



**Autor:** Andrés Restrepo González  
**Correo electrónico:** handres999@gmail.com

**Fecha de creación :** 08/07/17

**Ultima modificación:** 08/07/17

## Índice de contenido

1.Licencia (BSD).....	1
2.¿Qué es httpd?.....	2
2.1.Historia.....	2
2.2. Característica.....	2
3.Configuración.....	3
3.1.Configuración global.....	3
4.Ejemplos.....	5
5.Conclusiones.....	7
6.Enlaces relacionados.....	7
7.Historial de documentación.....	7

## 1. Licencia (BSD)

Copyright (c) 2017, Andrés Restrepo,

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the **OpenBSD Colombia** nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 2. ¿Qué es httpd?

httpd es un servidor Web creado y mantenido por el proyecto OpenBSD desde el año 2015. Su aparición se dio en la versión de OpenBSD 5.7. Este proyecto busca la simplicidad y seguridad adaptándose a la filosofía manejada por el OBSD.

### 2.1. Historia

OpenBSD incluye un nuevo servidor Web que comenzó apenas dos semanas antes de que se realizara el lanzamiento de OBSD 5.6 fuera terminado, ¿pero por qué se necesitó otro servidor Web?. Antes de la incorporación de httpd, se trabajó 17 años con el servidor Web Apache, el cual para su adopción fue necesaria la modificación de las fuentes. Con el paso del tiempo se optó por el proyecto *nginx*<sup>1</sup>, ya que este prometía mejores alternativas al viejo Apache. Este era un servidor Web moderno y con una licencia adecuada (BSD). A este se le realizaron ajustes en cuanto la seguridad se refiere, pero se repite la historia tal cual como pasó con Apache, una base de código creciente por su auge y una dirección comercial que creó descontento con los desarrolladores que soportaban el proyecto. Fue así que en *OpenBSD's g2k14 Hackathon* en Eslovenia se experimentó con *relayd*, y como resultado; se creó un simple servidor Web con soporte para archivos estáticos, FastCGI y TLS.

Pasaron 16 años para que el servidor Web de OpenBSD basado en Apache 1.3.29 fuera remplazado por un nuevo servidor desarrollado por **Reyk Floeter**. Dado a que el diseño antiguo no coincidía con los estándares modernos en OpenBSD, además la con la adopción de nginx que parecía ser el remplazo perfecto por ser rápido y moderno.

Nginx fue importado en OpenBSD a finales del 2011 y apareció por primera vez en el lanzamiento 5.1 en mayo de 2012. Durante algún tiempo ambos servidores (Apache y nginx) convivieron en OpenBSD al mismo tiempo, ya que nginx no contaba con el soporte clásico de CGI, fue en marzo de 2014 cuando Apache fue finalmente eliminado, ya que no podría ser retirado antes de que hubiera una forma de ejecutar estos *scripts* Con FastCGI bajo nginx.

El servidor httpd fue liberado en la versión 5.6 de OpenBSD, como un servicio completamente funcional pero básico, con soporte FastCGI y TLS, el cual cuenta con aproximadamente 10.000 líneas de código.

La simplicidad también significa que debe ser fácil de usar, por lo que se proporciona un archivo de configuración de lectura humana. Por defecto *httpd.conf* solo necesita tres líneas de configuración:

```
server "www.example.com" {
    listen on * port 80
}
```

Fig.1 - Configuración básica de httpd.

### 2.2. Característica

Las siguientes funcionalidades son soportadas actualmente por el servicio:

- **Archivos estáticos:** El propósito principal es servir archivos estáticos y directorios a través de la auto-indexación opcional. Si la auto indexación está habilitado, los directorios se devuelven como listas de HTML.
- **FastCGI:** Proporciona una interfaz para servir contenido dinámico como una alternativa

1 <https://www.nginx.com/resources/wiki/>

CGI o cualquier otro *script* incrustado o similar al mecanismo CGI. Soporta funciones asíncronas y directas FastCGI a través de un *socket's* UNIX o TCP/IP. Ejecuta *chroot* y cuenta con separación de privilegios por defecto.

- **TLS:** Soporta conexiones seguras a través de TLS Por LibreSSL y la libtls. La pila TLS utiliza estándares de seguridad de última generación y desactiva las las características anticuadas de SSL/TLS.
- **Servidores virtuales:** Configuración flexible con soporte para servidores IP basados en nombres y IP en IPv4 y IPv6.
- **Reconfiguración:** La configuración en ejecución puede ser cambiada y recargada sin interrupción al momento de recargar el servicio.
- **Registro:** Permite el registro por servidor a través de archivos. Como alternativa, se admite el registro a través de *syslog*.
- **Bloqueo:** Usado para impedir el acceso no autorizado a subdirectorios y para redirigir a URL alternas.

### 3. Configuración

El archivo de configuración de *httpd* utiliza un estilo legible y flexible, este se hace usando palabras claves en inglés usando bloques identificados por llaves «{}», respetando la gramática usada por OpenBSD, por ende aplica el uso de los Macros (variables definidas por el usuario, utilizadas para simplificar la configuración).

