

eZ Publish 4.2011 Community (4.5) with Nginx and PHP-FPM on Debian 6 (“Squeeze”)

By Daniel Arnrup-Øien
Open Concept - <http://openconcept.no>

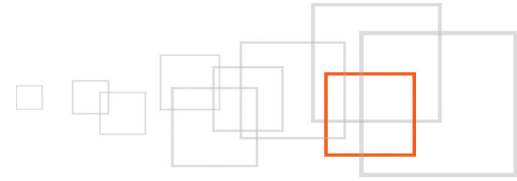
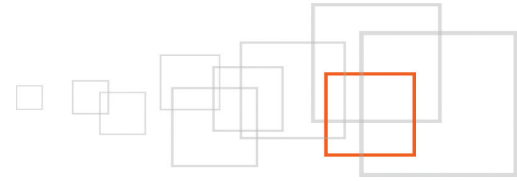


Table of Contents

Goal description.....	3
Introduction.....	3
Pre-requisites and target population.....	3
Step 1: Update Debian package lists.....	3
Step 2: Install required packages.....	4
Step 3: Configure PHP-FPM.....	4
Step 4: Configure Nginx.....	5
Step 5: Reload PHP and Nginx.....	7
Step 6: Install or move eZ Publish.....	7
Step 7: Update eZ Publish site settings.....	7
Step 8: Flush caches.....	7
Conclusion.....	8
Resources.....	8
About the author : Daniel Arnrup-Øien.....	8
License choice.....	8



Goal description

After completing this tutorial, you should have a working eZ Publish 4.2011 Community (or 4.5) installation running on a single Debian 6 server with the following features:

- Nginx 1.0.0+ as web server
- PHP 5.3.6+ invoked through PHP-FPM (FastCGI Process Manager)
- APC (Alternative PHP Cache)
- Rewrites for friendly URLs (virtual host mode)
- Proper http cache headers for static files
- Gzip compression

Introduction

Nginx is an event-driven, lightweight and high-performance HTTP server and reverse proxy/load balancer. With the recent release of Nginx 1.0.0, PHP-FPM 5.3.6 and improved documentation, it is now simpler to install and configure Nginx to serve eZ Publish sites with the same features as Apache HTTP Server while receiving the performance and memory consumption benefits of Nginx.

The tutorial covers the setup and basic configuration of Nginx, PHP/MySQL and eZ Publish 4.2011 Community Edition (4.5 Enterprise) on Debian 6. The steps can also be used as an outline for performing the same setup on other versions or distributions.

Pre-requisites and target population

Technical pre-requisites are a basic knowledge of web servers, Debian package installation using apt or similar, and some experience with installing or moving/upgrading eZ Publish.

The reference system for this tutorial is a clean network install of Debian 6 with admin tools only, running in a virtual machine with two 2-Ghz processors, 2 GB RAM and 16 GB hard disk space.

NOTE: *All operations are assumed to be run as the root user unless otherwise specified.*

Step 1: Update Debian package lists

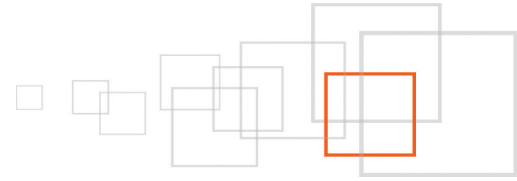
Nginx 1.0.0 and PHP-FPM (FastCGI Process Manager) 5.3.6 is available through dotdeb, and your apt sources file should be updated accordingly.

Add the following to your `/etc/apt/sources.list` file:

```
deb http://packages.dotdeb.org stable all
deb-src http://packages.dotdeb.org stable all
```

Update the dotdeb key:

```
wget http://www.dotdeb.org/dotdeb.gpg
cat dotdeb.gpg | apt-key add -
```

**Run:**

```
apt-get update
```

Step 2: Install required packages

```
apt-get install nginx
apt-get install mysql-server
apt-get install php5-cli php5-common php5-suhosin
apt-get install php5-fpm php5-cgi php5-curl php5-gd php5-imagick php5-mysql php5-apc
imagemagick
```

Step 3: Configure PHP-FPM

Update the files listed below. You need to substitute local variables where noted.

