Nginx Riferimento Rapido

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

Installazione

Installa per OS

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`

Gestione del Servizio

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

Configurazione di Base

Posizione dei File

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

Configurazione Minima

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

Struttura della Configurazione

`http { }`	Impostazioni server HTTP (livello principale)
`server { }`	Definizione di virtual host
`location { }`	Blocco di corrispondenza URI
`upstream { }`	Gruppo di server backend
`events { }`	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)

`= /path`	Corrispondenza esatta (priorità massima)
`^~ /path`	Prefisso, salta regex
`~ regex`	Regex case-sensitive
`~* regex`	Regex case-insensitive
`/path`	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

`proxy_pass`	URL del backend
`proxy_set_header`	Passa header personalizzati al backend
`proxy_read_timeout`	Timeout per la risposta del backend (default 60s)
`proxy_buffering off`	Disabilita il buffering della risposta
`proxy_redirect`	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

`ssl_protocols TLSv1.2 TLSv1.3`	Disabilita le versioni TLS obsolete
`ssl_prefer_server_ciphers on`	Il server sceglie il cipher
`ssl_session_cache shared:SSL:10m`	Riutilizzo della sessione per le prestazioni
`add_header Strict-Transport-Security`	Header HSTS
`ssl_stapling on`	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

`(predefinito)`	Round-robin
`least_conn`	Connessioni attive minime
`ip_hash`	Sessioni sticky per IP client
`hash $request_uri`	Hash consistente per URI

Opzioni Server

`weight=3`	Invia 3 volte più traffico
`max_fails=3`	Errori prima di segnare il server come inattivo
`fail_timeout=30s`	Tempo per segnare il server come inattivo
`backup`	Usato solo quando gli altri sono inattivi
`down`	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