

NGINX y Apache en Ubuntu server 23/46



Curso de Administración de Servidores Linux



→||INSTALACIÓN Y C...

NGINX y Apache son softwares para montar servidores web, puedes realizar la instalación de ambos en el sistema operativo, teniendo como base que pueden estar corriendo al mismo tiempo, siempre y cuando no estén a la espera de conexiones por el mismo puerto.

Para validar los puertos que tienen un proceso activo usamos:

```
sudo netstat -tulpn
```

Podríamos tener una infraestructura donde NGINX puede servir como proxy y Apache como servidor web.

nginx Reverse Proxy + Apache + WordPress

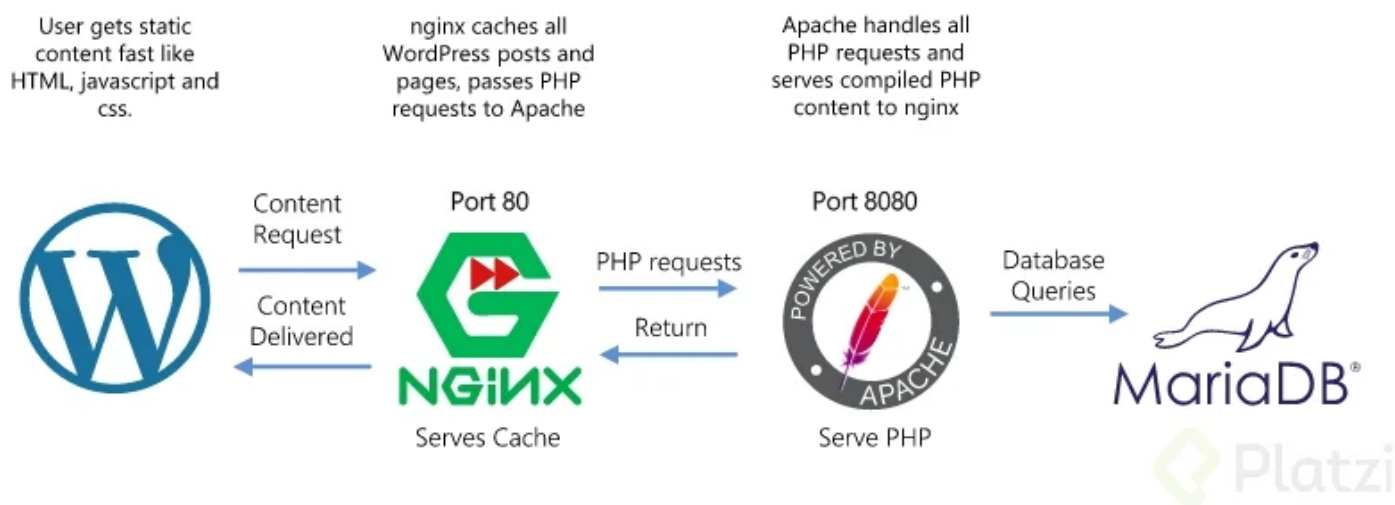
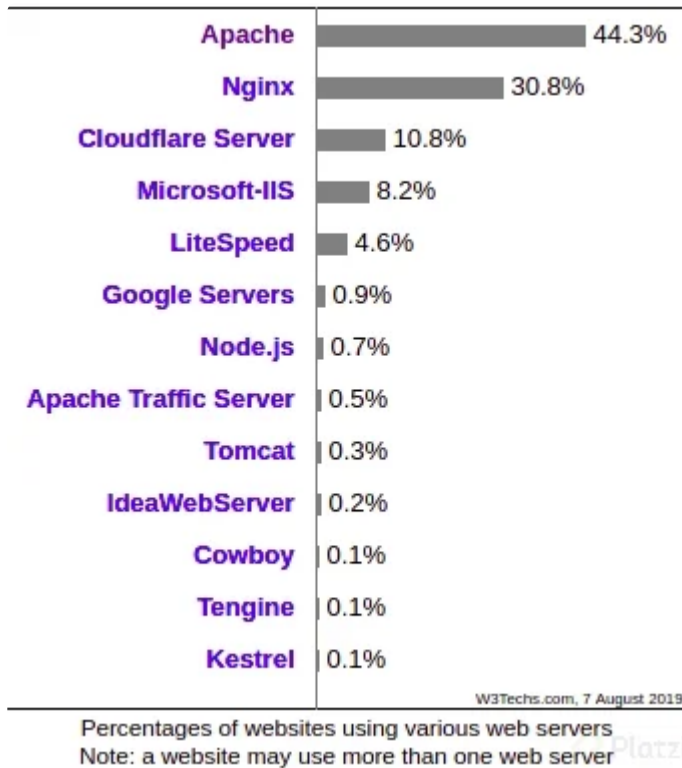


Imagen tomada de: <https://guides.wp-bullet.com/how-to-configure-nginx-reverse-proxy-wordpress-cache-apache/>

Si revisamos las estadísticas podemos ver que Apache aún es el líder del mercado en servidores web, seguido por NGINX, es por esta razón que veremos la instalación y configuración de ambos.



