

# REFERENCIA RÁPIDA DE NGINX

Bloques de servidor, proxy, SSL, balanceo de carga, logging

## Instalación

### Instalar por SO

```
Ubuntu / Debian `sudo apt install nginx`
RHEL / CentOS `sudo dnf install nginx`
macOS `brew install nginx`
Alpine `apk add nginx`
Docker `docker run -p 80:80 nginx`
```

### Gestión del servicio

```
sudo systemctl start nginx Iniciar Nginx
sudo systemctl stop nginx Detener Nginx
sudo systemctl reload nginx Recargar configuración (sin tiempo de inactividad)
sudo systemctl enable nginx Habilitar al inicio
nginx -t Verificar sintaxis de configuración
nginx -T Verificar y mostrar configuración completa
nginx -s reload Señalar al proceso en ejecución que recargue
```

## Configuración básica

### Ubicaciones de archivos

```
/etc/nginx/nginx.conf Archivo de configuración principal
/etc/nginx/conf.d/ Configuraciones de sitios adicionales (*.conf)
/etc/nginx/sites-available/ Configuraciones de sitios disponibles (Debian)
/etc/nginx/sites-enabled/ Symlinks a configuraciones activas
/var/log/nginx/ Logs de acceso y de error
/var/www/html/ Raíz de documentos por defecto
```

### Configuración mínima

```
server {
    listen 80;
    server_name example.com;
    root /var/www/mysite;
    index index.html;
}
```

### Estructura de configuración

```
http { } Configuración del servidor HTTP (nivel superior)
server { } Definición de host virtual
location { } Bloque de coincidencia de URI
upstream { } Grupo de servidores backend
events { } Configuración de manejo de conexiones
```

## Bloques de servidor

### Hosts virtuales basados en nombre

```
server {
    listen 80;
    server_name site-a.com;
    root /var/www/site-a;
}
server {
    listen 80;
    server_name site-b.com;
    root /var/www/site-b;
}
```

### Por defecto y captura total

```
server {
    listen 80 default_server;
    server_name _;
    return 444; # drop connection
}
```

### Redirección a HTTPS

```
server {
    listen 80;
    server_name example.com;
    return 301 https://$host$request_uri;
}
```

## Bloques de location

### Prioridad de coincidencia (mayor a menor)

```
= /path Coincidencia exacta (mayor prioridad)
^~ /path Coincidencia de prefijo, omitir regex
~ regex Regex sensible a mayúsculas
~* regex Regex insensible a mayúsculas
/path Coincidencia de prefijo (menor prioridad)
```

### Ejemplos de location

```
location = / {
    # exact root only
}
location /api/ {
    proxy_pass http://backend;
}
location ~* \.(jpg|png|gif)$ {
    expires 30d;
}
```

### try\_files

```
location / {
    try_files $uri $uri/ /index.html;
```

Intenta archivo, luego directorio, luego alternativa — esencial para SPAs

## Proxy inverso

### Proxy básico

```
location /api/ {
    proxy_pass http://localhost:3000/;
    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_set_header X-Forwarded-Proto $scheme;
```

## Proxy WebSocket

```
location /ws/ {
    proxy_pass http://localhost:3000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}
```

### Directivas de proxy

```
proxy_pass URL del backend
proxy_set_header Pasar cabeceras personalizadas al backend
proxy_read_timeout Tiempo de espera para respuesta del backend (60s por defecto)
proxy_buffering off Desactivar buffering de respuesta
proxy_redirect Reescribir cabeceras Location del backend
```

## SSL/TLS

### Servidor HTTPS

```
server {
    listen 443 ssl;
    server_name example.com;

    ssl_certificate /etc/ssl/certs/example.crt;
    ssl_certificate_key /etc/ssl/private/example.key;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;
}
```

