

Nginx Riferimento Rapido

Blocchi server, proxy, SSL, bilanciamento del carico, log

Installazione

Installa per OS

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

Gestione del Servizio

<code>sudo systemctl start nginx</code>	Avvia Nginx
<code>sudo systemctl stop nginx</code>	Ferma Nginx
<code>sudo systemctl reload nginx</code>	Ricarica la configurazione (senza downtime)
<code>sudo systemctl enable nginx</code>	Abilita all'avvio
<code>nginx -t</code>	Verifica la sintassi della configurazione
<code>nginx -T</code>	Verifica e stampa la configurazione completa
<code>nginx -s reload</code>	Invia segnale al processo in esecuzione per ricaricare

Configurazione di Base

Posizione dei File

<code>/etc/nginx/nginx.conf</code>	File di configurazione principale
<code>/etc/nginx/conf.d/</code>	Configurazioni aggiuntive dei siti (*.conf)
<code>/etc/nginx/sites-available/</code>	Configurazioni dei siti disponibili (Debian)
<code>/etc/nginx/sites-enabled/</code>	Symlink alle configurazioni attive
<code>/var/log/nginx/</code>	Log di accesso e di errore
<code>/var/www/html/</code>	Document root predefinito

Configurazione Minima

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

Struttura della Configurazione

<code>http { }</code>	Impostazioni server HTTP (livello principale)
<code>server { }</code>	Definizione di virtual host
<code>location { }</code>	Blocco di corrispondenza URI
<code>upstream { }</code>	Gruppo di server backend
<code>events { }</code>	Impostazioni per la gestione delle connessioni

Blocchi Server

Virtual Host Basati sul Nome

```
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;
}
```

Default e Catch-All

```
server {
    listen 80 default_server;
    server_name _;
    return 444; # chiude la connessione
}
```

Redirect HTTPS

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

Blocchi Location

Priorità di Corrispondenza (alta a bassa)

<code>= /path</code>	Corrispondenza esatta (priorità massima)
<code>^~ /path</code>	Prefisso, salta regex
<code>~ regex</code>	Regex case-sensitive
<code>~* regex</code>	Regex case-insensitive
<code>/path</code>	Corrispondenza per prefisso (priorità minima)

Esempi di Location

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

try_files

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

Prova file, poi directory, poi fallback — essenziale per le SPA

Reverse Proxy

Proxy di Base

```
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";
}
```

Direttive Proxy

<code>proxy_pass</code>	URL del backend
<code>proxy_set_header</code>	Passa header personalizzati al backend
<code>proxy_read_timeout</code>	Timeout per la risposta del backend (default 60s)
<code>proxy_buffering off</code>	Disabilita il buffering della risposta
<code>proxy_redirect</code>	Riscrive gli header Location dal backend

SSL / TLS

Server 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
```

Best Practice SSL

<code>ssl_protocols TLSv1.2 TLSv1.3</code>	Disabilita le versioni TLS obsolete
<code>ssl_prefer_server_ciphers on</code>	Il server sceglie il cipher
<code>ssl_session_cache shared:SSL:10m</code>	Riutilizzo della sessione per le prestazioni
<code>add_header Strict-Transport-Security</code>	Header HSTS
<code>ssl_stapling on</code>	OCSP stapling per handshake più veloci

Bilanciamento del Carico

Blocco 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;
    }
}
```

Metodi di Bilanciamento

<code>(predefinito)</code>	Round-robin
<code>least_conn</code>	Connessioni attive minime
<code>ip_hash</code>	Sessioni sticky per IP client
<code>hash \$request_uri</code>	Hash consistente per URI

Opzioni Server

<code>weight=3</code>	Invia 3 volte più traffico
<code>max_fails=3</code>	Errori prima di segnare il server come inattivo
<code>fail_timeout=30s</code>	Tempo per segnare il server come inattivo
<code>backup</code>	Usato solo quando gli altri sono inattivi
<code>down</code>	Segna il server come permanentemente offline

File Statici e Cache

Servire File Statici

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

Nginx Riferimento Rapido

Compressione Gzip

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

Direttive di Cache

expires 30d	Imposta Expires e Cache-Control max-age
expires off	Disabilita l'header expires
etag on	Abilita l'header ETag (predefinito)
sendfile on	Servizio file efficiente via kernel
tcp_nopush on	Ottimizza l'invio dei pacchetti

Logging

Configurazione dei Log

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

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

Livelli del Log degli Errori

debug	Dettagliato (richiede --with-debug)
info	Informativo
notice	Normale ma notevole
warn	Avvertenze
error	Errori (predefinito)
crit	Problemi critici

Logging Condizionale

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

Salta il log delle risposte 2xx/3xx per ridurre il volume

Sicurezza

Limitazione delle Richieste

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

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

Controllo degli Accessi

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

Header di Sicurezza

X-Frame-Options DENY	Previene il clickjacking
X-Content-Type-Options nosniff	Previene il MIME sniffing
X-XSS-Protection "1; mode=block"	Filtro XSS (browser legacy)
Content-Security-Policy	Controlla le sorgenti di caricamento risorse
Referrer-Policy no-referrer	Controlla le informazioni sul referrer

Pattern Comuni

SPA (Single-Page App)

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

Header 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;
}
```

Variabili Utili

\$host	Header Host della richiesta
\$uri	URI corrente (normalizzato)
\$request_uri	URI originale con query string
\$remote_addr	Indirizzo IP del client
\$scheme	http o https
\$args	Parametri della query string
\$status	Codice di stato della risposta