

# Montando un servidor web con nginx

**telloX :)**

# Primera parte: Instalando Nginx con soporte para PHP y MySQL en Debian Linux

## Introducción

En primer lugar, se presenta un poco de información sobre el servidor HTTP Nginx, MySQL y PHP.

Después de un poco de fundamento teórico, pasamos a la parte técnica. Lo primero a describir serán los pasos para la instalación de cada uno de los servicios. Una vez que estos fueron instalados, pasamos a configurar los archivos necesarios para que funcionen correctamente.

Para finalizar, se realizará una pequeña página en PHP para comprobar que esté funcionando correctamente.

## Conociendo a Nginx

La información que a continuación se presenta fue extraída de la wiki oficial de Nginx en español: <http://wiki.nginx.org/NginxEs>

Nginx ("engine x") es un servidor HTTP y proxy inverso de alto rendimiento, y un servidor proxy para IMAP/POP3/SMTP. Nginx fue desarrollado por Igor Sysoev para Rambler.ru, el segundo sitio web más visitado de Rusia, donde ha estado funcionando en producción más de dos años y medio. Igor ha lanzado el código fuente bajo una licencia estilo BSD. Aunque aún se encuentra en una etapa beta, Nginx es conocido por su estabilidad, gran conjunto de características, configuración simple, y bajo consumo de recursos.

Ya que la mayoría de la documentación está en Ruso, esta wiki está para ayudar a los castellano-parlantes a instalar, configurar, e intercambiar trucos sobre Nginx. La mayoría de la documentación está derivada del trabajo de [[MailTo(al-nginx AT none DOT at,Aleksandar Lazic).

Versiones de Nginx

Existen 3 versiones del programa: Estable(Stable), de Desarrollo(Development), de Legado(Legacy). La version Estable es la recomendada para utilizacion general. La version de Desarrollo es la version que contiene las nuevas funciones y en la que los errores del programa son eliminados primero, pero tambien en la que hay mayor probabilidad de que aparezcan nuevos errores. La version de Legado es la version que fue Estable hasta que la nueva version apareciece, se mantiene para eliminar problemas de seguridad mayormente.

### Sitios Web que usan Nginx

- Nginx fue originalmente creado para:
- Rambler - El segundo sitio web más visitado de Rusia.
- Sitios en la lista de Alexa Top 500
- WordPress
- FC2
- Youporn
- TorrentReactor
- Ultimate Guitar
- Scribd
- Badoo

### Características básicas servidor web

Según información extraída de la wikipedia:

- Servidor de archivos estáticos, índices y autoindexado.
- Proxy inverso con opciones de caché.
- Balance de carga.
- Tolerancia a fallos.
- Soporte de HTTP sobre SSL.
- Soporte para FastCGI con opciones de caché.
- Servidores virtuales basados en nombre y/o en dirección IP.
- Streaming de archivos FLV y MP4.8
- Soporte para autenticación.

- Compresión gzip.
- Habilitado para soportar más de 10.000 conexiones simultáneas.9

### **Características proxy de correo**

- Proxy SMTP, POP3 e IMAP.
- Soporta STARTTLS.
- Soporta SSL.

## ¿Qué es MySQL?

Fuente: <http://indira-informatica.blogspot.mx/2007/09/qu-es-mysql.html>

MySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. MySQL AB es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de datos. Los información que puede almacenar una base de datos puede ser tan simple como la de una agenda, un contador, o un libro de visitas, ó tan vasta como la de una tienda en línea, un sistema de noticias, un portal, o la información generada en una red corporativa. Para agregar, acceder, y procesar los datos almacenados en una base de datos, se necesita un sistema de administración de bases de datos, tal como MySQL.

## MySQL es Open Source

Open Source significa que la persona que quiera puede usar y modificar MySQL. Cualquiera puede descargar el software de MySQL de Internet y usarlo sin pagar por ello. Inclusive, cualquiera que lo necesite puede estudiar el código fuente y cambiarlo de acuerdo a sus necesidades. MySQL usa la licencia GPL (Licencia Pública General GNU), para definir qué es lo que se puede y no se puede hacer con el software para diferentes situaciones. Sin embargo, si uno está incómodo con la licencia GPL o tiene la necesidad de incorporar código de MySQL en una aplicación comercial es posible comprar una versión de MySQL con una licencia comercial. Para mayor información, ver la página oficial de MySQL en la cuál se proporciona mayor información acerca de los tipos de licencias.

