**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 demostrates a Nginx config file to deploy Tiki using PHP-FPM.

```nginx
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.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;
    }

    location ~ \.php$ {
        include fastcgi_params;

        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;

        # Avoid issues with HTTP header injections
        fastcgi_param HTTP_PROXY "";

        fastcgi_pass 127.0.0.1:9000;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }
}
```
The reason to set `HTTP_PROXY ""` is to avoid issues with HTTP header injections (https://httpoxy.org/).

**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.

```nginx
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.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;
    }

    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;
    }
}
```

**Enforcing HTTPS**

We can forbid insecure connections to Tiki by making Nginx redirect all content from `http://` to `https://`. But it is important to check Tiki configuration to avoid conflicts. Tiki also needs to be setup to allow https everywhere. The most known issue about misconfiguration is problems to login to Tiki.
server {
    listen 80;
    server_name example.com;
    return 301 https://$server_name$request_uri;
}

server {
    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.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;
    }

    location ~ \.php$ {
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;

        # Avoid issues with HTTP header injections
        fastcgi_param HTTP_PROXY "";

        fastcgi_pass 127.0.0.1:9000;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }
}

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_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:


**PHP General**

**Nginx-php-fpm**

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