Dentro de las secciones se puede especificar la dirección IPv4 o IPv6, *FQDN* o hombre de la máquina que luego será resuelta ya sea en el archivo *host* o a través de un *DNS*, en caso de que el servicio escuche por todas las interfaces, bastará con «\*», el cual será usado como alias de 0.0.0.0 para IPv4, así mismo se puede usar «:» para escuchar en todas las interfaces IPv6, tenga presente que se puede hacer uso de un puerto el cual será tomado de */etc/service*.

Una línea puede extenderse sobre varias líneas usando la barra invertida (*backslash*), además podrá hacer uso de comentarios con «#», un ejemplo de configuración sería:

```
#Se tiene la dirección IP 10.0.0.1 guardada en la variable ext_ip
ext_ip="10.0.0.1"

server "default" {
    listen on $ext_ip port 80
}
```

Fig.2 – Uso de macros.

#### 3.1. Configuración global

La configuración global puede ser cambiada, pero esta no será aplicada luego de recargar (*reload*) el servicio:

- **chroot directory:** Si no se especifica, por defecto es el directorio */var/www*, el cual es el directorio del usuario *www*.
- **logdir directory:** Especifica la ruta absoluta del directorio en el que se escribirán los registros, en caso tal no especificarse, estos recaen en dentro del *chroot directory*.
- **prefork number:** Números de procesos ejecutados por el servidor Web, por defecto son 3.

- **alias name:** Se puede especificar un nombre adicional al servidor.
- **[no] authenticate [realm] with htpasswd:** Es usado para autenticar un usuario comprobando las credenciales contra el archivo de autenticación, en este caso *htpasswd*. El nombre del archivo se ubica dentro del *chroot* y debe ser legible por el usuario *www*.
- **block drop:** Eliminar la conexión sin enviar una página de error.
- **block [return code [uri]]:** Cierre la conexión y envía una página de error. Si el código de retorno opcional no especificado, httpd niega el acceso con un «403 Forbidden».
- **connection option:** En esta opción se establecen límites del servidor.
  - **max request body number:** Tamaño en bytes que el cliente puede enviar al servidor, por defecto este valor es de 1 048 576 bytes que equivale a 1M.
  - **max requests number:** Número máximo de solicitudes por conexión HTTP. El número máximo de conexiones es 100.
  - **timeout seconds:** Especifica el tiempo de inactividad en segundos para las sesiones aceptadas. El tiempo de espera predeterminado es de 600 segundos (10 minutos). El máximo es de 2 147 483 647 segundos (68 años).
- **directory option:** acá se establecen las opciones de acceso al directorio, las opciones válidas son:
  - **[no] auto index If:** En caso tal que no se encuentre un archivo válido, el servidor listará el contenido. Esta opción por defecto se encuentra deshabilitada.
  - **index string:** Se establece un archivo predeterminado para su carga, en caso tal que no se especifique, será *index.html*.
  - **no index:** Deshabilita el índice, en este caso no se mostrará ningún archivo.
- **[no] fastcgi [socket socket]:** Habilita FastCGI. El *socket* está dentro del directorio raíz *chroot*.
- **listen on address [tls] port number:** Acá se especifica el puerto y la dirección IP (*socket*) de las conexiones seguras. Esta línea puede indicarse en varios sitios virtuales.
- **location path {...}:** Especifica reglas de configuración de servidor para una ubicación específica para un sitio virtual.
- **[no] log [option]:** El registro está habilitado de forma predeterminada haciendo uso de archivos de registro de errores y acceso, pero se pueden cambiar por servidor. Utilice la directiva *no log* para deshabilitar el registro de cualquier solicitud. Las opciones válidas son:
  - **access name:** Establece el nombre del archivo de registro de acceso en relación con el directorio de registro. Si no se especifica, por defecto es *access.log*.
  - **error name:** Establece el nombre del archivo de registro de acceso en relación con el directorio de registro. Si no se especifica, por defecto es *error.log*.
  - **style style:** Establece el estilo de registro. El estilo puede ser común, combinados o de conexión. Si no se especifica, el valor predeterminado es común.
  - **[no] syslog:** habilita o deshabilita el registro en *syslog* en lugar de los archivos de registro.
- **pass:** deshabilitar cualquier bloque anterior en una ubicación.

- **root option:** en este apartado, se configuran las opciones del documento raíz, las opciones válidas son:
  - **directory:** establece la raíz del documento a cargar. El directorio es un nombre de ruta dentro del *chroot*. Si no se especifica, por defecto es */htdocs*.
  - **strip number:** Cantidad de directorios desde la raíz a la que se podrá acceder.
  