## ¿Por qué usar MySQL?

El servidor de bases de datos MySQL es muy rápido, seguro, y fácil de usar. Si eso es lo que se está buscando, se le debe dar una oportunidad a MySQL. Se pueden encontrar comparaciones de desempeño con algunos otros manejadores de bases de datos en la página de MySQL.

El servidor MySQL fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha estado siendo usado exitosamente en ambientes de producción sumamente exigentes por varios años. Aunque se encuentra en desarrollo constante, el servidor MySQL ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad, y seguridad hacen de MySQL un servidor bastante apropiado para acceder a bases de datos en Internet.

## Algunos detalles técnicos de MySQL

El software de bases de datos MySQL consiste de un sistema cliente/servidor que se compone de un servidor SQL multihilo, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs). Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, más rápido, y más fácil de manejar. Para obtener información técnica más detallada, es necesario consultar la guía de referencia de MySQL.

## ¿Qué es PHP?

Fuente: [http://vitaminaweb.com/que-es-php\\_863](http://vitaminaweb.com/que-es-php_863)

PHP es un lenguaje de programación interpretado (de ahí que nos refiramos a "scripts PHP") que se ejecuta del lado del servidor y permite la generación de páginas web dinámicas.

En realidad, PHP no sólo sirve para ésto. Más allá del desarrollo web, PHP permite incluso la creación de aplicaciones de escritorio (con una interfaz gráfica independiente del navegador), gracias a las bibliotecas Qt y GTK+.

El nombre PHP es un acrónimo de PHP Hypertext Pre-processor, desarrollado inicialmente por Rasmus Lerdorf en el año 1.994 con el objetivo de publicar su currículum vitae de forma online y almacenar algunos datos de interés sobre éste (como cuantas visitas recibía).

Actualmente la última versión estable es la 5.x , también llamado PHP 5, y próximamente se espera que sea lanzado PHP 6. Se recomienda utilizar las últimas versiones del lenguaje, es decir, PHP 5, ya que las anteriores no tienen mantenimiento (PHP 4 dejó de recibir soporte el 13 de julio de 2007).

El código fuente de un script PHP se escribe entre el código HTML de la página web, al igual que ocurre con JavaScript, sólo que PHP es interpretado en el servidor y nos devuelve el resultado de la ejecución, al contrario que JavaScript que se ejecuta del lado del cliente, en el navegador.

Entre las características más importantes de PHP tenemos:

- Es software libre.
- Es multiplataforma, se interpreta y ejecuta de igual forma un script independientemente del tipo de plataforma (tipo de sistema operativo o tipo de servidor web) donde sea ejecutado.
- Se lleva excelentemente bien con MySQL y otras bases de datos, facilitando así el desarrollo de aplicaciones web dinámicas que acceden a bases de datos en tiempo real.
- Es invisible al usuario de la página web, no puede verse el código fuente PHP desde el navegador web, ya que este código es interpretado en el servidor y nos devuelve la salida en HTML (que es lo que le llega al navegador).
- Permite la programación orientada a objetos.
- Dispone de una amplia biblioteca de funciones nativas.
- Las variables no necesitan ser definidas, sino que se evalúan en tiempo de ejecución, es decir, con darlas un valor ya quedan definidas también en su tipo.

Para ampliar, tenemos publicados un montón de artículos de PHP, y más que publicaremos próximamente, y también encontrarás más información, descargas y referencias en la web de PHP en [www.php.net](http://www.php.net)

## Instalación de los servicios

Al fin la parte práctica, recordemos que estamos instalando los servicios en Debian, sin embargo es posible usar este manual para montar el servidor web en alguna otra distribución simplemente adaptándolo un poco.

