

nginx Befehle

Praxiserprobte nginx Befehle und Konfig-Schnipsel für den Betrieb – durchsuchbar und mit einem Klick kopierbar. Konfig immer erst testen, dann neu laden.

daniele.michelin //
cheat sheet

Dienst & Konfiguration

Konfiguration auf Syntaxfehler prüfen

```
sudo nginx -t
```

Gesamte aktive Konfiguration inkl. Includes ausgeben

```
sudo nginx -T
```

Konfiguration neu laden ohne Verbindungen zu kappen

```
sudo nginx -s reload
```

Reload über systemd (Standardweg auf Debian/Ubuntu)

```
sudo systemctl reload nginx
```

! Dienst komplett neu starten (kappt Verbindungen)

```
sudo systemctl restart nginx
```

Erst testen, nur bei Erfolg neu laden

```
sudo nginx -t && sudo systemctl reload nginx
```

Logs & Auswertung

Access-Log live mitlesen

```
sudo tail -f /var/log/nginx/access.log
```

Error-Log live mitlesen

```
sudo tail -f /var/log/nginx/error.log
```

HTTP-Statuscodes zählen

```
sudo awk '{print $9}' /var/log/nginx/access.log | sort |  
uniq -c | sort -rn
```

Letzte Requests mit 5xx-Antwort

```
sudo awk '$9 ~ /^5/' /var/log/nginx/access.log | tail
```

Top-10-Client-IPs nach Requests

```
sudo awk '{print $1}' /var/log/nginx/access.log | sort |  
uniq -c | sort -rn | head
```

! Access-Log leeren (Datei bleibt bestehen)

```
sudo truncate -s 0 /var/log/nginx/access.log
```

Server-Blöcke & Redirects

Domainnamen eines Server-Blocks festlegen

```
server_name example.com www.example.com;
```

Auf Port 443 mit TLS lauschen

```
listen 443 ssl http2;
```

Alles von HTTP auf HTTPS weiterleiten

```
return 301 https://$host$request_uri;
```

www auf die Hauptdomain umleiten

```
return 301 https://example.com$request_uri;
```

Datei, Verzeichnis oder 404 ausliefern

```
try_files $uri $uri/ =404;
```

Upload-Limit anheben (z. B. Nextcloud)

```
client_max_body_size 512M;
```

Reverse Proxy

Requests an ein Backend weiterreichen

```
proxy_pass http://127.0.0.1:3000;
```

Originalen Host-Header ans Backend geben

```
proxy_set_header Host $host;
```

Client-IP-Kette an das Backend weitergeben

```
proxy_set_header X-Forwarded-For  
$proxy_add_x_forwarded_for;
```

Timeout für langsame Backends erhöhen (gegen 504)

```
proxy_read_timeout 300s;
```

WebSocket-Verbindungen durchreichen

```
proxy_http_version 1.1; proxy_set_header Upgrade  
$http_upgrade; proxy_set_header Connection "upgrade";
```

Zwei Backends als Upstream-Gruppe bündeln

```
upstream app { server 10.0.0.11:8080; server  
10.0.0.12:8080; }
```

Rate Limiting, Zugriff & gzip

Request-Limit pro IP definieren (im http-Block)

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

Limit anwenden, kurze Spitzen erlauben

```
limit_req zone=perip burst=20 nodelay;
```

Nur ein Netz zulassen, alle anderen sperren

```
allow 192.168.1.0/24; deny all;
```

! Passwortdatei für Basic Auth anlegen (-c überschreibt!)

```
sudo htpasswd -c /etc/nginx/.htpasswd admin
```

gzip-Kompression einschalten

```
gzip on;
```

Zusätzliche MIME-Typen komprimieren

```
gzip_types text/css application/javascript  
application/json image/svg+xml;
```

Browser-Cache für statische Dateien setzen

```
expires 30d;
```

Debugging

Server-Block lokal per Host-Header testen

```
curl -sI -H 'Host: example.com' http://127.0.0.1
```

Redirect-Kette nachverfolgen

```
curl -sIL http://example.com | grep -iE 'HTTP/|location'
```

Debug-Logging aktivieren (nur kurz, sehr geschäftig)

```
error_log /var/log/nginx/error.log debug;
```

Prüfen, auf welchen Ports nginx lauscht

```
sudo ss -tlnp | grep nginx
```

Wer belegt Port 80? (bei „address already in use“)

```
sudo lsof -iTCP:80 -sTCP:LISTEN
```

Rechte entlang des Pfads prüfen (bei 403)

```
sudo namei -l /var/www/example.com/index.html
```