

 Subscribe

¿Cómo configurar Virtual Hosts de Apache en Ubuntu 16.04?


14Posted December 12, 2016  140.3k APACHE UBUNTU 16.04

By: Brennen Bearnes

Introducción

Apache es el servidor web más popular para servir contenido en Internet. Cuenta con más de la mitad de los sitios activos en Internet y es extremadamente poderoso y flexible.

Apache rompe su funcionalidad y componentes en unidades separadas que pueden ser personalizadas y configuradas de manera independiente. La unidad básica que describe a un sitio o dominio es denominada `virtual host` (ó alojamiento virtual en español).

Esta designación permite al administrador hacer uso de un servidor para alojar múltiples dominios o sitios en una única interfaz o IP utilizando un mecanismo de coincidencias. Esto es relevante para cualquiera que desee alojar más de un sitio en un mismo VPS.

Cada dominio configurado enviará al visitante a un directorio específico manteniendo la información del mismo, a simple vista nunca indicará que el mismo servidor es también responsable de otros sitios. Este esquema es flexible sin alguna limitación de software siempre y cuando el servidor pueda manejar la carga.

En esta guía, lo encaminaremos a configurar los Virtual Host de Apache en un VPS con Ubuntu 16.04. Durante este proceso, aprenderá como servir contenido diferente a distintos visitantes dependiendo de cual dominio estén solicitando.

Pre-requisitos

Antes de iniciar con este tutorial, debería crear un usuario no-root como se describe en los

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics. 

Sign Up

También necesitará tener Apache instalado para trabajar estos pasos. Si aún no tiene Apache, puede instalarlo en su servidor mediante `apt-get` :

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

Una vez completados dichos pasos, podemos empezar.

Para propósitos de esta guía, nuestra configuración creará un virtual host para `example.com` y otro para `test.com`. Estos serán mencionados a lo largo de la guía, pero usted deberá utilizar sus propios dominios o valores mientras nos sigue.

Para aprender a configurar tus dominios con DigitalOcean siga este enlace. Si aún *no* tiene un dominio disponible para utilizar, puede usar cualquier valor.

Le mostraremos como editar su archivo host local posteriormente para probar la configuración si está utilizando valores aleatorios. Esto le permitirá probar su configuración desde su computadora personal, aún si su contenido no está disponible a través de un dominio a otros visitantes.

Paso Uno — Crear la Estructura del Directorio

El primer paso será crear una estructura de directorios que alojará los datos del sitio que vamos a proporcionar a nuestros visitantes.

Nuestro documento `root` (ó documento raíz, es el directorio más alto en el que Apache buscará contenido para mostrar) será configurado en directorios individuales bajo el directorio `/var/www`. Crearemos un directorio aquí para cada uno de los virtual hosts que pretendemos crear.

Dentro de cada uno de *estos* directorios, crearemos una carpeta `public_html` que mantendrá los archivos. Esto nos dará algo de flexibilidad en nuestro hosting.

Por ejemplo, para nuestros sitios, vamos a crear los directorios así:

```
$ sudo mkdir -p /var/www/example.com/public_html
$ sudo mkdir -p /var/www/test.com/public_html
```

Las partes en rojo representan los nombres de dominio que deseamos servir desde nuestro VPS.

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.



Sign Up

Paso Dos — Otorgar Permisos

Ahora tenemos la estructura de directorios para nuestros archivos, pero son propiedad de nuestro usuario `root`. Si queremos que nuestro usuario regular sea capaz de modificar archivos dentro de nuestros directorios web, debemos cambiar la propiedad haciendo lo siguiente:

```
$ sudo chown -R $USER:$USER /var/www/example.com/public_html
$ sudo chown -R $USER:$USER /var/www/test.com/public_html
```

La variable `$USER` tomará el valor del usuario con el cual estás autenticado actualmente cuando presiones **Enter**. Haciendo esto, nuestro usuario regular ahora será el propietario del directorio `public_html` y sus respectivos sub-directorios donde almacenaremos nuestro contenido.

Debemos además, modificar un poco nuestros permisos para asegurarnos de que el acceso de lectura esté habilitado en el directorio web general y todos los archivos y directorios en él para que todas las páginas puedan ser servidas correctamente:

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

Su servidor web ahora debe tener los permisos que requiere para servir el contenido, y su usuario deberá ser capaz de crear contenido entre las carpetas necesarias.

Paso Tres — Crear Páginas de Prueba para cada Virtual Host

Tenemos nuestra propia estructura de directorios en forma. Vamos a crear algo de contenido para servir.

Vamos a ir con una demostración, así que nuestras páginas serán muy simples. Vamos a crear una página `index.html` para cada sitio.

Vamos a empezar con `example.com`. Podemos abrir un archivo `index.html` en nuestro editor escribiendo:

```
$ nano /var/www/example.com/public_html/index.html
```

En éste archivo, crea un documento HTML simple que indica el sitio al cual está conectado. Mi archivo quedaría así:

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.



