Nginx

Nginx (read engine X) is a light weight webserver, reverse proxy and load balancer. It can be an alternative to Apache when using PHP-FPM protocol or can be a proxy to Apache.

Config examples

PHP FPM

Nginx provide modules to communicate to PHP-FPM (FastCGI Process Manager). PHP-FPM can listen for connections using TCP port or sockets. The following example demonstrates a Nginx config file to deploy Tiki using PHP-FPM.

Enforcing HTTPS # To enforce insecure connections to Tiki, you can make Nginx redirect all traffic # from http:// to https://. But note that: # 1) It is important to check https configuration in tiki preferences to avoid # conflicts. Tiki needs to allow https everywhere (it does by default). # The most common issue in case of configuration conflict is problems to login into Tiki. # 2) Tiki can do the equivalent of this in it's own preference (pref: session_protected) # #server { # listen 80; # server_name example.com; # return 301 https://$server_name$request_uri; #} server { listen 80; #Comment this out if enabling forced https above # http2 allows http multiplexing and is important for performance for tiki # that inevitably serves a lot of small files that can't be bundled together. # And yes, it means serving over https is faster. listen 443 ssl http2; server_name example.com; #tiki.local is a typical choice for local development. You then add 127.0.0.1 tiki.local to /etc/hosts ssl_certificate /etc/nginx/ssl/example.com.crt; ssl_certificate_key /etc/nginx/ssl/example.com.key; root /var/www/html; #Change this to where you have tiki installed index tiki-index_raw.php index.php index.html; location / { # Use route.php to have SEO-friendly URLs try_files $uri $uri/ /route.php?q=$uri&$args; } location ~ \.(bak|exe|inc|ini|lib|pl|py|sh|sql|tpl)$ { deny all; } # Prevent browsing of .git and node_modules content location ~ /(?:\.git|node_modules)/ { return 403; } location ~ \.php$ { #Fastcgi default nginx config https://blog.martinfjordvald.com/nginx-config-history-fastcgi_params-versus-fastcgi-conf/ include fastcgi.conf; fastcgi_split_path_info ^(.+\.php)(/.+)$; fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name; # Avoid issues with HTTP header injections in PHP-FPM See https://httpoxy.org/ for more information. fastcgi_param HTTP_PROXY

