DevSheets

Nginx

nginx 1.30: server blocks, locations, TLS and redirects.

01 Install & version

```
sudo apt install nginx      # Debian/Ubuntu
sudo dnf install nginx      # RHEL/Fedora
brew install nginx          # macOS
docker run -d -p 80:80 nginx
nginx -v                    # version
nginx -V                    # build, modules
```

Linux packages run nginx as a systemd service. nginx.org ships a stable line (1.30) and a mainline (1.31).

02 Control & config test

```
sudo nginx -t                # test config
sudo nginx -T                # test + dump
sudo systemctl reload nginx  # apply changes
sudo nginx -s reload         # via signal
sudo nginx -s quit           # graceful stop
sudo nginx -s stop           # fast stop
sudo nginx -s reopen         # reopen logs
```

Reload checks the config, starts new workers and lets old ones finish; a broken config keeps the old one running.

03 nginx.conf layout

```
worker_processes auto;  # one per core
events {
    worker_connections 4096;
}
http {
    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;
}
```

Debian and Ubuntu enable a site by symlinking it from sites-available/ into sites-enabled/. Logs: /var/log/nginx/.

04 Server block

```
server {
    listen 80;
    server_name example.com www.example.com;
    root /var/www/example.com;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

The Host header picks the server by server_name. Other hosts go to the first server, or to listen 80 default_server.

05 Location matching

`= /login`	Exact match; search stops
`^~ /static/`	Prefix that skips regexes
`~ \.php$`	Regex, case-sensitive
`~* \.(png|jpg)$`	Regex, case-insensitive
`/api/`	Prefix: the longest wins

nginx keeps the longest matching prefix, then tries regexes in file order and takes the first hit. No regex hit: that prefix wins.

06 try_files & PHP-FPM

```
location / {
    try_files $uri $uri/ /index.php?$args;
}
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/run/php/php-fpm.sock;
}
```

SPAs end try_files with /index.html. fastcgi-php.conf ships with Debian's nginx; the socket name follows the PHP install.

07 HTTPS server

```
server {
    listen 443 ssl;
    http2 on;
    server_name example.com;
    ssl_certificate     /path/fullchain.pem;
    ssl_certificate_key /path/privkey.pem;
}
```

listen 443 ssl http2 is deprecated since 1.25.1 (use http2 on); ssl on is gone. TLSv1.2 and TLSv1.3 are on by default.

08 Let's Encrypt (Certbot)

```
sudo apt install certbot \
    python3-certbot-nginx
sudo certbot --nginx -d example.com \
    -d www.example.com
sudo certbot renew --dry-run
```

Certbot edits the server block. Let's Encrypt dropped OCSP in 2025, so ssl_stapling does nothing for its certificates.

09 Redirects & rewrites

```
server {
    listen 80;
    return 301 https://$host$request_uri;
}
location = /old { return 301 /new; }
location /blog/ {
    rewrite ^/blog/(.*)$ /news/$1 permanent;
}
```

Prefer return for plain redirects. rewrite flags: last searches locations again, break stays in this location, redirect is 302, permanent 301.

Nginx

Proxying, load balancing, caching and security.

10 Reverse proxy

```
location / {
    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;
}
```

Since 1.29.7 nginx proxies over HTTP/1.1 with keepalive. A level with its own proxy_set_header inherits none. Bodies over 1m get 413: raise client_max_body_size.

11 proxy_pass & the slash

```
location /api/ {   # /api/x -> /x
    proxy_pass http://localhost:8080/;
}
location /app/ {   # /app/x -> /app/x
    proxy_pass http://localhost:8080;
}
```

A URI in proxy_pass, even /, replaces the matched prefix; with none the URI passes as is. Regex locations take none.

12 WebSockets

```
location /ws/ {
    proxy_pass http://localhost:8081;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}
```

For mixed traffic, map $http_upgrade to upgrade or close. Idle sockets close after 60s (proxy_read_timeout).

13 Load balancing

```
upstream backend {
    least_conn;
    server 10.0.0.1:8080 weight=3;
    server 10.0.0.2:8080 max_fails=3;
    server 10.0.0.3:8080 backup;
}
```

Use it with proxy_pass http://backend. Round robin is the default; ip_hash pins clients. health_check is NGINX Plus only.

14 Proxy cache

```
proxy_cache_path /var/cache/nginx
    keys_zone=app:10m max_size=1g;
location / {
    proxy_cache app;
    proxy_cache_valid 200 10m;
    add_header X-Cache-Status
        $upstream_cache_status;
}
```

proxy_cache_path goes in http. Add proxy_cache_use_stale error timeout; to serve old copies when the backend fails.

15 Static files & gzip

```
location ~* \.(css|js|png|jpg|svg|woff2)$ {
    expires 30d;
    access_log off;
}
gzip on;
gzip_types text/css application/javascript;
```

text/html is always gzipped. Gotcha: an add_header in a location drops the server's add_header lines; add_header_inherit merge; (1.29.3+) keeps them.

16 if, map & geo

```
geo $blocked {
    default    0;
    1.2.3.0/24 1;
}
if ($blocked) { return 403; }
if ($http_user_agent ~* bot) { return 403; }
```

geo and map (http level) set variables. Inside a location, if is only safe with return or a rewrite using last.

17 Rate limiting

```
limit_req_zone $binary_remote_addr
    zone=api:10m rate=10r/s;
location /api/ {
    limit_req zone=api burst=20 nodelay;
    limit_req_status 429;
}
```

The zone goes in http. burst queues extra requests; nodelay serves them at once. Rejects get 503 by default.

18 Access control & headers

```
server_tokens off;
add_header X-Content-Type-Options
    "nosniff" always;
location /admin/ {
    allow 10.0.0.0/8;
    deny all;
    auth_basic "Admin";
    auth_basic_user_file .htpasswd;
}
```

Rules run in order. With both, a client needs an allowed IP and a password. Create /etc/nginx/.htpasswd with htpasswd -c.