nginx QuickRef

Recipes for the vhosts on this box: flat files, proxying local services to subdomains, and PHP apps through php-fpm. Vhosts live in `/etc/nginx/sites-available/` symlinked into `sites-enabled/`. Certbot owns the `listen 443` / port-80 redirect boilerplate, only the location logic is shown here. After any edit:

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

Flat files

```
server {
  listen 443 ssl;
  server_name files.kyaruc.moe;
  root /var/www/files;
  index index.html;

  location / {
    try_files $uri $uri/ =404;
  }
}
```

`try_files` checks each argument in order and uses the first hit; the last argument (`=404` here) is the fallback. Important: when a hit is found, the request is processed **in the current location's context**. Normally that means the static handler, but if the location also declares `proxy_pass`, the hit gets proxied. See the gotcha below.

Proxy a local port to a subdomain

For anything listening on `127.0.0.1:<port>` (web UIs, gitea, synapse, ...):

```
server {
  listen 443 ssl;
  server_name app.kyaruc.moe;

  location / {
    proxy_pass http://127.0.0.1:8080;
    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_pass http://ip:port;` with no URI part passes the original request URI through

untouched. Adding a URI part (`http://ip:port/`, the slash counts) makes nginx replace the matched location prefix: `location /app/` + a trailing slash sends `/bar` upstream instead of `/app/bar`.
- Forward the Host header, or apps that build absolute URLs generate links pointing at the backend origin instead of the subdomain.
- WebSocket endpoints additionally need:

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

PHP through php-fpm

One php block serves any app; only `root` and app-specific rewrites change per vhost:

```
server {
  listen 443 ssl;
  server_name php.kyaruc.moe;
  root /var/www/app;
  index index.php;

  location / {
    try_files $uri $uri/ /index.php?$args;
  }

  location ~ \.php$ {
    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    try_files $uri =404;
    fastcgi_pass unix:/run/php/php8.4-fpm.sock;
  }
}
```

- `SCRIPT_FILENAME` is the one variable Debian's stock `fastcgi_params` does not set. Missing it gives `Primary script unknown` in the fpm log.
- `try_files $uri =404` inside the php block stops fabricated `.php` URIs from reaching fpm.
- The socket name tracks the distro PHP version (`/run/php/php8.4-fpm.sock`). After a distro PHP upgrade the old socket vanishes and every site 502s until the vhosts are updated. `connect() failed` in the error log is the tell.
- FreshRSS and most flat PHP apps need nothing more than this block. Only apps with custom routing need rewrites.

DokuWiki

The working memex vhost. `@dokuwiki` is a named location carrying the pretty-URL rewrites:

```
server {
  server_name memex.kyaruc.moe;
```

```
  root /var/www/dokuwiki;

  location / {
    index doku.php;
    try_files $uri $uri/ @dokuwiki;
  }

  location ~ ^/lib.*\.(gif|png|ico|jpg)$ { expires 30d; }

  location ^~ /conf/ { return 403; }
  location ^~ /data/ { return 403; }

  location @dokuwiki {
    rewrite ^/_media/(.*) /lib/exe/fetch.php?media=$1 last;
    rewrite ^/_detail/(.*) /lib/exe/detail.php?media=$1 last;
    rewrite ^/_export/([^/]+)/(.*) /doku.php?do=export_$1&id=$2 last;
    rewrite ^/(.*) /doku.php?id=$1 last;
  }

  location ~ \.php$ {
    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_pass unix:/run/php/php8.4-fpm.sock;
  }
}
```

`/conf/` and `/data/` are blocked outright; media still flows through `/_media/` into fetch.php, so nothing under data/ needs to be web-accessible.

Location matching order

- `= /path` – exact match, wins over everything, search ends. Also (with `location = /`) skips prefix checks entirely.
- `^~ /path` – prefix match; if it is the longest matching prefix, regex locations are skipped.
- Otherwise nginx remembers the longest matching prefix, then tries regexes (~ sensitive, ~* insensitive) **in file order**; the first regex hit wins even over a longer prefix.

Consequence seen on memex: `/doku.php` matches regex `\.php$` and goes straight to fpm, while `/` stays in the prefix `location /`. The two entry points of the same vhost can behave completely differently, which is exactly how the gotcha below hid for a while.

Gotcha: proxy_pass to the vhost's own server_name

Symptom: one browser request floods the access log with dozens of same-second `GET / HTTP/1.1` lines from the **server's own IP** (nginx-to-nginx hops speak HTTP/1.1), ending in a 500 whose body is ~177 bytes, nginx's stock error page rather than PHP output. Direct PHP paths like `/doku.php` keep working because the regex php location outranks the prefix one.

Cause: a line like `proxy_pass https://memex.kyaruc.moe;` left inside `location /`. `try_files $uri $uri/` matches `$uri/` (the docroot directory exists), and that hit is processed in the current location's context, i.e. proxied right back to the same vhost, which proxies it again. Loop until nginx gives up.

Fix: delete it. `proxy_pass` is only for real backends, and those are always a different host or `127.0.0.1:port`, never the `server_name` of the vhost the directive lives in. A locally-served static + php-fpm site does not want `proxy_pass` at all.

Debug

```
curl -skI https://host/                    # headers: which layer answered
curl -sk https://host/ -o- | head -c 300   # tiny body == nginx stock error
page
tail -n50 /var/log/nginx/error.log
```

Error log phrases and what they mean:

- `rewrite or internal redirection cycle` – try_files/rewrite loop; nginx aborts with 500 after 10 hops.
- `connect() failed` on the fpm socket – wrong socket path or the pool is down (client sees 502).
- `Primary script unknown` – SCRIPT_FILENAME unset or wrong `root`.
- `worker_connections are not enough` – often a proxy loop eating connections.

Set `access_log` and `error_log` inside each server block to keep vhosts separated; memex does this.

From:
https://memex.kyaruc.moe/ - **kyaruc memex**

Permanent link:
https://memex.kyaruc.moe/quickref:nginx

Last update: **2026-11-23 Mon 18:05**