nginx Web server and proxy

CONCEPTS

Always run nginx -t before reload: a broken config is rejected and the old one keeps running.

main user, workers, pid
http shared web settings
server one virtual host
location one URL path

Directives are inherited inward; the innermost value wins

PROCESS CONTROL

`nginx -t`
Check config syntax

`nginx -s reload`
Reload config gracefully

`nginx -s quit`
Graceful shutdown

SERVER BLOCKS

`listen 80;`
Port to accept connections on

`listen 443 ssl;`
Accept TLS on this port

`server_name example.com www.example.com;`
Hosts this server answers

`root <dir>;`
Document root; URI is appended

`index index.html index.htm;`
File served for a directory

`include <file>;`
Pull in more config files, globs allowed

`client_max_body_size 10m;`
Max upload size (default 1m, else 413)

LOCATIONS

`location / { }`
Prefix match: the catch-all

LOCATIONS (cont.)

`location = /path { }`
Exact match, stops the search

`location ~ \.php$ { }`
Case-sensitive regex

`alias <dir>/;`
Replace the location prefix with a path

`try_files $uri $uri/ =404;`
Try file, directory, then 404

`try_files $uri /index.html;`
SPA fallback to the index page

`return 301 <url>;`
Redirect permanently

Location matching order

Step	Rule
1	Exact = match ends the search
2	Longest prefix is remembered; ^~ stops here
3	Regex ~ and ~* in file order, first wins
4	No regex hit: first matching predicate $var location (1.31.5+)
5	Nothing else matched: use the remembered prefix

REVERSE PROXY

`proxy_pass http://127.0.0.1:8080;`
Forward requests to a backend

`proxy_set_header Host $host;`
Pass the original host to the backend

`proxy_set_header X-Real-IP $remote_addr;`
Tell the backend the client IP

TLS

`ssl_certificate <file>;`
Certificate (leaf first, then chain)

`ssl_certificate_key <file>;`
Private key, PEM

`return 301 https://$host$request_uri;`
Redirect HTTP to HTTPS