

Instalando servidor Web Nginx + PHP (Ubuntu)



Para instalar un servidor Web en nuestro VPS vamos a seguir los siguientes pasos:

1. Instalar Nginx con apt (Linux Ubuntu)

```
$ sudo apt-get update  
$ sudo apt-get upgrade$ sudo apt-get install nginx
```

1.1 Firewall

Verificamos las aplicaciones disponibles para el firewall de nuestra VPS (no olvidar que hay otro firewall distinto, administrable desde la interfaz web de VPS.Guru):

```
$ sudo ufw app list
```

Esto podría darnos un listado similar a este:

```
Available applications:
```

```
Nginx Full
Nginx HTTP
Nginx HTTPS  OpenSSH
```

Si configuramos el firewall con la aplicación Nginx Full, se abren los puertos 443 y 80 (HTTPS y HTTP respectivamente), mientras que si lo configuramos con la aplicación Nginx HTTPS, se abre sólo el 443. Si configuramos con Nginx HTTP se abre el puerto 80. Es decir, si queremos permitir que se abra sólo el HTTP corremos el siguiente comando:

```
$ sudo ufw allow 'Nginx HTTP'
```

Y para comprobar qué aplicaciones están permitidas por el firewall, tipeamos:

```
$ sudo ufw status
```

Y nos da un resumen parecido al siguiente:

```
Status: active
To              Action    From--      - - - - -
OpenSSH         ALLOW     Anywhere    Nginx HTTP
ALLOW          Anywhere
Nginx HTTP (v6) ALLOW     Anywhere (v6)
```

En el listado superior vemos que está permitido Nginx HTTP para IP v6 e IP v4. Otra opción que impediría que funcione nuestro servidor web es que *Status* se encuentre en *inactive*.

Para asegurarnos de que Nginx se encuentra corriendo tipeamos:

```
$ systemctl status nginx
```

```
● nginx.service - A high performance web server and a reverse proxy server   Loaded: loaded
(/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)   Active: active
(running) since Mon 2017-05-18 16:14:00 EDT; 2min 1s ago
Main PID: 12855 (nginx)
CGroup: /system.slice/nginx.service      └─12855 nginx: master process
```

```
/usr/sbin/nginx -g daemon on; master_process on
```

```
└─12856 nginx: worker process
```

Debes comprobar que diga active (running). Y para verificar que nuestro servidor esté funcionando bien, vamos a tomar nuestra IP pública (ej: 123.12.123.12) del VPS y la vamos a tipear en un navegador web:

`http://123.12.123.12`

El navegador debería mostrarnos una página similar a esta:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Si vemos este mensaje quiere decir que ya tenemos nuestro servidor Nginx funcionando correctamente.

Los comandos básicos de Nginx son:

```
$ sudo systemctl start nginx #iniciar el servidor (por defecto luego de instalarlo va a
iniciar automáticamente)
$ sudo systemctl stop nginx #parar el servidor
$ sudo systemctl restart nginx #para el servidor
y luego hace start
$ sudo systemctl reload nginx #reinicia el servidor sin perder las
conexiones abiertas
$ sudo systemctl disable nginx #deshabilita el inicio automático de nginx
a iniciar el sistema operativo de nuestro VPS
$ sudo systemctl enable nginx #habilita el inicio automático de nginx cuando inicie el sistema operativo
```

2. Instalar PHP

Instalamos el paquete *php-fpm* con:

```
$ sudo apt-get install php-fpm
```

Luego buscamos el archivo de configuración de PHP:

```
$ php --ini |grep Loaded
```

```
Loaded Configuration File:      /etc/php/7.0/cli/php.ini
```

Editamos el archivo `/etc/php/7.0/cli/php.ini` con *vi*, *vim* o *nano*:

```
$ vi /etc/php/7.0/cli/php.ini
# Nos aseguramos que contenga esto:cgi.fix_pathinfo=0
```

Reiniciamos el php-fpm con:

```
$ systemctl restart php7.0-fpm
```

Por último configuramos el Nginx para que permita PHP editando el siguiente archivo con *vi*:

```
$ vi /etc/nginx/sites-available/default
```

Nos aseguramos que la línea del bloque que comienza con `location ~\.``php$` exista y esté descomentado (sin `#`)

```
server {
    listen 80 default_server;

    root /var/www/html;
    # Add index.php to the list if you are using PHP      index index.php index.html
index.htm index.nginx-debian.html;
    server_name _;
    location / {      # First attempt to serve request as file,
then
        # as directory, then fall back to displaying a 404.      try_files
$uri $uri/ =404;
```

```

    }
    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass
unix:/run/php/php7.0-fpm.sock;
    }
    # deny access to .htaccess files, if Apache's document root    # concurs with
nginx's one
    #
    location ~ /\.ht {
        deny all;
    }}

```

Guardamos el archivo y ejecutamos la prueba de la configuración del Nginx:

```
$ nginx -t
```

Debería arrojar que la sintaxis es correcta y la prueba del archivo de configuración fue exitosa (*syntax is ok* y *test is successful*):

```

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

```

Por último reiniciamos el servidor Nginx:

```
$ systemctl restart nginx
```

Para verificar si está funcionando correctamente el PHP creamos el siguiente archivo con *vi*:

```
$ sudo vi /var/www/html/info.php
```

Escribimos esto dentro y guardamos:

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

Ahora cargamos con un navegador la URL de nuestra IP pública (ej: 123.12.123.12):

<http://123.12.123.12/info.php>

La URL debería mostrarnos una página Web, con un banner similar a la siguiente imagen, que dice *PHP Version* y la versión de nuestro PHP junto con más información técnica.



Por último, eliminamos el archivo *info.php* por razones de seguridad:

```
$ sudo rm /var/www/html/info.php
```

Revision #16

Created 6 years ago by [Admin](#)

Updated 6 years ago by [Admin](#)