

PRÁCTICA 2.1: Servidores Web virtuales con Nginx



Índice:

PREGUNTAS:	3
PRÁCTICA:	3

PREGUNTAS:

- Indica los 3 modos que permite Apache/Nginx de crear Servidores Web Virtuales.
 - IP
 - PUERTO
 - NOMBRE
- En el fichero de configuración de un site:
 - ¿Con qué directiva se da nombre a cada Servidor Web Virtual diferente?
Server_name
 - ¿Con qué directiva se indica dónde están alojados los contenidos a servir por cada Host Virtual? En Apache es "DocumentRoot" y en Nginx es **Root**

PRÁCTICA:

- Instala dos servidores web virtuales accesibles por distinto nombre (igual IP y puerto, pero distinto nombre).

- Un servidor web : `www.[tuapellido]DAW2.org`

- Otro: `www.[tuapellido2]DAW2.org`

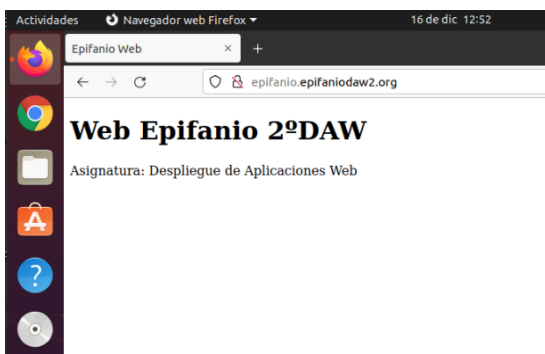
`epifanio.epifaniodaw2.org` y `lopez.epifaniodaw2.org`

Cada uno debe tener un index diferente que muestre en qué site estás.

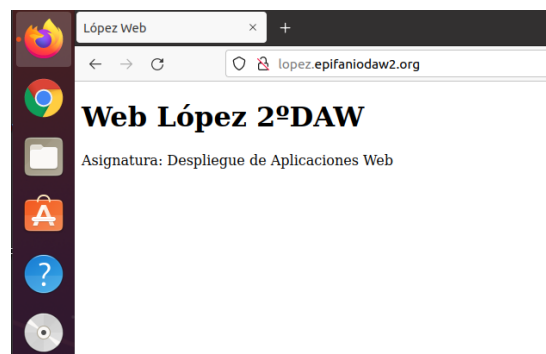
- Captura pantalla mostrando que la sintaxis es OK

```
javiepi@ubusrv-jepi:/var/www/html/epifanio$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
javiepi@ubusrv-jepi:/var/www/html/epifanio$ _
```

- Capturas de pantalla funcionando; prueba desde el server y después desde el cliente. Haz la captura accediendo a cada site desde un cliente



```
javiepi@ubusrv-jepi:~$ ping epifanio.epifaniodaw2.org
PING epifanio.epifaniodaw2.org (192.168.1.1) 56(84) bytes of data:
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=2 ttl=64 time=0.091 ms
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=3 ttl=64 time=0.110 ms
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=4 ttl=64 time=0.039 ms
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=5 ttl=64 time=0.092 ms
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=6 ttl=64 time=0.092 ms
```



```
javiepi@ubusrv-jepi:~$ ping lopez.epifaniodaw2.org
PING lopez.epifaniodaw2.org (192.168.1.1) 56(84) bytes of data:
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=1 ttl=64 time=0.019 ms
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=2 ttl=64 time=0.097 ms
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=3 ttl=64 time=0.061 ms
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=4 ttl=64 time=0.074 ms
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=5 ttl=64 time=0.043 ms
64 bytes from epifanioDAW2.org (192.168.1.1): icmp_seq=6 ttl=64 time=0.089 ms
```

2. En tu site "tuapellido1" crea un index.html con varios enlaces: a varios directorios (mínimo crea dir1 y dir2), a varios files (dir1/file1_1.html y dir1/file1_2.html).
pruebas de accesibilidad de los siguientes apartados.
Haz captura de los directorios de archivos.

Esta es mi página principal



Aquí he creado dos dir, dir1 y dir2, donde dentro de cada uno he creado un index.html

Dir1/file1_1.html



dir2/file1_2.html



3. Opciones de "location" en Nginx (/Directory en Apache, Options).

3.1. Directiva "autoindex on" en location. Para el primer site tuapellido1 haz los siguientes cambios :

```
location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
    autoindex on;
}
```