### Let's Encrypt con Certbot

```
sudo certbot --nginx -d example.com
sudo certbot renew --dry-run
```

### Buenas prácticas SSL

```
ssl_protocols TLSv1.2 TLSv1.3 Deshabilitar versiones antiguas de TLS
ssl_prefer_server_ciphers on El servidor elige el cifrado
ssl_session_cache shared:SSL:10m Reutilización de sesión para rendimiento
add_header Strict-Transport-Security Cabecera HSTS
ssl_stapling on Grapado OCSP para handshake más rápido
```

## Balanceo de carga

### Bloque upstream

```
upstream backend {
    server 10.0.0.1:3000;
    server 10.0.0.2:3000;
    server 10.0.0.3:3000;
}
server {
    location / {
        proxy_pass http://backend;
    }
}
```

### Métodos de balanceo de carga

```
(default) Round-robin
least_conn Menos conexiones activas
ip_hash Sesiones pegajosas por IP del cliente
hash $request_uri Hash consistente por URI
```

### Opciones de servidor

```
weight=3 Enviar 3 veces más tráfico
max_fails=3 Fallos antes de marcar como inactivo
fail_timeout=30s Tiempo para marcar servidor como inactivo
backup Usar solo cuando los demás estén caídos
down Marcar servidor como permanentemente fuera de línea
```

## Archivos estáticos y caché

### Servir archivos estáticos

```
location /static/ {
    alias /var/www/assets/;
    expires 30d;
    add_header Cache-Control "public, immutable";
}
```

### Compresión Gzip

```
gzip on;
gzip_types text/plain text/css application/javascript;
gzip_min_length 1000;
gzip_comp_level 5;
```

### Directivas de caché

```
expires 30d Establecer Expires y Cache-Control max-age
expires off Desactivar la cabecera expires
etag on Habilitar cabecera ETag (por defecto)
sendfile on Servicio eficiente de archivos via kernel
tcp_nopush on Optimizar el envío de paquetes
```

## Logging

### Configuración de logs

```
access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log warn;

# Custom log format
log_format main "$request" $body_bytes_sent";
access_log /var/log/nginx/access.log main;
```

### Niveles de log de error

```
debug Detallado (requiere --with-debug)
info Informativo
notice Normal pero notable
warn Advertencias
error Errores (por defecto)
```

### crit Problemas críticos

### Logging condicional

```
map $status $loggable {
    ~{2} 0;
    default 1;
}
access_log /var/log/nginx/access.log combined if=$loggable;
Omitir logging de respuestas 2xx/3xx para reducir el volumen de logs
```

## Seguridad

### Limitación de velocidad

```
limit_req_zone $binary_remote_addr
zone=api:10m rate=10r/s;

location /api/ {
    limit_req zone=api burst=20 nodelay;
}
```

### Control de acceso

```
location /admin/ {
    allow 192.168.1.0/24;
    deny all;
}
```

### Cabeceras de seguridad

```
X-Frame-Options DENY Prevenir clickjacking
X-Content-Type-Options nosniff Prevenir MIME sniffing
X-XSS-Protection "1; mode=block" Filtro XSS (navegadores legados)
```

### Content-Security-Policy

```
Controlar fuentes de carga de recursos
Referer-Policy no-referrer Controlar información de referencia
```

## Patrones comunes

### SPA (aplicación de página única)

```
location / {
    root /var/www/app;
    try_files $uri $uri/ /index.html;
}
```

### Cabeceras CORS

```
location /api/ {
    add_header Access-Control-Allow-Origin *;
    add_header Access-Control-Allow-Methods "GET, POST, PUT, DELETE, OPTIONS";
    if ($request_method = OPTIONS) {
        return 204;
    }
    proxy_pass http://backend;
}
```

### Variables útiles

```
$host Cabecera Host de la petición
$uri URI actual (normalizada)
$request_uri URI original con cadena de consulta
$remote_addr Dirección IP del cliente
$scheme http o https
$args Parámetros de cadena de consulta
$status Código de estado de la respuesta
```