

¿Cómo configurar Virtual Host de Apache en Ubuntu 14.04 LTS?

Dec 3, 2014 Apache, Getting Started, Linux Basics Ubuntu

Introducción

El servidor web de Apache es uno de los más populares para proveer contenido web en Internet. Cuenta con más de la mitad de todos los sitios web activos en la red y es extremadamente poderoso y flexible.

Apache divide su funcionalidad y componentes en unidades independientes que pueden ser configuradas independientemente. La unidad básica que describe un sitio individual o el dominio llamado `virtual host`.

Estas asignaciones permiten al administrador utilizar un servidor para alojar varios dominios o sitios en una simple interface o IP utilizando un mecanismo de coincidencias. Esto es relevante para cualquiera que busque alojamiento para más de un sitio en un solo VPS.

Cada dominio que es configurado apuntará al visitante a una carpeta específica que contiene la información del sitio, nunca indicará que el mismo servidor es responsable de otros sitios. Este esquema es expandible sin limites de software tanto como el servidor pueda soportar la carga.

En esta guía, te diremos como puedes configurar tus virtual hosts de Apache en tu VPS con Ubuntu 14.04. Durante este proceso, tu aprenderás como configurar diferente contenido para diferentes visitantes dependiendo del dominio que soliciten.

Pre-Requisitos

Antes de empezar este tutorial, deberías crear un usuario no-root siguiendo los pasos del 1 al 4 en esa guía.

Además necesitas tener instalado Apache para poder continuar los siguientes pasos. Si no lo has hecho aún, puedes instalar Apache en tu servidor mediante `apt-get`:

```
sudo apt-get update
sudo apt-get install apache2
```

Después de completar estos pasos, podemos espezar.

Para propósitos de ésta guía, mi configuración creará un virtual host para `ejemplo.com` y otro para `pruebas.com`. Se hará referencia a ellos en esta guía, pero tu deberías sustituirlos por tus propios dominios durante el proceso.

Para aprender como configurar tus dominios con DigitalOcean, sigue ese enlace. Si no tienes dominios disponibles para utilizar, puedes usar valores cualquiera.

Te mostraremos como editar tus archivos locales posteriormente, en la prueba de configuración si estás utilizando valores cualquiera. Esto te permitirá probar tu configuración desde casa, aún si tu contenido no está disponible a través del dominio para otros visitantes.

Paso Uno - Crear la Estructura del Directorio

El primer paso que necesitamos es crear la estructura de directorios que mantendrán la información de nuestro sitio.

Nuestro documento `raíz` (el directorio principal en el cual Apache busca el contenido para mostrar) será configurado en directorios individuales dentro de la

ruta `/var/www`. Crearemos los directorios aquí para los dos virtual hosts que pretendemos configurar.

Dentro de cada uno de estos directorios crearemos un directorio denominado `public_html` el cual mantendrá la información pública del sitio y sus respectivos archivos. Esto nos dará más flexibilidad en nuestro alojamiento.

Para asegurarnos, para cada uno de nuestros sitios, vamos a crear los directorios así:

For instance, for our sites, we're going to make our directories like this:

```
sudo mkdir -p /var/www/ejemplo.com/public_html
sudo mkdir -p /var/www/pruebas.com/public_html
```

Las marcas en rojo representan el dominio que esperamos que sirva nuestro VPS.

Paso Dos - Otorgar Permisos

Ahora tenemos la estructura de los directorios para nuestros archivos, pero el usuario `root` es el propietario de ellos. Si queremos que nuestro usuario regular pueda modificar los archivos en nuestro directorio web, necesitamos cambiar el propietario haciendo lo siguiente:

```
sudo chown -R $USER:$USER /var/www/ejemplo.com/public_html
sudo chown -R $USER:$USER /var/www/pruebas.com/public_html
```

La variable `$USER` tomará el valor del usuario con el cual actualmente estás identificado. Al hacer esto, nuestro usuario regular ahora es propietario de los directorios `public_html` donde se almacenará nuestro contenido.

Debemos además modificar los permisos un poco para asegurarnos que el permiso de lectura pueda ser aplicado a archivos y directorios para que las páginas puedan ser desplegadas correctamente:

```
sudo chmod -R 755 /var/www
```

Tu servidor ahora tiene los permisos necesarios para mostrar el contenido, y el usuario deberá ser capaz de crear contenido en los directorios a medida que sea necesario.

Paso Tres — Crear una Página de Prueba para cada Virtual Host

Actualmente tenemos la estructura en su lugar. Vamos a crear contenido para mostrar.