Nuestros aliados serán una terminal y apt. Los paquetes que se necesiten instalar recomiendo siempre buscarlos antes de instalarlo, es decir, ejecutar primero la orden:

```
# apt-cache search nombre_paquete
```

Y una vez que arroje los resultados, buscar el paquete y comprobar que se llame como en este manual. ¿Para qué hacer esto? A veces de una versión a otra los nombres de los paquetes cambian un poco y, si escribimos exactamente igual los comandos como en este manual, es posible que alguno marque error, así que, primero comprobamos que efectivamente se llamen así, y luego procedemos a instalarlos.

¿Qué hacer en caso de que cambie el nombre del paquete? Sencillo, simplemente se tiene que adaptar el nombre del paquete presentado en el presente manual por el que les arroje en la lista.

Pasemos ahora al primer paso:

### Instalación de MySQL

Instalaremos el servidor y el cliente, esto lo hacemos con la siguiente instrucción:

```
# apt-get install mysql-server mysql-client
```

Recordemos que la almohadilla o símbolo sharp (#) indica que la instrucción debe ser ejecutada como superusuario.

Ahora, cambiemos la contraseña de root para MySQL

```
# mysqladmin -u root password tu_contraseña
```

Una vez terminado de instalar, pasemos a la instalación de PHP.

### Instalación de PHP

Ahora instalaremos PHP en su versión 5. Además, instalaremos los paquetes necesarios para que se enlace con MySQL, el cli y el cgi. Junto con esto, instalaremos también el paquete spawn-fcgi de lighttpd para el cgi.

```
# apt-get install apt-get install php5 php5-cli php5-cgi spawn-fcgi
```

Terminado esto, pasemos a la instalación de Nginx.

### Instalación de Nginx

Para realizar la instalación, simplemente escribimos lo siguiente:

```
# apt-get install nginx
```

## Configuración básica

Lo primero que haremos será arrancar spawn-fcgi en el puerto que deseemos. En este caso lo haremos en el puerto 9000 pero puede cambiarse:

```
# /usr/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -u www-data -g www-data -f /usr/bin/php5-cgi -P /var/run/fastcgi-php.pid
```

Ahora editaremos el archivo del sitio por defecto de Nginx (/etc/nginx/sites-enabled/default):

```
# nano /etc/nginx/sites-enabled/default
```

El archivo debería quedar, de forma simple, algo así:

```
server {
    listen      80;
    server_name localhost;
    root        /var/www;

    location / {
        index   index.html index.php;
    }

    location ~* \.(gif|jpg|png)$ {
        expires 30d;
    }

    location ~ \.php$ {
        fastcgi_pass   localhost:9000;
        fastcgi_param  SCRIPT_FILENAME
                       $document_root$fastcgi_script_name;
        include        fastcgi_params;
    }
}
```

Guardamos los cambios. Ahora editaremos:

```
# nano /etc/php5/cgi/php.ini
```

Buscamos la línea “cgi.fix\_pathinfo = 1” la descomentamos (le borramos el ; al inicio de la línea, y nos aseguramos que tenga el valor de 1 y guardamos los cambios.



Reiniciamos Nginx

```
# /etc/init.d/nginx restart
```

Hasta este momento ya tenemos todo configurado, sin embargo falta algo muy importante. En el archivo de configuración de Nginx le dijimos que nuestro root (la ruta que contendrá los sitios web) se deben encontrar en /var/www; sin embargo, esta carpeta no está creada. Así que pasamos a crearla:

```
# mkdir /var/www
```

Y modificamos el dueño de esta carpeta, indicando que nginx es el dueño y que pueda hacer uso de ella. El usuario de nginx es www-data:

```
# chown www-data:www-data /var/www
```

Al fin tenemos todo listo para hacer uso de nuestro servidor web. Para finalizar la primera parte de este manual, crearemos una página muy simple en PHP, casi clásica para probar este tipo de cosas.

```
# nano /var/www/info.php
```

En este archivo debemos escribir lo siguiente:

```
<?php
    phpinfo();
?>
```

Guardamos cambios y cerramos el archivo. Para probar esta pequeña página nos vamos a nuestro navegador web favorito y escribimos en la barra de direcciones: <http://localhost/info.php>

Al dar enter nos debería arrojar información sobre la versión de PHP que tengamos instalado y varias cosas más relacionadas con PHP. Para probar que funcione correctamente MySQL con PHP, es necesario también crear una pequeña página de prueba.