

Nginx, para servidores con multiples vhosts a pelo (sin panel) + PHP-FPM

Nginx

Instalado de forma oficial siguiendo la ruta de instalación estándar de Ubuntu.

Mejoras sobre la configuración original

A continuación se describen las mejoras implementadas sobre la configuración estándar. Estas optimizaciones se centran en:

- Reducción del TTFB (Time To First Byte)
- Refuerzo de seguridad (especialmente en ciphers)
- Optimización de rendimiento general

Las directivas que comienzan con `# comentario` son personalizaciones, ya sea por adición o modificación.

```
user www-data;
worker_processes auto;
worker_rlimit_nofile 65535; # Incrementa el límite de archivos abiertos

# load_module      modules/nginx_http_modsecurity_module.so;
error_log /var/log/nginx/error.log notice;
pid        /var/run/nginx.pid;

include          /etc/nginx/modules-enabled/*.conf;
```

Configuración optimizada de eventos

```
events {  
    worker_connections 1024;  
    use                epoll; # Método eficiente para Linux  
    multi_accept       on;    # Acepta múltiples conexiones por proceso  
}
```

http {

```
include    /etc/nginx/mime.types;  
default_type application/octet-stream;
```

```
log_format main '$remote_addr - $remote_user [$time_local] "$request" '  
                '$status $body_bytes_sent "$http_referer" '  
                '"$http_user_agent" "$http_x_forwarded_for"';
```

```
access_log /var/log/nginx/access.log main;
```

Mejoras de rendimiento y seguridad probadas durante años

```
sendfile                on;    # Optimiza el envío de archivos  
tcp_nopush              on;    # Optimiza paquetes TCP  
tcp_nodelay             on;    # Reduce latencia  
client_header_timeout   60s;   # Tiempo máximo para recibir cabeceras  
client_body_timeout     60s;   # Tiempo máximo para recibir cuerpo  
client_header_buffer_size 2k;   # Tamaño del buffer para cabeceras  
client_body_buffer_size 256k;  # Tamaño del buffer para el cuerpo  
client_max_body_size    256m;  # Tamaño máximo de petición  
large_client_header_buffers 4 8k; # Buffers para cabeceras grandes  
send_timeout            60s;   # Tiempo máximo de envío  
keepalive_timeout       30s;   # Tiempo máximo de conexión persistente  
reset_timedout_connection on;   # Libera conexiones que expiran  
server_tokens           off;    # Oculta la versión de Nginx  
server_name_in_redirect off;    # No incluye nombre del servidor en redirecciones  
server_names_hash_max_size 512; # Tamaño máximo de la tabla hash  
server_names_hash_bucket_size 512; # Tamaño del bucket hash
```