Sign Up

```
<html>
  <head>
    <title>¡Bienvenido a Example.com!</title>
  </head>
  <body>
    <h1>¡Lo lograste! El virtual host example.com está funcionando</h1>
  </body>
</html>
```

Guarde y cierre el archivo cuando concluya.

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

```
$ cp /var/www/example.com/public_html/index.html /var/www/test.com/public_html/index.html
```

Podemos entonces abrir el archivo y modificar las partes relevantes de información:

```
$ nano /var/www/test.com/public_html/index.html
```

```
/var/www/example.com/public_html/index.html
```

```
<html>
  <head>
    <title>¡Bienvenido a Test.com!</title>
  </head>
  <body>
    <h1>¡Lo lograste! El virtual host test.com está funcionando</h1>
  </body>
</html>
```

Proceda a guardar y cerrar este archivo. Ahora tiene las páginas necesarias para probar la configuración del virtual host.

Paso Cuatro — Crea un Nuevo Archivo para su Virtual Host

Los archivos virtual host son archivos que especifican la configuración actual de un virtual host e indican como el servidor Apache va a responder a varias solicitudes de dominio.

Apache viene con un archivo virtual host por defecto llamado `000-default.conf` que podemos

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.

✕ a cada uno

Enter your email address

Sign Up

Vamos a iniciar con un dominio, configurarlo, copiarlo para nuestro segundo dominio, y después hacer algunos cambios para ajustes tanto como se requieran. La configuración por defecto de Ubuntu requiere que cada archivo de virtual host termine en `.conf`.

Crea el Primer Archivo Virtual Host

Inicie copiando el archivo para el primer dominio:

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

Abra el nuevo archivo en su editor con permisos de root:

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

El archivo se verá algo así (he removido los comentarios aquí para hacer el contenido más apreciable):

`/etc/apache2/sites-available/example.com.conf`

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

Como podrá apreciar, no hay mucho aquí. Personalizaremos los datos aquí para el primer dominio y agregaremos algunas directivas adicionales. Esta sección del virtual host, coincide *cualquier* solicitud que sea hecha por el puerto 80, el puerto por defecto de HTTP.

Primero, necesitamos cambiar la directiva `ServerAdmin` a un correo electrónico en donde el administrador del sitio pueda recibir correos.

```
ServerAdmin admin@example.com
```

Después de esto, necesitamos *agregar* dos directivas. La primera llamada `ServerName`, que establece el dominio base que debe coincidir para la definición de este virtual host. Esto comúnmente es su dominio. La segunda, llamada `ServerAlias`, define nombres alternativos por los cuales podría ser encontrado como alternativa al dominio base. Esto es útil para definir

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.



Sign Up

```
ServerName example.com
ServerAlias www.example.com
```

El único cambio restante que hay que cambiar para un archivo básico virtual host es la ubicación del documento raíz para este dominio. Ya hemos creado el directorio que necesitamos, así que solo necesitamos alterar la directiva `DocumentRoot` para que refleje el directorio que hemos creado.

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

En totalidad, nuestro archivo virtual host debería verse así:

/etc/apache2/sites-available/example.com.conf

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

Guarde y cierre el archivo.

Copiar el Primer Virtual Host y Configuralo para el Segundo Dominio

Ahora que tenemos nuestro primer archivo virtual host establecido, podemos crear nuestro segundo archivo copiando el primero y ajustándolo como sea necesario.

Inicie copiándolo:

```
sudo cp /etc/apache2/sites-available/example.com.conf /etc/apache2/sites-available/test.com.
```

Abra el nuevo archivo con privilegios root en su editor:

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

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.



Sign Up

Ahora necesitará modificar todas las piezas de información para hacer referencia al segundo dominio. Cuando concluya, se verá algo así:

```
/etc/apache2/sites-available/test.com.conf
```

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

Guarda y cierra el archivo cuando concluyas.

Paso Cinco — Habilitar los Nuevos Archivos Virtual Host

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

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

```
$ sudo a2ensite example.com.conf
$ sudo a2ensite test.com.conf
```

Posteriormente, deshabilite el sitio poder defecto definido en `000-default.conf`:

```
$ sudo a2dissite 000-default.conf
```

Cuando concluyas, deberá reiniciar Apache para hacer que estos cambios sean efectivos:

```
$ sudo systemctl restart apache2
```

En otra documentación, podrá encontrar un ejemplo utilizando el comando `service`:

```
sudo service apache2 restart
```

Este comando aún funciona, pero puede no mostrar la respuesta a la que está acostumbrado a

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.



systemd.

Sign Up

Paso Seis — Configure su Archivo Hosts Local (Opcional)

Si aún no está utilizando un dominio real para probar estos procedimientos y ha utilizado un dominio ejemplo para ello, entonces puede al menos probar la funcionalidad de este proceso modificando temporalmente el archivo `hosts` en su computadora local.

