

nginx

- [Certificados en Nginx](#)
- [Nginx para Wordpress en Plesk](#)
- [Parametrización Nginx](#)
- [Php con php-fpm y nginx](#)
- [Comandos útiles de nginx](#)

Certificados en Nginx

Como configurar los certificados en Nginx

Generamos el certificado a partir de nuestro certificado y el ca-crt

Por ejemplo tenemos nuestro certificado al que llamaremos micertificado.crt y el certificado raiz ca.crt

```
cat micertificado.crt ca.crt >> ssl-bundle.crt
```

Editar archivo de configuración NGINX

Después de que se cargue el certificado, debe modificar su archivo de configuración NGINX (por defecto se llama nginx.conf).

Luego, edita o agrega Virtual Host para el puerto 443 para el sitio web. Si no hay un host virtual para el puerto 443, puedes duplicar el registro para el puerto 80 (debe estar en el archivo de configuración por defecto) y cambiar el puerto 80 al puerto 443. Simplemente agregarlo debajo del módulo no seguro.

Luego deberás agregar estas líneas específicas al registro:

ssl on;

ssl_certificate (debe apuntar a la ubicación del archivo de certificado creado)

ssl_certificate_key (debe apuntar a la ubicación de la clave privada generada junto con la CSR que se utilizó durante la activación del certificado)

El host virtual completo debería verse así:

```
server {
    listen 443;
    ssl on;
    ssl_certificate /etc/ssl/ssl-bundle.crt;
    ssl_certificate_key /etc/ssl/ssl-tutorials.key;
    server_name ssl-tutorials.com;
    access_log /var/log/nginx/nginx.vhost.access.log;
    error_log /var/log/nginx/nginx.vhost.error.log;

    location / {
```

```
root /var/www/;  
index index.html;  
}  
}
```

Nginx para Wordpress en Plesk

Para que Wordpress (u otras aplicaciones) funcionen correctamente en Plesk, es necesario modificar las directivas de nginx con las siguientes:

```
if (!-e $request_filename) {
    rewrite ^.*$ /index.php last;
}
location ~* \.(eot|ttf|woff|woff2)$ {
    add_header Access-Control-Allow-Origin *;
}
types {
    #Default MIME types
    text/html html htm shtml;
    text/css css;
    text/xml xml;
    image/gif gif;
    image/svg+xml svg svgz;
    image/jpeg jpeg jpg;
    application/javascript js;
    application/atom+xml atom;
    application/rss+xml rss;
    #Custom MIME types
    audio/x-mpegurl m3u;
    audio/x-scpls pls;
}
location ~* \.(js|jpg|jpeg|gif|png|css|tgz|gz|rar|bz2|doc|pdf|ppt|tar|wav|bmp|rtf|swf|ico|flv|txt) {
    etag on;
    expires 30d;
    if_modified_since exact;
    add_header Pragma "public";
    add_header Cache-Control "max-age=31536000, public";
}
location ~* \.(mp4)$ {
    expires 365d;
}
# enable gzip compression
gzip on;
gzip_min_length 50;
gzip_buffers 4 32k;
gzip_proxied any;
gzip_comp_level 9;
gzip_types
text/css
text/plain
```

```
text/javascript
application/javascript
application/json
application/x-javascript
application/xml
application/xml+rss
application/xhtml+xml
application/x-font-ttf
application/x-font-opentype
application/vnd.ms-fontobject
image/svg+xml
image/x-icon
application/rss+xml
application/atom+xml;
gzip_vary on;
# end gzip configuration
```

Parametrización Nginx

Parámetros varios y configuraciones de Nginx

Habilitar http2

Para habilitar http2 en el apartado del server 443 inserta

```
server {  
    listen 443 ssl http2;
```

Permitir o denegar por ip

En este ejemplo, denegamos la 192.168.1.1, permitimos el rango 192.168.1.0/24 (excepto la 192.168.1.1), la 10.1.1.0/26 y la 2001:0db8::/32. Denegamos el resto.

```
location / {  
    deny 192.168.1.1;  
    allow 192.168.1.0/24;  
    allow 10.1.1.0/16;  
    allow 2001:0db8::/32;  
    deny all;  
}
```

Php con php-fpm y nginx

Por defecto Nginx no es capaz de procesar páginas web dinámicas con PHP, por lo que es necesario modificar el handler de php en nginx

En muchos casos podemos ver una sección como esta:

```
# pass PHP scripts to FastCGI server
location ~ .php$ {
    include snippets/fastcgi-php.conf;
    #
    # # With php-fpm (or other unix sockets):
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    # # With php-cgi (or other tcp sockets):
    # fastcgi_pass 127.0.0.1:9000;
}
```

En este caso estamos usando php7.4, cambiaremos la versión en función de nuestra versión de php

Comandos útiles de nginx

Una serie de comandos útiles desde consola para gestionar nuestras instancias de Nginx

Comprobar los ficheros de configuración

Cuando se realizan modificaciones en los ficheros de configuración de los sitios Nginx, antes de reiniciar es conveniente comprobar que todo funciona para evitar dejar a los demás usuarios no afectados por las modificaciones sin servicio. Para eso ejecutaremos Nginx desde consola con el argumento `t` de test.

```
nginx -t
```

Recargar Nginx sin reiniciar el servicio

Es conveniente aplicar la configuración sin reiniciar el servicio. Esto no finaliza las conexiones actuales y elimina el período de espera al cargar el servicio, es decir, no hay tiempo de inactividad. Para realizar esto se pueden usar varios comandos.

```
nginx -s reload
```

O bien

```
/etc/init.d/nginx reload
```

O también

```
service nginx reload
```