

CREACIÓN DE SERVIDORES VIRTUALES

El término VirtualHost (Hosting Virtual) se refiere a hacer funcionar más de un sitio web en una sola máquina. Los sitios web virtuales pueden estar basados en direcciones IP, lo que significa que cada sitio web tiene una dirección IP diferente, o basados en nombres diferentes, lo que significa que con una sola dirección IP están funcionando sitios web con diferentes nombres de dominios. El hecho de que estén funcionando en la misma máquina física pasa completamente desapercibido para el usuario, que no se da cuenta de que el sitio que visita está conviviendo con otros sitios.

Directiva <VirtualHost>

Descripción:	Contiene directivas que solo se aplican al hostname or IP address especificado
Syntax:	<VirtualHost <i>addr[:port]</i> [<i>addr[:port]</i>] ...> ... </VirtualHost>
Contexto:	Configuración del servidor
Status:	Core
Module:	Core

Las etiquetas <VirtualHost> </VirtualHost> se usan para englobar un grupo de directivas que serán aplicadas sólo a un virtual host particular. Cuando el servidor recibe una petición de un documento en un servidor virtual particular, usa las directivas de configuración encerradas entre esas etiquetas.

Addr puede ser:

- La dirección IP del host virtual;
- Un nombre de dominio completo para la dirección IP del host virtual;
- El carácter *, el cual puede usarse en combinación con NameVirtualHost * para que equivalga a todas las direcciones IP; o
- La cadena de caracteres _default_, que se usa solo con hosting virtual IP para detectar direcciones IP sin emparejar.

Ejemplo

```
<VirtualHost *:80>

    ServerAdmin webmaster@despliegue.daw01.net

    DocumentRoot "C:/apache/htdocs/despliegue"

    ServerName despliegue.daw01.net

    ServerAlias www.despliegue.daw01.net

    ErrorLog logs/despliegue-error_log.log
```

```
CustomLog logs/despliegue-access_log.log common  
</VirtualHost>
```

Cada host virtual se corresponde con una dirección IP diferente, un número de puerto diferente o un nombre de host diferente para el servidor, en el primer caso la máquina del servidor debe estar configurada para aceptar paquetes IP para múltiples direcciones. (Si la máquina no tiene múltiples interfaces de red, entonces esto puede conseguirse con el comando `ifconfig alias` -- si su sistema operativo lo soporta).

Nota

El uso de `<VirtualHost>` no afecta a las direcciones en las que escucha Apache. Puede que necesite asegurarse de que Apache está escuchando en la dirección correcta usando [Listen](#). Cuando se usa hosting virtual basado en IP, puede especificarse el nombre especial `_default_`, en cuyo caso, este host virtual equivaldrá a cualquier dirección IP que no esté específicamente listada en otro host virtual. En ausencia de un host virtual `_default_` el server config "principal", consistente en todas las definiciones fuera de una sección `VirtualHost`, se usa cuando la IP no coincide con ninguna. (Pero tenga en cuenta que cualquier dirección IP que equivalga a la directiva [NameVirtualHost](#) no usará ni el server config "principal" ni el host virtual `_default_` virtual host.

Puede especificar `:port` para cambiar el puerto de equivalencia. Si no especifica ninguno, entonces por defecto se usa el mismo puerto de la directiva [Listen](#) mas reciente del servidor principal. También puede especificar `*` para hacer coincidir con todos los puertos en esa dirección. (Esto se recomienda cuando se usa con `_default_`.)

Seguridad

Consulte la documentación de [consejos de seguridad](#) para obtener más información sobre por qué pone en riesgo la seguridad si en el directorio donde almacena los archivos log tiene permisos de escritura alguien que no sea el usuario que inicia el servidor.

