep squid

```

Hacemos una prueba con el navegador:



El navegador no puede acceder a Internet.

3. Servidor HTTP Nginx

Configuramos un servidor HTTP Nginx en una distribución de tipo Debian. Tiene que poder gestionar un host virtual, `gestion`.

El nombre de host se puede definir como un alias DNS de la máquina host o declarar en los archivos `hosts` del servidor y de los clientes de test.

Comandos y archivos útiles

- ✓ `apt-get`
- ✓ `systemctl`
- ✓ `nginx`
- ✓ `host`
- ✓ `/etc/nginx/sites-available`
- ✓ `/etc/nginx/sites-enabled/`
- ✓ `wget`
- ✓ `firefox`

Etapas

1. Compruebe que el servidor HTTP Apache no está activo. Instale, si fuera necesario, el paquete del servidor Nginx.
2. Declare el nombre de host `gestion` (a través de DNS o en el archivo `/etc/hosts`).
3. Configure el servidor HTTP Nginx con el host virtual `gestion`.
4. Cree el directorio de datos del host virtual, con una página HTML de prueba. Asócielos a la cuenta de usuario y al grupo del servicio del servidor Nginx.
5. Compruebe y cargue la configuración.
6. Compruebe el acceso al host virtual `gestion`, desde la línea de comandos o un navegador local.
7. Compruebe el acceso al host virtual `gestion`, desde un navegador remoto.

Resumen de los comandos y resultado en pantalla

1. Compruebe que el servidor HTTP Apache no está activo. Instale, si fuera necesario, el paquete de software del servidor Nginx.

Paramos el servidor HTTP Apache:

```
root@debian10:~# systemctl stop apache2
```

Miramos si el paquete de software `nginx` está instalado:

```
root@debian10:~# dpkg -l nginx
```

```

Deseado=desconocido(U)/Instalar/eliminaR/Purgar/retener(H)
| Estado=No/Inst/ficheros-Conf/desempaquetado/medio-conf/medio-inst(H)/
espera-disparo(W)/pendiente-disparo
|/ Err?=(ninguno)/requiere-Reinst (Estado,Err: mayúsc.=malo)
||/ Nombre      Versión    Arquitectura Descripción
+++-----
un nginx        <ninguna>  <ninguna>    (no hay ninguna descripción
disponible)

```

El paquete no está instalado. Procedemos a su instalación:

```

root@debian10:~# apt-get install nginx
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
libnginx-mod-http-auth-pam libnginx-mod-http-dav-ext libnginx-mod-http-echo
libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-sub-
filter libnginx-mod-http-upstream-fair
libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream nginx-common
nginx-full
Paquetes sugeridos:
fcgiwrap nginx-doc
Se instalarán los siguientes paquetes NUEVOS:
libnginx-mod-http-auth-pam libnginx-mod-http-dav-ext libnginx-mod-http-echo
libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-sub-
filter libnginx-mod-http-upstream-fair
libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream nginx nginx-
common nginx-full
0 actualizados, 13 nuevos se instalarán, 0 para eliminar y 26 no actualizados.
Se necesita descargar 1.760 kB de archivos.
Se utilizarán 3.295 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
[...]
Configurando nginx-full (1.14.2-2+deb10u4) ...
Configurando nginx (1.14.2-2+deb10u4) ...
Procesando disparadores para man-db (2.8.5-2) ...
Procesando disparadores para systemd (241-7~deb10u8) .....

```

Se comprueba el estado del servidor Nginx:

```
root@debian10:~# systemctl status nginx
nginx.service - A high performance web server and a reverse proxy server
Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset:
enabled)
Active: active (running) since Sat 2020-10-23 17:22:20 CEST; 19min ago
Docs: man:nginx(8)
Main PID: 7458 (nginx)
Tasks: 3 (limit: 4558)
Memory: 4.6M
CGroup: /system.slice/nginx.service
       7458 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
       7459 nginx: worker process
       7460 nginx: worker process

oct 23 17:22:20 debian10 systemd[1]: Starting A high performance web server and a
reverse proxy server...
oct 23 17:22:20 debian10 systemd[1]: Started A high performance web server and a
reverse proxy server.
```

El servidor está configurado en inicio automático y está activo.

2. Declare el nombre de host `gestion` (a través de DNS o en el archivo `/etc/hosts`).

Como ya configuramos un servidor DNS en la máquina `centos8` durante los anteriores trabajos prácticos, vamos a usarlo para declarar un nuevo alias del nombre de host `debian10` en la zona `midns.es`:

```
[root@centos8 ~]# vi /var/named/db.midns.es
; Archivo de zona midns.es.
$TTL 1D ; Duración de vida por defecto 1 día
; Registro y declaración de la zona:
@ IN SOA centos8.midns.es. admin.midns.es. (
2020102302; serial
```