Existen en internet artículos interesantes de comparación entre ambos y el caso de uso de cada uno de ellos.

<https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations>

<https://www.nginx.com/faq/what-is-nginx-how-different-is-it-from-e-g-apache/>

Proceso de instalación.

Apache

Ejecuta el siguiente comando

```
sudo apt install apache2
```

NGINX

Ejecuta el siguiente comando

```
sudo apt install nginx nginx-extras
```

Para verificar si los servicios están corriendo se debe ejecutar los siguientes comandos:

```
systemctl status apache2
```

```
systemctl status nginx
```

Si se siguió el orden de instalación, NGINX no debe estar ejecutándose, pues por defecto intentará levantarse en el puerto 80, el cual ya se encuentra ocupado por Apache, para ello cambiaremos el puerto de Apache al puerto alternativo `http 8080`.

```
sudo nano /etc/apache2/ports.conf
```

A continuación tenemos que cambiar el puerto al 8080, para esto se debe cambiar la instrucción `Listen 8080` dentro del documento `ports.conf`.

Después abrimos nuestro archivo de configuración de Apache **`sudo nano /etc/apache2/sites-available/000-default.conf`** y cambiamos el `VirtualHost` a `8080` `<VirtualHost *:8080>`

Después realizamos el proceso de detener `apache2` y volverlo a encender, con los siguientes comandos

```
sudo systemctl restart apache2.
```

```
systemctl status apache2
```

```
systemctl status nginx.
```

Ambos sitios deberían estar activos y en ejecución.

Paso siguiente, dirígete al archivo de configuración de NGINX donde te asegurarás que exista una directiva en el *location* llamada `proxy_pass` que contenga lo siguiente:

```
proxy_pass http://127.0.0.1:8080;
```

```
server {  
    listen 80;  
    listen [::]:80;  
  
    server_name platzi;  
  
    location / {  
        proxy_set_header X-Forwarded-Host $host;  
        proxy_set_header X-Forwarded-Server $host;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header Host $host;  
        proxy_pass http://localhost:8080/;  
    }  
}
```

Si por alguna razón el servidor Apache no se encuentra en la misma máquina, debemos cambiar la dirección IP y el puerto respectivo.

Apache tiene un comando para activar sitios que es `a2ensite` que recibe como parámetro el archivo de configuración definido en `/etc/apache2/sites-available`. NGINX no cuenta con este comando, motivo por el cual se tiene un enlace blando, es decir, cuando creamos un archivo de configuración en `/etc/nginx/sites-available` debemos ejecutar `sudo ln -s /etc/nginx/sites-available/configuracion_nginx /etc/nginx/sites-enabled/`

Apache también me permite deshabilitar sitios y agregar módulos

```
sudo a2dissite 000-default
```

```
sudo a2enmod rewrite headers env dir mime
```

Si queremos activar `letsecrypt` en NGINX, debemos agregar una línea en el `.htaccess` en la ruta `/var/www/html/nombre_host/.htaccess`. La línea es `SetEnvIf X-`

Forwarded-Proto https

```
SetEnvIf X-Forwarded-Proto https HTTPS  
# BEGIN WordPress  
<IfModule mod_rewrite.c>  
RewriteEngine On  
RewriteBase /  
RewriteRule ^index\.php$ - [L]  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteRule . /index.php [L]  
</IfModule>  
# END WordPress
```

Conclusión

Antes de realizar la elección de uno de los dos, deberías mirar el tipo de proyecto en el que estás trabajando y que se acople mejor a tus necesidades, es un proceso de evaluación y prueba en cada uno de los aspectos que esperamos como administradores de sistemas. Existen múltiples diferencias entre ambos proyectos, que tienen impacto real en el rendimiento y tiempo de configuración para lograr que el servicio quede funcionando perfectamente. Algunos prefieren NGINX por la sintaxis de configuración, otros eligen basado en las estadísticas presentadas y otros por simple experiencia con trabajos anteriores. Yo te recomiendo probar ambos y elegir según el proyecto, o quizás puedes usarlos ambos y sacar lo mejor de cada uno.