→ Hay dos tipos diferentes de *host* virtuales:

1. **Basados en IP:** donde una sola instancia del servidor tiene múltiples direcciones IP asignadas y un nombre de dominio asociado se asocia a cada una de esas IPs. Por tanto, cada web tiene una IP diferente, aunque están servidas por un único servidor.
2. **Basados en el nombre:** donde el servidor tiene una única dirección IP asociada y varios nombres de dominio que se asocian con esa única dirección IP. Los diferentes sitios *web* se distinguen en este caso por ese nombre de *host*, dato que es el que usa *Apache* para servir las peticiones correctamente. El *virtual hosting* basado en nombre necesita que se pase un nombre de dominio en cada petición de la página. El servidor ve ese nombre y sirve la página que encaje con ese nombre de dominio. Para que esto funcione, el nombre de dominio debe estar registrado con una entrada válida en el servidor *DNS*.

El siguiente ejemplo muestra dos dominios diferentes (ejemplo.org y ejemplo.net), que se alojaran en el mismo servidor. Ambos *hosts* están usando un `DocumentRoot` (directorio base donde se copian los contenidos de la *web*), *logs* de acceso y de error separados.

```
#Virtual Host 1 – ejemplo.org
```

```

<VirtualHost *:80>
ServerName www.ejemplo.org
ServerAdmin admin@ejemplo.org
DocumentRoot /var/www/html/www.example.org
CustomLog /var/log/apache2/www.example.org-access_log

ErrorLog /var/log/apache2/www.example.org-error_log
</VirtualHost>

```

#Virtual Host 2 – ejemplo.net

```

<VirtualHost *:80>
ServerName www.ejemplo.net
ServerAdmin admin@ejemplo.net
DocumentRoot /var/www/www.example.net
CustomLog /var/log/apache2/www.example.net-access_log

ErrorLog /var/log/apache2/www.example.net-error_log
</VirtualHost>

```

La siguiente imagen muestra la configuración del *host* virtual por defecto que aparece en una instalación estándar de *Apache*:

```

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    <Directory /var/www/html>
        Options Indexes
        AllowOverride None
        Require all granted
    </Directory>
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

```

Configuración de *host* virtuales

Ejemplo de cómo configurar un *host* virtual, asociado a la IP 192.168.1.17 Se asume que la máquina tiene una interfaz de red con esta IP asignada.

- Ir a `/etc/apache2/sites-available`
- Crear un archivo `site2` con el siguiente contenido. En él veremos cómo el *sitio* tendrá sus propios ficheros de *logs*, de acceso y errores para no mezclar la información de ambos sitios (algo que se recomienda). Se supone además que previamente habremos creado el directorio `/var/www/html/site2` (DocumentRoot) y que se habrán movido los archivos de la *web* a servir al mismo).

```
<VirtualHost 192.168.56.3>
ServerName Site2
ServerAdmin webmaster@site2
DocumentRoot /var/www/html/site2
ErrorLog /var/log/apache2/site2-error_log
CustomLog /var/log/apache2/site2-access_log

<Directory /var/www/html/site2>

Options Indexes

AllowOverride None

Require all granted

</Directory>
</VirtualHost>
```

- Ejecutar el comando `a2ensite site2` para habilitar el *host* virtual.
- Reiniciar *Apache*. Debería existir una entrada correspondiente a *site2* en el directorio `sites-enabled` de *Apache*, tal y como se muestra en la siguiente imagen:

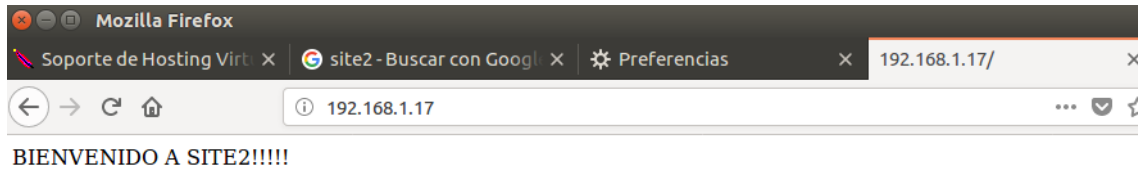
```
drwxr-xr-x 2 root root 4096 dic 12 21:24 .
drwxr-xr-x 8 root root 4096 dic 12 21:09 ..
lrwxrwxrwx 1 root root 35 nov 16 22:47 000-default.conf -> ../sites-available/000-default.conf
lrwxrwxrwx 1 root root 29 dic 12 21:24 site2.conf -> ../sites-available/site2.conf
alumno@ubuntuServer:/etc/apache2/sites-enabled$
```