Esto interceptará todas las solicitudes para el dominio que desea configurar y las apuntará a su VPS, como lo hace el sistema DNS con los dominios registrados. Esto funcionará solamente desde su computadora, y es simplemente útil para propósito de pruebas.

Asegúrese de realizar los siguientes pasos en su computadora local y no en su VPS. Para ello, deberá conocer la contraseña administrativa o de lo contrario al menos ser miembro del grupo administrativo.

Si estás en una Mac o una Linux PC, edite su archivo local con privilegios de administrador escribiendo:

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

Si está utilizando una Windows PC, puede encontrar las instrucciones para alterar tu archivo `host` aquí.

Los detalles que necesite agregar son la dirección IP pública de su VPS seguido del dominio que desea utilizar para localizar el VPS.

Para los dominios que he usado en esta guía, asumiendo que la dirección IP de mi VPS es `111.111.111.111`, yo puedo agregar las siguientes líneas en la parte inferior de mi archivo `hosts`:

```
                                /etc/hosts

127.0.0.1    localhost
127.0.1.1    guest-desktop

111.111.111.111 example.com
111.111.111.111 test.com
```

Esto detectará cualquier solicitud para `example.com` y `test.com` en mi computadora y la enviará a mi servidor en `111.111.111.111`. Esto es lo debemos hacer si no utilizamos un dominio real para probar nuestros virtual hosts.

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.



Sign Up

Paso Siete — Pruebe sus Resultados

Ahora que cuenta con sus virtual hosts configurados, puede probar su configuración fácilmente dirigiéndose a los dominios que ha configurado directamente desde su navegador web:

`http://example.com`

Deberá ver una página que luce así (el texto puede variar):

Success! The example.com virtual host is working!

Por otra parte, si visita la segunda página:

`http://test.com`

Podrá visualizar el archivo que ha creado para el segundo sitio (igualmente, el texto puede variar):

Success! The test.com virtual host is working!

Si ambos sitios funcionan bien, entonces ha configurado correctamente **ambos** virtual hosts en el mismo servidor.

Si realizó un ajuste en el archivo hosts en su computadora, entonces debería borrar las líneas que ha agregado ahora que ya ha verificado que la configuración funciona. Esto previene que su archivo hosts se llene de entradas innecesarias.

Si necesita acceso por un periodo largo, considere adquirir un dominio para cada sitio y configurarlo para apuntar a su VPS.

Conclusión

Si me ha seguido hasta aquí, ahora deberá tener un servidor que maneja dos dominios separados. Puede expandir este proceso siguiendo los pasos que indicamos arriba para crear virtual hosts adicionales.

No hay límite de software para el número de dominios que Apache puede manejar, así que siéntase libre de crear tantos dominios como su servidor pueda manejar.

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.



Sign Up

By: Brennen Bearnes

 Upvote (14) SubscribeTranslation:
TonnyORG

Take the Developer Landscape Survey

Tell us how DO and other cloud providers are serving your needs. Help DO improve its offerings for you and other Developers around the world.

[SHARE YOUR FEEDBACK](#)

Related Tutorials

How To Migrate your Apache Configuration from 2.2 to 2.4 Syntax.

How To Get Started With mod_pagespeed with Apache on a CentOS and Fedora Cloud Server

How To Use the .htaccess File

How To Set Up Mod_Rewrite (page 2)

How To Create a Custom 404 Page in Apache

1 Comment

Leave a comment

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.

Enter your email address

Sign Up

Log In to Comment

^ jcarlos17 May 22, 2017



0 Hola. Gracias por el tutorial.

Solo tengo una pregunta, espero me pueda ayudar.

Tengo un droplet con su correspondiente dominio (a modo de ejemplo lo llamaré **misistema.com**).

Y ahora quiero tener un dominio que apunte a una ruta dentro de mi sistema, de forma que cuando alguien entre a **mipagina.com** vea **misistema.com/datos/6**.

Primero que todo pienso apuntar desde **mipagina.com** a la IP del droplet.

A partir de allí creo que debería configurar usando Virtual Hosts.

Pero, veo que en los Virtual Hosts se indica una ruta que existe como carpeta, pero en este caso, se trata de **rutas que existen solo en el sistema, y no existen como carpetas** (el sistema está hecho en Laravel y esta es una de sus rutas).

Muchas gracias de antemano. Por favor, cualquier idea me será de ayuda.



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



Copyright © 2019 DigitalOcean™ Inc.

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.



Enter your email address

Sign Up

[Community](#) [Tutorials](#) [Questions](#) [Projects](#) [Tags](#) [Newsletter](#) [RSS](#) [Distros & One-Click Apps](#) [Terms, Privacy, & Copyright](#) [Security](#) [Report a Bug](#) [Write for DOnations](#) [Shop](#)

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.

[Sign Up](#)