Solo vamos a hacer una demostración, así que nuestras páginas serán muy simples. Solo crearemos un archivo `index.html` para cada sitio.

Empecemos con `ejemplo.com`. Podemos abrir un archivo `index.html` mediante un editor escribiendo:

```
nano /var/www/ejemplo.com/public_html/index.html
```

En este archivo, crea un documento HTML simple que indicara que el sitio está conectado. Mi archivo quedó así:

```
<html>
  <head>
    <title>Bienvenido a Ejemplo.com!</title>
  </head>
  <body>
    <h1>Éxito! El Virtual Host ejemplo.com esta funcionando!</h1>
  </body>
</html>
```

Guarda y cierra el archivo cuando termines.

Podemos copiar este archivo y usarlo de base para nuestro segundo sitio escribiendo:

```
cp /var/www/ejemplo.com/public_html/index.html
/var/www/pruebas.com/public_html/index.html
```

Ahora podemos abrir el archivo y modificar la información relevante:

```
nano /var/www/pruebas.com/public_html/index.html
<html>
  <head>
    <title>Bienvenido a Pruebas.com!</title>
  </head>
  <body>
    <h1>Éxito! El Virtual Host pruebas.com esta funcionando!</h1>
  </body>
</html>
```

Guarda y cierra como el en caso anterior. Ahora tienes páginas suficientes para probar tu configuración.

Paso Cuatro — Crear Nuevos Archivos Virtual Host

Los archivos Virtual Host son archivos que contienen información y configuración específica para el dominio y que le indican al servidor Apache como responden a las peticiones de varios dominios.

Apache incluye un archivo Virtual Host por defecto denominado `000-default.conf` que podemos usar para saltarnos al punto. Realizaremos una copia para trabajar sobre ella y crear nuestro Virtual Host para cada dominio.

Iniciaremos con un dominio, configuralo, copialo para el segundo dominio, y después realiza los ajustes necesarios. La configuración por defecto de Ubuntu requiere que cada archivo de configuración de Virtual Host termine en `.conf`.

Crear el Archivo Virtual Host

Empezando por copiar el archivo para el primer dominio:

```
sudo cp /etc/apache2/sites-available/000-default.conf
/etc/apache2/sites-available/ejemplo.com.conf
```

Abre el nuevo archivo con tu editor como usuario root:

```
sudo nano /etc/apache2/sites-available/ejemplo.com.conf
```

Este archivo se verá algo como esto (he removido los comentarios aquí para hacer el archivo más legible):

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Como puedes ver, no hay mucho aquí. Personalizaremos los datos aquí para nuestro primer dominio y agregaremos algunas directivas adicionales. Esta sección del Virtual Host coincide cualquier petición que es solicitada al puerto 80, el puerto por defecto de HTTP.

Primero, necesitamos cambiar la directiva `ServerAdmin` por un correo del administrador del sitio que pueda recibir correos.

```
ServerAdmin admin@ejemplo.com
```

Después de esto, necesitamos agregar dos directivas. La primera llamada `ServerName`, que establece la base del dominio que debe coincidir para este Virtual Host. Esto será como tu dominio. La segunda, llamada `ServerAlias`, determina nombres futuros que pueden coincidir y servirse como el nombre base o dominio principal. Esto es útil para host tipo `www`:

```
ServerName ejemplo.com
ServerAlias www.ejemplo.com
```

Lo que resta por cambiar para la configuración básica de un Virtual Host es la ubicación del directorio raíz para el dominio. Ya hemos creado lo que necesitamos, así que solo necesitamos modificar `DocumentRoot` para apuntarlo al directorio que hemos creado:

```
DocumentRoot /var/www/ejemplo.com/public_html
```

En total, nuestro archivo de Virtual Host debe verse así:

```
<VirtualHost *:80>
    ServerAdmin admin@ejemplo.com
    ServerName ejemplo.com
    ServerAlias www.ejemplo.com
    DocumentRoot /var/www/ejemplo.com/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Guarda y cierra el archivo.

Copia el primer Archivo Virtual Host y cambialo para el Segundo Dominio

Ahora que tenemos nuestro primer archivo Virtual Host configurado, podemos crear el segundo copiando el primero y realizando los cambios necesarios.

Empecemos por copiarlo:

```
sudo cp /etc/apache2/sites-available/ejemplo.com.conf
/etc/apache2/sites-available/pruebas.com.conf
```

Abre el nuevo archivo con privilegios root en tu editor:

```
sudo nano /etc/apache2/sites-available/pruebas.com.conf
```

Ahora tenemos que modificar todas las piezas de información para referirnos al segundo dominio. Cuando hayas terminado, deberá verse algo así:

```
<VirtualHost *:80>
    ServerAdmin admin@pruebas.com
    ServerName pruebas.com
    ServerAlias www.pruebas.com
    DocumentRoot /var/www/pruebas.com/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Guarda y cierra al concluir.