Conviene comprobar además que los *logs* del nuevo *sitio* existen y tienen contenido (una vez que se hagan peticiones a la *web*), para confirmar así que está sirviéndose de forma correcta. Esto requiere comprobar el contenido de `/var/log/apache2`.

```
alumno@ubuntuServer:/var/log/apache2$ ls -la
total 20
drwxr-x--- 2 root adm 4096 dic 12 21:25 .
drwxrwxr-x 9 root syslog 4096 nov 16 22:31 ..
-rw-r----- 1 root adm 1299 nov 16 22:54 access.log
-rw-r----- 1 root adm 3780 dic 12 21:25 error.log
-rw-r----- 1 root adm 2750 nov 16 22:47 other_vhosts_access.log
-rw-r--r-- 1 root root 0 dic 12 21:25 site2_access_log
-rw-r--r-- 1 root root 0 dic 12 21:25 site2_error_log
alumno@ubuntuServer:/var/log/apache2$
```

Ejecutar `a2dissite 000-default.conf` para deshabilitar el servidor virtual por defecto.

- Acceder al nuevo virtual *host* haciendo uso de su *IP* con `http://192.168.1.17`.



A este nuevo *virtual host* podemos aplicarle configuraciones. Es importante recordar que podemos elegir entre hacer que todos los *host* virtuales tengan una configuración común o bien hacer que cada uno tenga la suya particular.

Finalmente, es importante ver cómo se puede lograr tener varias *IPs* asignadas a una misma máquina para que de esta forma se puedan tener varios *host* virtuales diferenciados por *IP*. Fundamentalmente existen dos formas:

1. Teniendo varias interfaces de red físicas asignadas a un servidor o bien un interfaz que admita la posibilidad de tener múltiples *IPs* asignadas. En caso de *VirtualBox*, podemos crear hasta 4 interfaces de red, algunos de los cuales pueden servir para este propósito.
2. Creando interfaces de red "falsos" que sean "alias" de interfaces de red reales. Estos interfaces de red falsos funcionarán como los reales y les podremos asignar nuevas *IPs* que tengamos disponibles, sirviendo para éste propósito. Existen dos formas de hacerlo:
 - De manera temporal, con un comando como el siguiente: `sudo ifconfig eth0:0 192.168.56.3 up`
 - De manera permanente, modificando el fichero `/etc/network/interfaces`.

```
GNU nano 2.2.2      File: interfaces      Modified
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth1
iface eth1 inet dhcp

auto eth2
iface eth2 inet static
address 192.168.56.2
netmask 255.255.255.0
network 192.168.56.0

#alias de eth2
auto eth2:0
iface eth2:0 inet static
name Ethernet alias LAN card
address 192.168.56.3
netmask 255.255.255.0
broadcast 192.168.56.255
network 192.168.56.0

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

En los DNS públicos o privados deben estar registrados los dominios y/o alias que apuntarán a la IP del servidor.

Enlaces

Apache: <http://httpd.apache.org/docs/2.0/vhosts/>
<http://httpd.apache.org/docs/2.0/mod/core.html#virtualhost>

- <http://www.control-escape.com/web/configuring-apache2-debian.html>
- http://www.brennan.id.au/13-Apache_Web_Server.html#users
- <http://www.ubuntugeek.com/howto-create-name-based-and-ip-based-virtual-hosts-in-apache.html>
- <http://articles.slicehost.com/2008/4/29/ubuntu-hardy-apache-virtual-hosts-1>
- <http://www.petersblog.org/node/840>
- <http://httpd.apache.org/docs/2.2/mod/core.html#virtualhost>
- <http://www.debianhelp.co.uk/virtualhosts.htm>
- <http://www.networkcomputing.com/unixworld/tutorial/017.html#05>
- <http://www.cyberciti.biz/tips/ubuntu-linux-creating-ethernet-alias-for-eth0-network-device.html>
- <https://httpd.apache.org/docs/2.0/es/vhosts/name-based.html>