Compresión para optimizar el tráfico

```
gzip      on;      # Activa la compresión  
gzip_static on;     # Busca versiones pre-comprimidas
```

```
gzip_vary      on;          # Añade cabecera Vary
gzip_comp_level 6;          # Nivel de compresión (equilibrio rendimiento/tamaño)
gzip_min_length 1024;       # Tamaño mínimo para comprimir
gzip_buffers   16 8k;       # Buffers para compresión
gzip_types     text/plain text/css text/javascript text/js text/xml application/json application/javascript
application/x-javascript application/xml application/xml+rss application/x-font-ttf image/svg+xml font/opentype;
gzip_proxied   any;         # Comprime respuestas proxy
gzip_disable   "MSIE [1-6]\."; # Desactiva para IEs antiguos

# Configuración de proxy
proxy_redirect off;
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_pass_header Set-Cookie;
proxy_buffers   32 4k;
proxy_connect_timeout 30s;
proxy_send_timeout 90s;
proxy_read_timeout 90s;

# Configuración para Cloudflare (actualizada: 28/09/2023)
# Permite identificar la IP real del visitante detrás de Cloudflare
set_real_ip_from 103.21.244.0/22;
set_real_ip_from 103.22.200.0/22;
set_real_ip_from 103.31.4.0/22;
set_real_ip_from 104.16.0.0/13;
set_real_ip_from 104.24.0.0/14;
set_real_ip_from 108.162.192.0/18;
set_real_ip_from 131.0.72.0/22;
set_real_ip_from 141.101.64.0/18;
set_real_ip_from 162.158.0.0/15;
set_real_ip_from 172.64.0.0/13;
set_real_ip_from 173.245.48.0/20;
set_real_ip_from 188.114.96.0/20;
set_real_ip_from 190.93.240.0/20;
set_real_ip_from 197.234.240.0/22;
set_real_ip_from 198.41.128.0/17;
#set_real_ip_from 2400:cb00::/32; # IPv6 (comentados por compatibilidad)
```

```
#set_real_ip_from 2606:4700::/32;
#set_real_ip_from 2803:f800::/32;
#set_real_ip_from 2405:b500::/32;
#set_real_ip_from 2405:8100::/32;
#set_real_ip_from 2c0f:f248::/32;
#set_real_ip_from 2a06:98c0::/29;
real_ip_header CF-Connecting-IP;
```

```
# Configuración SSL que cumple con PCI Compliance
```

```
# Basado en https://blog.ss88.us/secure-ssl-https-nginx-vestacp
```

```
ssl_protocols TLSv1.2 TLSv1.3; # Elimina protocolos obsoletos (SSLv3)
```

```
ssl_prefer_server_ciphers on;
```

```
ssl_session_cache shared:SSL:10m;
```

```
ssl_ciphers "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384
```

```
EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH EDH+aRSA
RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS !MEDIUM";
```

```
#ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-
SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384;
```

```
ssl_dhparam dh4096.pem;
```

```
ssl_ecdh_curve secp384r1;
```

```
ssl_session_tickets off;
```

```
ssl_stapling on;
```

```
ssl_stapling_verify on;
```

```
resolver 1.1.1.1 8.8.8.8 valid=300s;
```

```
resolver_timeout 5s;
```

```
add_header Strict-Transport-Security "max-age=63072000; includeSubdomains; preload";
```

```
add_header X-Frame-Options SAMEORIGIN;
```

```
add_header X-Content-Type-Options nosniff;
```

```
# Configuración de caché
```

```
# IMPORTANTE: Verificar permisos y propiedad de los directorios
```

```
# Especialmente útil en entornos con alta carga de solicitudes
```

```
proxy_cache_path /var/cache/nginx levels=2 keys_zone=cache:10m inactive=60m max_size=1024m;
```

```
proxy_cache_key "$host$request_uri $cookie_user";
```

```
proxy_temp_path /var/cache/nginx/temp;
```

```
proxy_ignore_headers Expires Cache-Control;
```

```
proxy_cache_use_stale error timeout invalid_header http_502;
```

```
proxy_cache_valid any 1d;
```

```

# Bypass de caché para sesiones activas
map $http_cookie $no_cache {
    default 0;
    ~SESS 1;
    ~wordpress_logged_in 1;
}

# Configuración de caché de archivos
open_file_cache      max=10000 inactive=30s;
open_file_cache_valid 60s;
open_file_cache_min_uses 2;
open_file_cache_errors off;

# Configuración de cabeceras y expiración
# Mapa de tiempos de expiración según tipo de contenido
map $sent_http_content_type $expires {
    default            off;
    text/html          epoch; # No cachear HTML (dinámico)
    text/css           max;   # Cachear CSS al máximo
    application/javascript max; # Cachear JS al máximo
    ~image/            max;   # Cachear imágenes al máximo
}

##
## Configuración de Hosts Virtuales
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}

```

Permisos y propiedad del caché de Nginx

Para que Nginx funcione correctamente con la configuración de caché definida, los directorios implicados deben tener los permisos y propiedad adecuados:

Propiedad (Owner y Grupo)

- **Propietario:** El usuario con el que se ejecuta Nginx (típicamente `www-data` en Ubuntu)
- **Grupo:** El grupo correspondiente (típicamente `www-data` en Ubuntu)

Permisos recomendados

- `/var/cache/nginx`: 750 (drwxr-x---)
- `/var/cache/nginx/temp`: 750 (drwxr-x---)

Comandos para establecer estos permisos

```
# Crear los directorios si no existen
mkdir -p /var/cache/nginx/temp

# Establecer la propiedad correcta
chown -R www-data:www-data /var/cache/nginx

# Establecer los permisos adecuados
chmod 750 /var/cache/nginx
chmod 750 /var/cache/nginx/temp
```

Estos permisos garantizan que los directorios sean accesibles únicamente por el usuario que ejecuta Nginx, mejorando la seguridad del sistema al evitar accesos no autorizados a posible información sensible almacenada en caché.

Configuración de los Virtual Hosts

Existen dos enfoques para gestionar los archivos de configuración de hosts virtuales:

1. Enfoque centralizado (recomendado para entornos de producción)

Los archivos de configuración se colocan en `/etc/nginx/sites-available/` y se enlazan simbólicamente a `/etc/nginx/sites-enabled/`:

```
ln -s /etc/nginx/sites-available/domain.tld.conf /etc/nginx/sites-enabled/
```

Ventajas:

- Mayor control y seguridad
- Centralización de configuraciones
- Facilita auditorías de seguridad
- Previene cambios accidentales por usuarios sin privilegios

2. Enfoque por usuario (útil en entornos único administrador)

Los archivos se colocan en directorios de usuario y se enlazan a la configuración principal.