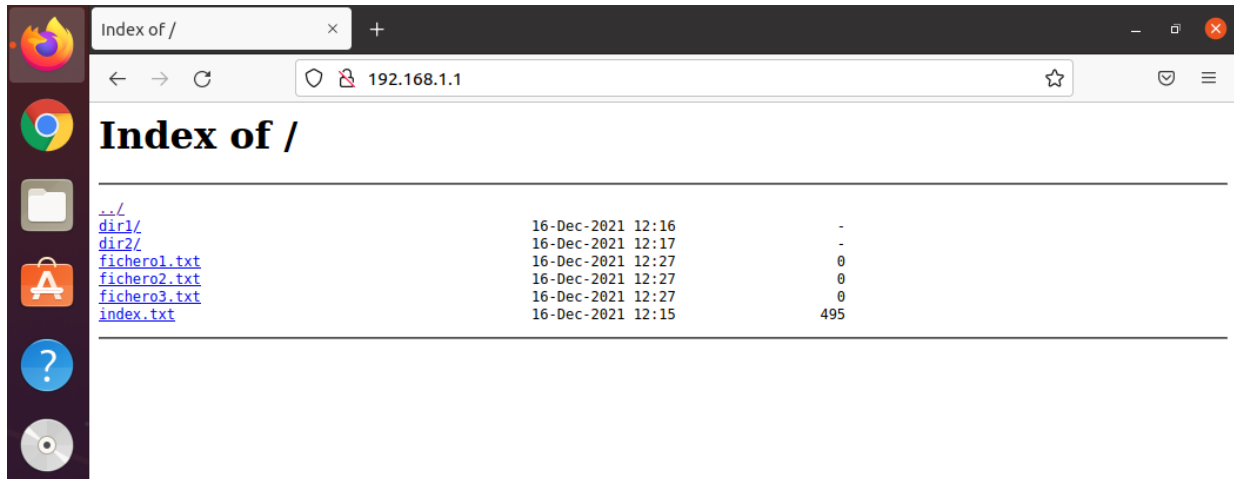
a) Borramos el fichero index.html -> #mv index.html index.txt

```
javiepi@ubusrv-jepi:/var/www/html/epifanio$ sudo mv index.html index.txt
javiepi@ubusrv-jepi:/var/www/html/epifanio$ ls -l
total 12
drwxr-xr-x 2 root root 4096 dic 16 12:16 dir1
drwxr-xr-x 2 root root 4096 dic 16 12:17 dir2
```

b) Creamos varios ficheros *.txt y directorios -> # touch fichero1.txt fichero2.txt fichero3.txt#mkdir dir1 dir2

```
javiepi@ubusrv-jepi:/var/www/html/epifanio$ sudo touch fichero1.txt
javiepi@ubusrv-jepi:/var/www/html/epifanio$ sudo touch fichero2.txt
javiepi@ubusrv-jepi:/var/www/html/epifanio$ sudo touch fichero3.txt
```

c) Accedemos desde el navegador. ¿Qué se muestra? Debería verse el índice de directorios y archivos.

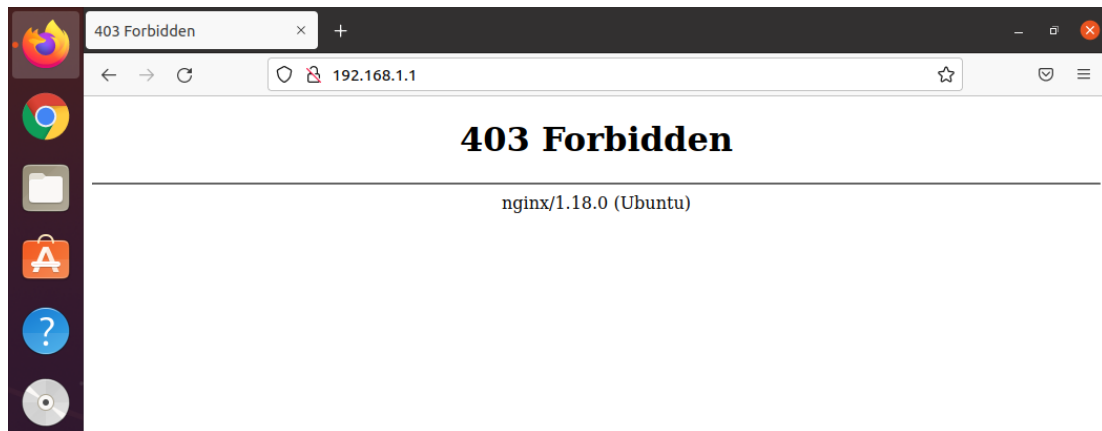


d) Quitamos la opción de índice "autoindex off"

```
location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
    #autoindex off;
}
```

e) Y volvemos a comprobar. ¿Qué se muestra? Justifícalo

Se muestra el error 403 que quiere decir que hemos logrado acceder al servidor, pero no tenemos permisos para editarlo ni verlo, por lo que nos prohíbe acceder a él o ver su contenido desde la maquina cliente.



4. Prohibiendo acceso a parte de tu web. (para hacer el Jueves en clase)

a) Prueba a que sólo desde una IP concreta se pueda acceder al directorio dir1 (en este caso del propio server y no del cliente)

```
location /dir1/ {  
    allow localhost;  
    deny all;  
}
```

Para probar indica en tu navegador la URL/dir1

- Activa el módulo de Nginx que sea necesario

b) Recuerda que el alcance las directivas depende de su ubicación: ¿Qué pasaría si lo hiciese bajo location /? Pruébalo