- **tcp option:** habilita o deshabilita el protocolo TCP/IP, las opciones son:
  - **backlog number:** Establezca la longitud máxima cola de conexiones pendientes.
  - **ip minttl number:** Esta puede utilizarse para descartar paquetes con un TTL inferior al valor especificado.
  - **ip ttl number:** Cambia el valor predeterminado de tiempo de vida en los encabezados IP.
  - **[no] nodelay:** Esta opción es usada para evitar retrasos en el flujo de datos.
  - **[no] sack:** Esta opción es usada para las confirmaciones selectivas para la conexión.
  - **socket buffer number:** en caso tal de establecer el tamaño del búfer de nivel de *socket* para la entrada y salida para esta conexión. Esto afectará al tamaño de la ventana TCP.
  
- **tls option:** En este punto se indicarán las opciones TLS siempre y cuando sean habilitado.
  - **certificate file:** Se especifica el certificado que se va a utilizar. El archivo debe contener una codificación PEM.
  - **ciphers string:** Especifique la cadena de cifrado TLS. Si no se especifica, el valor por defecto «*HIGH:!aNULL*».
  - **dhe params:** Se especifican los parámetros DHE que se utilizarán para los conjuntos de cifrado DHE. El valor predeterminado es «*none*», lo que deshabilita los conjuntos de cifrado DHE.
  - **ecdhe curve:** Especifique la curva ECDHE que se utilizará para los conjuntos de cifrado ECDHE. El valor predeterminado es *auto*.
  - **key file:** Se especifica la clave privada que se va a utilizar. El archivo debe contener una codificación PEM.
  - **protocols string:** Especifique los protocolos TLS que se habilitarán para este servidor. Si no se especifica, el valor predeterminado es «*all*» (todos los protocolos disponibles).

## 4. Ejemplos

Luego de ver la posibilidad de realizar diferentes configuraciones en el servicio, es necesario realizar ciertos ajustes para que el servicio opere en nuestra máquina, empezando por habilitar el servicio, además logrando así que vez que la máquina inicia, sea iniciado el mismo:

```
# rcctl enable httpd
```

Fig.3 – Habilitar el servicio.

Nota: En el directorio `/etc/examples` se encuentra un archivo de ejemplo otorgado luego de la instalación habitual del sistema operativo.

```
# vi /etc/httpd.conf

ext_addr=""
server "default" {
    listen on $ext_addr port 80
}

# /etc/rc.d/httpd restart
```

Fig.4 – Configuración básica del servicio.

Como se presenta en la Fig. 4, se crea inicialmente el archivo de configuración del servicio en la ruta `/etc/`, el archivo es `httpd.conf`, luego de esto se adicionan las siguientes líneas:

- **ext\_addr=""**: Variable que indica que se escucharán peticiones en todas las interfaces del servidor.
- **server "default"**: Parámetro por defecto, el cual indica que el servidor local.
- **listen on \$ext\_addr port 80**: En esta línea aparecen dos opciones, las cuales indican que se escucharán por todas las interfaces por el puerto 80.

Para iniciar el servicio, basta con:

```
# /etc/rc.d/httpd restart
httpd(ok)
httpd(ok)
```

Fig.5 – Reinicio del servicio.

Tenga presente que las opciones que otorga el servicio con respecto a la manipulación del estado del servicio, son: *start*, *stop*, *restart*, *reload* y *check*.

El sitio deberá ser creado en la ruta `/var/www/htdocs`.

```
# vi /var/www/htdocs/index.html

<html>
  <title>OpenBSD Colombia</title>
  <body>Mi primer sitio Web en OpenBSD con httpd</body>
</html>
```

Fig.6 – Creación de un sitio de prueba.

Para verificar el estado del servicio, bastará con digitar: `netstat -nat | grep LISTEN | grep 80`:

```
# netstat -nat | grep LISTEN | grep 80
tcp      0      0 *.80          *.*          LISTEN
tcp6     0      0 fe80::1%lo0.25 *.*          LISTEN
```

Fig.7 – Estado de la conexión del puerto 80.

En caso que no se haya realizado un cambio referente a la ubicación de los archivos *log*, estos se encuentran en */var/www/logs*, en donde se encuentra los archivos *access.log* y *error.log*.

## 5. Conclusiones

Debido a la facilidad de configuración que recae en un solo archivo de configuración, este podrá ser leído haciendo uso de las Macros respetando la gramática usada por la ideología OpenBSD, generando así un entorno más amigable.

Este servicio pretende cumplir con las exigencias que hay actualmente en el mercado, por tal motivo soluciones como Wordpress y hasta ownCloud podrían ser expuestas haciendo uso de httpd, ya que el mismo es compatible con lenguajes de programación como lo es PHP.

Un servidor fácil de configurar, ligero en cuanto consumo de servicios y seguro, sería un aval para llevar acabo la implementación en ambientes de producción en donde sera vital contar con plataformas que cumplan a la medida.

## 6. Enlaces relacionados

[1] <https://www.openbsd.org/papers/httpd-asiabsdcon2015.pdf>

[2] <http://man.openbsd.org/httpd.conf>

## 7. Historial de documentación

Nombre	E-mail	Fecha modificación
Andrés Restrepo González	handres999@gmail.com	10/07/17