

# Tham Khảo Nhanh Nginx

Server blocks, proxying, SSL, load balancing, logging

## Cài Đặt

### Cài Đặt Theo Hệ Điều Hành

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

### Quản Lý Service

sudo systemctl start nginx	Khởi động Nginx
sudo systemctl stop nginx	Dừng Nginx
sudo systemctl reload nginx	Tải lại cấu hình (không downtime)
sudo systemctl enable nginx	Kích hoạt khi khởi động
nginx -t	Kiểm tra cú pháp cấu hình
nginx -T	Kiểm tra và in toàn bộ cấu hình
nginx -s reload	Báo hiệu process đang chạy tải lại

## Cấu Hình Cơ Bản

### Vị Trí File

/etc/nginx/nginx.conf	File cấu hình chính
/etc/nginx/conf.d/	Cấu hình site bổ sung (*.conf)
/etc/nginx/sites-available/	Cấu hình site có sẵn (Debian)
/etc/nginx/sites-enabled/	Symlinks đến cấu hình đang hoạt động
/var/log/nginx/	Log truy cập và lỗi
/var/www/html/	Document root mặc định

### Cấu Hình Tối Giản

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

## Cấu Trúc Cấu Hình

http { }	Cài đặt HTTP server (cấp cao nhất)
server { }	Định nghĩa virtual host
location { }	Khối khớp URI
upstream { }	Nhóm backend servers
events { }	Cài đặt xử lý kết nối

## Server Blocks

### Virtual Hosts Theo Tên

```
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;
}
```

### Mặc Định & Catch-All

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

## Chuyển Hướng HTTPS

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

## Location Blocks

### Ưu Tiên Khớp (cao đến thấp)

= /path	Khớp chính xác (ưu tiên cao nhất)
^~ /path	Khớp prefix, bỏ qua regex
~ regex	Regex phân biệt hoa/thường
~* regex	Regex không phân biệt hoa/thường
/path	Khớp prefix (ưu tiên thấp nhất)

### Ví Dụ 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;
}
```

Thử file, rồi thư mục, rồi fallback -- cần thiết cho SPAs

## Reverse Proxy

### Proxy Cơ Bản

```
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";
}
```

### Proxy Directives

proxy_pass	URL backend
proxy_set_header	Truyền header tùy chỉnh đến backend
proxy_read_timeout	Timeout cho phản hồi backend (mặc định 60s)
proxy_buffering off	Tắt buffering phản hồi
proxy_redirect	Viết lại Location headers từ backend

## SSL / TLS

### HTTPS Server

```
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 với Certbot

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

### Thực Hành Tốt Nhất SSL

ssl_protocols TLSv1.2 TLSv1.3	Tắt các phiên bản TLS cũ
ssl_prefer_server_ciphers on	Server chọn cipher
ssl_session_cache shared:SSL:10m	Tái sử dụng session để tối ưu hiệu suất
add_header Strict-Transport-Security	HSTS header
ssl_stapling on	OCSP stapling cho handshake nhanh hơn

## Load Balancing

### Upstream Block

```
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;
    }
}
```

### Phương Pháp Load Balancing

(mặc định)	Round-robin
least_conn	Ít kết nối hoạt động nhất
ip_hash	Sticky sessions theo IP client
hash \$request_uri	Consistent hash theo URI

### Tùy Chọn Server

weight=3	Gửi nhiều gấp 3 lần traffic
max_fails=3	Số lần thất bại trước khi đánh dấu down
fail_timeout=30s	Thời gian đánh dấu server là down
backup	Chỉ dùng khi các server khác down
down	Đánh dấu server offline vĩnh viễn

## File Tĩnh & Caching

### Phục Vụ File Tĩnh

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

# Tham Khảo Nhanh Nginx

## Nén Gzip

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

## Directives Caching

<b>expires 30d</b>	Đặt Expires và Cache-Control max-age
<b>expires off</b>	Tắt expires header
<b>etag on</b>	Bật ETag header (mặc định)
<b>sendfile on</b>	Phục vụ file hiệu quả qua kernel
<b>tcp_nopush on</b>	Tối ưu hóa việc gửi gói tin

## Logging

### Cấu Hình Log

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

```
# Custom log format
log_format main '$remote_addr - $status '
               '$request' $body_bytes_sent';
access_log /var/log/nginx/access.log main;
```

### Mức Độ Log Lỗi

<b>debug</b>	Chi tiết (cần --with-debug)
<b>info</b>	Thông tin
<b>notice</b>	Bình thường nhưng đáng chú ý
<b>warn</b>	Cảnh báo
<b>error</b>	Lỗi (mặc định)
<b>crit</b>	Vấn đề nghiêm trọng

### Log Có Điều Kiện

```
map $status $loggable {
    ~^[23] 0;
    default 1;
}
access_log /var/log/nginx/access.log combined if=$loggable;
```

Bỏ qua log phản hồi 2xx/3xx để giảm khối lượng log

## Bảo Mật

### Rate Limiting

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

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

### Kiểm Soát Truy Cập

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

### Security Headers

<b>X-Frame-Options DENY</b>	Ngăn chặn clickjacking
<b>X-Content-Type-Options nosniff</b>	Ngăn chặn MIME sniffing
<b>X-XSS-Protection "1; mode=block"</b>	Bộ lọc XSS (trình duyệt cũ)
<b>Content-Security-Policy</b>	Kiểm soát nguồn tài nguyên được tải
<b>Referrer-Policy no-referrer</b>	Kiểm soát thông tin referrer

## Mẫu Phổ Biến

### SPA (Single-Page App)

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

### CORS Headers

```
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;
}
```

### Biến Hữu Dụng

<b>\$host</b>	Host header của request
<b>\$uri</b>	URI hiện tại (đã chuẩn hoá)
<b>\$request_uri</b>	URI gốc với query string
<b>\$remote_addr</b>	Địa chỉ IP client
<b>\$scheme</b>	http hoặc https
<b>\$args</b>	Tham số query string
<b>\$status</b>	Mã trạng thái phản hồi