```

6H; refresh
1H; retry
2D; expire
1H); minimum
; Servidores DNS:
@      IN    NS    centos8.midns.es.
@      IN    NS    debian10.midns.es.
; Direcciones (IPv4):
centos8    IN    A    192.168.1.60
debian10   IN    A    192.168.1.70
puesto     IN    A    192.168.1.24
puesto1    IN    A    192.168.1.25
puesto2    IN    A    192.168.1.26
www61      IN    A    192.168.1.61
; Alias:
www        IN    CNAME centos8
rh         IN    CNAME centos8
ftp        IN    CNAME debian10
gestion    IN    CNAME debian10

```

Hemos declarado un registro de tipo `CNAME`, alias del host `debian10.midns.es.`, y aumentado el número de serie del archivo de zona.

Comprobamos el archivo de zona:

```

[root@centos8 ~]# named-checkzone midns.es /var/named/db.midns.es
zone midns.es/IN: loaded serial 2020102302
OK

```

Recargamos la zona:

```

[root@centos8 ~]# rndc reload
server reload successful

```

Comprobamos la resolución de nombres:

```

[root@centos8 ~]# host gestion.midns.es

```

`gestion.midns.es` is an alias for `debian10.midns.es`.
`debian10.midns.es` has address 192.168.1.70

3. Configure el servidor HTTP Nginx con el host virtual `gestion`.

Creamos un archivo de configuración específico para el host virtual, en el directorio previsto para ello:
`/etc/nginx/sites-available`.

Tenemos que declarar el puerto de escucha del servidor (80), su nombre (`gestion.midns.es`), así como el directorio raíz `/usr/share/nginx/html/gestion`.

```
root@debian10:~# vi /etc/nginx/sites-available/gestion
server {
    listen 80;
    server_name gestion.midns.es;
    root /usr/share/nginx/html/gestion;
    index index.html;
}
```

A continuación hay que "activar" el sitio creando un enlace en el directorio: `/etc/nginx/sites-enabled`:

```
root@debian10:~# ln -s /etc/nginx/sites-available/gestion /etc/nginx/
sites-enabled/gestion
root@debian10:~# ls -l /etc/nginx/sites-enabled/gestion
lrwxrwxrwx 1 root root 34 mayo 23 17:56 /etc/nginx/sites-enabled/
gestion -> /etc/nginx/sites-available/gestion
```

4. Cree el directorio de datos del host virtual, con una página HTML de prueba. Asócielos a la cuenta de usuario y al grupo de servicio del servidor Nginx.

Creamos el directorio de datos del host virtual, con una página HTML de prueba:

```
root@debian10:~# mkdir /usr/share/nginx/html/gestion
root@debian10:~# vi /usr/share/nginx/html/gestion/index.html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Servidor intranet de gestión</title>
</head>
<body>
<h1>Bienvenido al sitio gestion.midns.es</h1>
</body>
</html>
```

Cambiamos el propietario y el grupo del directorio y de todos los elementos que se encuentran en él:

```
root@debian10:~# chown -R www-data:www-data /usr/share/nginx/html/gestion
```

Compruebe y cargue la configuración.

Comprobamos la configuración del servidor HTTP Nginx:

```
root@debian10:~# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Recargamos la configuración:

```
root@debian10:~# systemctl reload nginx
```

Compruebe el acceso al host virtual `gestion`, desde la línea de comandos o un navegador local.

Utilizamos el comando `wget` para comprobar el acceso al host virtual.

```
root@debian10:~# wget -O - gestion.midns.es
--2020-05-23 18:08:44-- http://gestion.midns.es/
Resolviendo gestion.midns.es (gestion.midns.es)... 192.168.1.70
Conectando con gestion.midns.es (gestion.midns.es)[192.168.1.70]:80... conectado.
Petición HTTP enviada, esperando respuesta ... 200 OK
Taille:335 [text/html]
Grabando en :« STDOUT »

-          0%[          ] 0 --.-KB/s
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title> Servidor intranet de gestión </title>
</head>
<body>
<h1>Bienvenido al sitio gestion.midns.es</h1>
</body>
</html>
-          100%[=====] 335 --.-KB/s  ds 0s

2020-05-23 8:08:44 (25,4 MB/s) — escritos a stdout [335/335]
```

7. Compruebe el acceso al host virtual `gestion`, desde un navegador remoto.

Usamos el navegador Firefox de la máquina de prueba de la red local, configurado como cliente DNS del servidor DNS `centos8`. El navegador está configurado sin proxy HTTP.

Desde el navegador Firefox, solicitamos la URL `gestion.midns.es` :



El host virtual está operativo.