""; # With php5-cgi alone #fastcgi_pass 127.0.0.1:9000; # With php5-fpm: # Except in recent ubuntu and debian, this path used to change for every php version, so you may have to change it fastcgi_pass unix:/var/run/php/php-fpm.sock; fastcgi_index index.php; #Tiki serves huge files, and this is probably not where you want to set a lower limit client_max_body_size 2000M; } error_page 500 502 503 504 /50x.html; location = /50x.html { root /usr/share/nginx/html; } # MIME Types. As of 2024-08-13 this is required even on very recent ubuntu to serve native esm modules with a mjs extension. types { application/javascript mjs; } # Performance tweaks # Gzip compression is crucial for performance, especially for css and js files gzip on; gzip_types text/plain text/css application/xml application/javascript image/svg+xml }
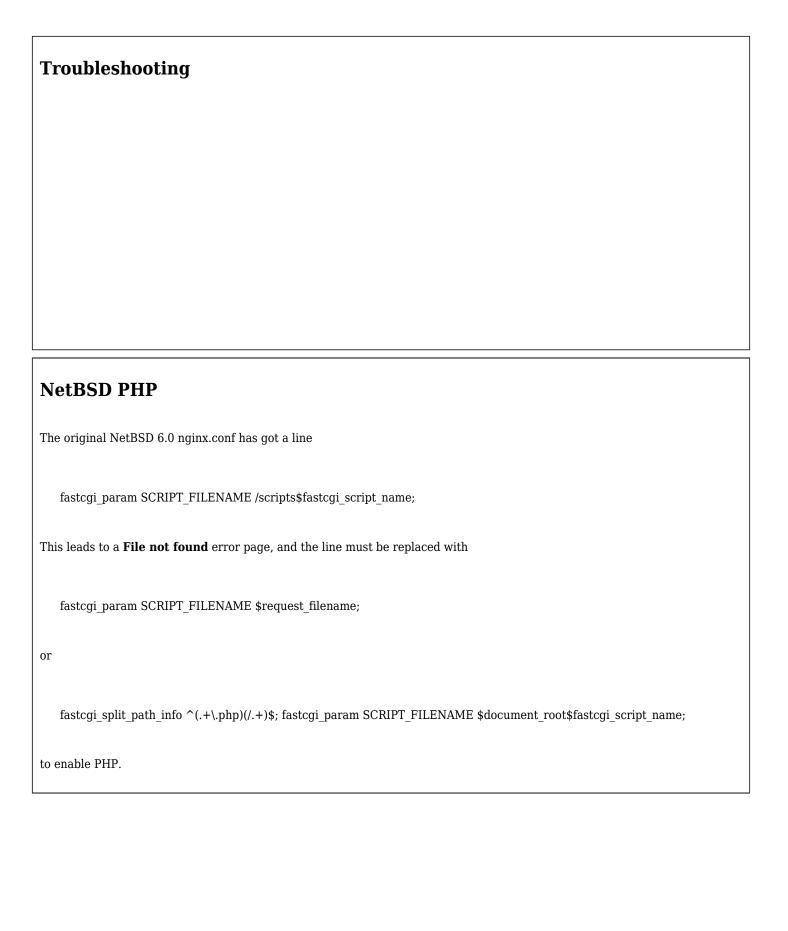
Proxying Apache

Sometimes, Tiki deployments are too coupled to `.htaccess` file and it is not possible to use the PHP-FPM and Nginx only. In this case it is possible to use Nginx as a reverse proxy to Apache. Nginx can directly deliver to browser static files and send to Apache just the requests to PHP files. The next example demonstrates this idea, supposing Apache is running on port 8080.

server { listen 80; listen 443 ssl; server_name example.com ssl_certificate /etc/nginx/ssl/example.com.crt; ssl_certificate_key /etc/nginx/ssl/example.com.key; root /var/www/html; index tiki-index_raw.php index.php index.html; location / { # Use route.php to have SEO-friendly URLs try_files $uri $uri/ /route.php?q=$uri&$args; } location ~ \.(bak|exe|inc|ini|lib|pl|py|sh|sql|tpl)$ { deny all; } # Prevent browsing of .git and node_modules content location ~ /(?:\.git|node_modules)/ { return 403; } location ~ \.php$ { proxy_set_header Host $http_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://127.0.0.1:8080; } error_page 500 502 503 504 /50x.html; location = /50x.html { root /usr/share/nginx/html; } }

Performance

The base php-fpm on debian based distributions works fine. But you may want to tweak it if you have a lot of traffic and a server with significant unused CPU and RAM resources. On ubuntu the active one will be in /etc/php/8.1/fpm/pool.d/www.conf or similar.

Troubleshooting

NetBSD PHP

The original NetBSD 6.0 nginx.conf has got a line

```
fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
```

This leads to a **File not found** error page, and the line must be replaced with

```
fastcgi_param SCRIPT_FILENAME $request_filename;
```

or

```
fastcgi_split_path_info ^(.+\.php)(/.+)$; fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
```

to enable PHP.

Debian PHP

The same error as in NetBSD was observed as Debian 6.0 (Squeeze), the file is: /etc/nginx/sites-available/default.

The problem is solved by using dotdeb:
http://www.howtoforge.com/installing-php-5.3-nginx-and-php-fpm-on-ubuntu-debian

PHP General

Nginx-php-fpm

- http://www.stevestreeting.com/2012/05/09/apache-to-nginx-php-fpm-part-1/
- http://www.stevestreeting.com/2012/05/24/apache-to-nginx-part-2/