Paso Cinco — Habilita los nuevos Archivos Virtual Host

Ahora que hemos creado nuestros archivos virtual hosts, debemos habilitarlos. Apache incluye herramientas que nos permiten hacer esto.

Podemos usar la herramienta `a2ensite` para habilitar cada uno de nuestros sitios haciendo esto:

```
sudo a2ensite ejemplo.com.conf
sudo a2ensite pruebas.com.conf
```

Cuando hayas concluido, deberás reiniciar Apache para asegurarte de que tus cambios surtan efecto:

```
sudo service apache2 restart
```

Deberás recibir un mensaje de información similar a esto:

```
* Restarting web server apache2
AH00558: apache2: Could not reliably determine the server's fully
qualified domain name, using 127.0.0.1. Set the 'ServerName' directive
globally to suppress this message
```

Este mensaje no afecta nuestro sitio.

Paso Seis — Configura Archivos Locales (Opcional)

Si aún no estás utilizando nombres de dominio de tu propiedad para este procedimiento y utilizaste dominios ejemplo en su lugar, puedes al menos hacer pruebas de funcionalidad de este proceso modificando temporalmente el archivo `hosts` de tu computadora local.

Esto interceptará cualquier petición a los dominios que configures y apunten a tu VPS, solo si estas utilizando dominios registrados.

Esto solo funciona a través de tu computadora, y es simplemente útil para propósitos de prueba.

Asegúrate de estar trabajando en tu computadora local para los siguientes pasos y no en tu VPS. Deberás conocer la contraseña del administrador o ser miembro del grupo administrativo.

Si estas en una Mac o una computadora con Linux, edita tu archivo local con privilegios de administrador escribiendo:

```
sudo nano /etc/hosts
```

Si estás en una máquina con Windows, puedes buscar las instrucciones para modificar tu archivo hostsaquí.

Los detalles que necesitas agregar son la IP pública de tu VPS seguido del dominio que desees apuntar a ese VPS.

Para los dominios que utilizamos en esta guía, asumiremos que la IP de nuestro VPS es `111.111.111.111`, podemos agregar las siguientes líneas al final del archivo hosts:

```
127.0.0.1    localhost
127.0.1.1    guest-desktop
111.111.111.111 ejemplo.com
111.111.111.111 pruebas.com
```

Esto apuntará directamente cualquier petición para `ejemplo.com` y `pruebas.com` en nuestra computadora y enviarlas a nuestro servidor en `111.111.111.111`. Esto es lo que queremos si no somos propietarios de esos dominios aún, solo con fines de prueba para nuestros Virtual Hosts.

Guarda y cierra el archivo.

Paso Siete — Prueba tus Resultados

Ahora que tenemos nuestros Virtual Hosts configurados, podemos realizar pruebas de configuración simplemente visitando el dominio que hemos configurado mediante nuestro navegador web:

`http://ejemplo.com`

Deberás ver algo como esto:

Success! The example.com virtual host is working!

Del mismo modo, si visitamos la segunda página:

`http://pruebas.com`

Podrás observar el archivo que has creado para el segundo sitio:

Success! The example.com virtual host is working!

Si en ambos sitios funciona bien, entonces has configurado correctamente **dos** Virtual Hosts en el mismo servidor.

Si necesitas ajustar el archivo hosts de tu computadora, probablemente solo deberás borrar las líneas que has agregado y verificar que tu configuración funciona. Esto previene que tu archivo hosts se llene de entradas que no son necesarias.

Si necesitas acceso constante a estos Virtual Host, considera adquirir dominios para cada sitio y configurarlos para que apunten a tu VPS.

Conclusión

Si me has seguido, deberás tener un servidor respondiendo a dos dominios separados. Ahora puedes expandir este procedimiento siguiendo los pasos que hemos llenado arriba para crear Virtual Hosts adicionales.

No hay limite de software en el número de dominios que Apache pueda manejar, así que eres libre de agregar tantos como tu servidor pueda soportar.