Limitaciones:

- Los usuarios sin acceso `sudo` no podrán:
 - Validar configuraciones con `nginx -t`
 - Recargar el servicio con `systemctl reload nginx`
 - Reiniciar el servicio con `systemctl restart nginx`

Configuración inicial (pre-certificado)

Antes de obtener el certificado SSL, se debe crear un archivo de configuración sin asignación específica de puertos. Certbot (Let's Encrypt) se encargará de esto durante el proceso de certificación.

Ejemplo:

```
server {
    server_name domain.tld www.domain.tld;

    root    /home/user/web/domain.tld/domain/dist; # Ruta para despliegues JS/VUE/React
    index   index.html;

    # Es ALTAMENTE RECOMENDABLE activar estas cabeceras de seguridad
    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";

    charset utf-8;

    ## Logs ubicados en la carpeta del usuario para facilitar su acceso
    access_log /home/user/logs/web/domain.tld.log combined;
    error_log  /home/user/logs/web/domain.tld.error.log error;

    expires $expires; # Utiliza la variable map definida en nginx.conf

    # Si has compilado y tienes operativo mod_security, descomenta la línea siguiente
    # include /etc/nginx/modsec/active.conf;

    location = /favicon.ico {
        log_not_found off;
    }
}
```

```

    access_log off;
}

location = /robots.txt {
    allow all;
    log_not_found off;
    access_log off;
}

#error_page 404 /index.php;

location / {
    root /home/user/web/domain.tld/domain/dist;
    index index.html;
    try_files $uri $uri/ /index.html; # Configuración para SPAs (React, Vue, etc.)
    #try_files $uri $uri/ =404;      # Alternativa para sitios estáticos

    location ~* ^.+\.(\.jpeg|jpg|png|gif|bmp|ico|svg|css|js)$ {
        access_log      off;
        log_not_found    off;
        expires          max;
    }
}

# Bloqueo de archivos de configuración sensibles
location ~* "\.(htaccess|htpasswd|env)$" {
    deny    all;
    return 404;
}

# Bloqueo de archivos y directorios ocultos
location ~ /\.(!well-known).* {
    deny all;
}
}

```

Configuración de la variable `$expires`

Esta variable se declara mediante la directiva `map` en el archivo `nginx.conf`. Si no está declarada, su uso generará un error.


```
map $sent_http_content_type $expires {  
    default                off;  
    text/html              epoch; # No cachear HTML (fecha 1/1/1970)  
    text/css               max;   # Cachear CSS al máximo  
    application/javascript max;   # Cachear JS al máximo  
    ~image/               max;   # Cachear imágenes al máximo  
}
```

“ **Buena práctica:** Después de cada modificación, ejecutar `nginx -t` para verificar la sintaxis de la configuración antes de recargar el servicio.

Certificados con Certbot

Certbot ya viene instalado en la mayoría de distribuciones modernas. Para nuevas instalaciones, su [manual oficial](#) proporciona instrucciones detalladas.

Se recomienda crear certificados específicos por dominio en lugar de certificados wildcard:

```
certbot --nginx -d domain.tld -d www.domain.tld
```

“ **Nota:** En sistemas con restricciones de red, es necesario abrir temporalmente los puertos 80 y 443 durante el proceso de renovación.

PHP

Instalado desde el repositorio de [Ondrej Surý](#), específicamente la versión PHP-FPM.

Ubicaciones importantes:

- Configuración de pools: `/etc/php/X.X/fpm/pool.d/www.conf` (donde X.X es la versión de PHP)
- Configuración general: `/etc/php/X.X/fpm/php.ini`

Configuración crítica para logging

Es **esencial** modificar la configuración del pool para activar el registro de errores:

- Editar `/etc/php/X.X/fpm/pool.d/www.conf` y cambiar:

```
catch_workers_output = yes
```

Sin esta modificación, PHP-FPM no registrará correctamente los errores en los logs de los dominios virtuales, lo que puede dificultar enormemente la depuración. Para más información, consulte [PHP log cuando usamos PHP-FPM con host virtuales](#).

JIT Compiler

PHP 8.x introduce el compilador JIT (Just-In-Time), que puede mejorar significativamente el rendimiento en determinados escenarios. Sin embargo, puede presentar desafíos durante el desarrollo activo con cambios frecuentes.

Para más información sobre la activación y configuración del JIT, consulte [Activar PHP8.2 JIT Compiler](#).

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

Revision #3

Created 4 June 2025 09:47:32 by Abkrim

Updated 4 June 2025 09:52:10 by Abkrim