

NGINX 빠른 참조

서버 블록, 프록시, SSL, 로드 밸런싱, 로깅

설치	
OS별 설치	
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
서비스 관리	
sudo systemctl start nginx	Nginx 시작
sudo systemctl stop nginx	Nginx 중지
sudo systemctl reload nginx	설정 재로드 (무중단)
sudo systemctl enable nginx	부팅 시 자동 시작
nginx -t	설정 검증
nginx -s reload	재로드 시 전체 출력
nginx -S reload	재로드 시 프로세스에 재로드 신호 전송

기본 설정	
파일 위치	
/etc/nginx/nginx.conf	주 설정 파일
/etc/nginx/conf.d/	사이드별 설정 드로인 (*conf)
/etc/nginx/sites-available/	사용 가능한 사이트 설정 (Debian)
/etc/nginx/sites-enabled/	실제로 실행되는 사이트 설정
/var/log/nginx/	로그 디렉터리
/var/www/html/	웹사이트 루트
최소 설정	

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

설정 구조	
http { }	HTTP 서버 설정 (최상위)
server { }	가상 호스트 정의
location { }	URI 매칭 블록
upstream { }	백엔드 서버 그룹
events { }	연결 처리 설정

서버 블록	
이름 기반 가상 호스트	
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;
}	

기본 및 캐치올	
server {	listen 80 default_server;
	server_name _;
	return 444; # drop connection
}	

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

Location 블록	
매칭 우선순위 (높음->낮음)	
= /path	정확한 매칭 (최우선)
^~ /path	접두사 매칭 정규식 건너뛰기
~ regex	대소문자 구분 정규식
~* regex	대소문자 무시 정규식
/path	접두사 매칭 (최하위)

Location 예시	
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;
}	

파일 디렉터리 순서로 시도 후 폴백 - SPA에 필수

리버스 프록시	
기본 프록시	
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;
}	

WebSocket 프록시	
---------------	--

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

프록시 지시어	
proxy_pass	백엔드 URL
proxy_set_header	백엔드에 사용자 정의 헤더 전달
proxy_read_timeout	백엔드 응답 타임아웃 (기본 60s)
proxy_buffering off	응답 버퍼링 비활성화
proxy_redirect	백엔드의 Location 헤더 재작성

SSL/TLS	
HTTPS 서버	
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;
Certbot으로 Let's Encrypt	
sudo certbot -n --nginx -d example.com	
sudo certbot renew --dry-run	
SSL 모범 사례	
ssl_protocols TLSv1.2 TLSv1.3	구버전 TLS 비활성화
ssl_prefer_server_ciphers on	서버가 암호화 방식 선택
ssl_session_cache shared:SSL:10m	성능을 위한 세션 재사용
add_header Strict-Transport-Security	HSTS 헤더
ssl_stapling on	빠른 핸드셰이킹을 위한 OCSP 스테이플링

로드 밸런싱	
Upstream 블록	
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;
}	
로드 밸런싱 방식	
(기본값)	라운드 로빈
least_conn	활성 연결이 가장 적은 서버
ip_hash	클라이언트 IP 고정 세션
hash \$request_uxi	URI 기반 일관된 해시
서버 옵션	
weight=3	3배 더 많은 트래픽 전송
max_fails=3	다운 처리 전 실패 횟수
fail_timeout=30s	서버를 다운으로 표시하는 시간
backup	다른 서버가 모두 다운됐을 때만 사용
down	서버를 영구적으로 오프라인 처리

정적 파일 및 캐싱	
정적 파일 제공	
location /static/ {	alias /var/www/assets/;
	expires 30d;
	add_header Cache-Control "public, immutable";
}	
Gzip 압축	
gzip on;	
gzip_types text/plain text/css	
	application/json application/javascript;
gzip_min_length 1000;	
gzip_comp_level 5;	
캐싱 지시어	
expires 30d	Expires 및 Cache-Control max-age 설정
expires off	expires 헤더 비활성화
etag on	ETag 헤더 활성화 (기본값)
sendfile on	커널을 통한 효율적인 파일 제공
tcp_nopush on	패킷 전송 최적화

로깅	
로그 설정	
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;	
오류 로그 레벨	
debug	상세 로그 (--with-debug 필요)
info	정보성 메시지
notice	정상이지만 주목할 사항
warn	경고
error	오류 (기본값)
crit	심각한 문제
조건부 로깅	

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

로그 용량 절감을 위해 2xx/3xx 응답 로깅 건너뛰기

보안	
속도 제한	
limit_req_zone \$binary_remote_addr	zone=api:10m rate=10r/s;
location /api/ {	limit_req zone=api burst=20 nodelay;
}	
접근 제어	
location /admin/ {	allow 192.168.1.0/24;
	deny all;
}	

보안 헤더	
X-Frame-Options DENY	클릭재킹 방지
X-Content-Type-Options nosniff	MIME 스니핑 방지
X-XSS-Protection "1; mode=block"	XSS 필터 (구형 브라우저)
Content-Security-Policy	리소스 로딩 소스 제어
Referrer-Policy no-referrer	리퍼러 정보 제어

일반 패턴	
SPA (단일 페이지 앱)	
location / {	root /var/www/app;
	try_files \$uri \$uri/ /index.html;
}	
CORS 헤더	
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;
}	

유용한 변수	
\$host	요청 Host 헤더
\$uri	현재 URI (정규화된)
\$request_uri	원래 문자열 포함 원본 URI
\$remote_addr	클라이언트 IP 주소
\$scheme	http 또는 https
\$args	쿼리 문자열 파라미터
\$status	응답 상태 코드