In /etc/php5/fpm/pool.d/www.conf:

```
listen = /tmp/php-fpm.sock
```

Since this is a single-server installation, we will configure PHP-FPM to listen on a unix socket as opposed to a tcp/ip socket, thus bypassing the tcp/ip stack.

The socket file can be placed anywhere you like – this is just an example for convenience. Just make sure it's in a writable directory, and remember that Nginx has to be pointed to the same location in Step 4.

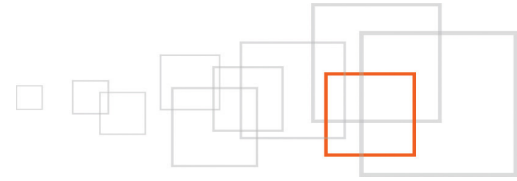
In /etc/php5/fpm/php.ini:

```
date.timezone = "YOUR TIME ZONE, such as Europe/Oslo"
short_open_tag = On
memory_limit = 256M
max_execution_time = 120
cgi.fix_pathinfo=0
```

NOTE: *the last setting is very important as it prevents a security issue when running PHP in cgi mode.*

In /etc/php5/cli/php.ini:

```
date.timezone = "YOUR TIME ZONE, such as Europe/Oslo"
max_execution_time = 120
```



Step 4: Configure Nginx

Now we need to configure Nginx to run with PHP over FastCGI.

Update the files listed below. You need to substitute local variables where noted.

In /etc/nginx/nginx.conf:

```
user www-data;
worker_processes 4;
pid /var/run/nginx.pid;

events {
    worker_connections 2048;
    # multi_accept on;
}

http {
    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    server_tokens off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # Logging Settings
    ##

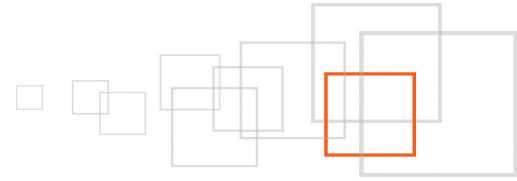
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    ##
    # Gzip Settings
    ##

    gzip on;
    gzip_vary on;
    gzip_disable "msie6";
    gzip_proxied any;
    gzip_comp_level 5;
    gzip_buffers 32 8k;
    gzip_http_version 1.1;
    gzip_types text/plain text/css application/json application/x-javascript text/xml
    application/xml application/xml+rss text/javascript;

    ##
    # Virtual Host Configs
    ##

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;
}
```



In /etc/nginx/fastcgi_params:

```
fastcgi_connect_timeout 60;
fastcgi_send_timeout 180;
fastcgi_read_timeout 180;
fastcgi_buffer_size 128k;
fastcgi_buffers 4 256k;
fastcgi_busy_buffers_size 256k;
fastcgi_temp_file_write_size 256k;
fastcgi_intercept_errors on;
```

The above are standard settings to prevent PHP-FPM from choking, and you may want to experiment with tuning these to your particular environment. They serve as a good starting point for most installations.

In /etc/nginx/sites-available/default:

```
server {
    server_name YOUR.INTERNET.HOSTNAME;
    root YOUR_EZPUBLISH_ROOT;

    index index.php;

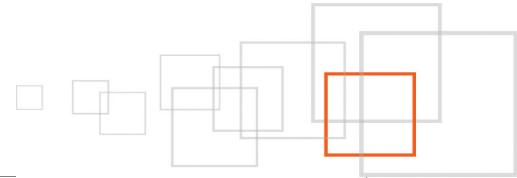
    location / {

        rewrite "^/var/storage/(.*)$" "/var/storage/$1" break;
        rewrite "^/var/([^/]+)/storage/(.*)$" "/var/$1/storage/$2" break;
        rewrite "^/var/(([/]+/)?cache/(texttoimage|public)/(.*))$" "/var/$1cache/$3/$4" break;
        rewrite "^/design/([^/]+)/ (stylesheets|images|javascript)/(.*)$" "/design/$1/$2/$3"
        break;
        rewrite "^/share/icons/(.*)$" "/share/icons/$1" break;
        rewrite "^/extension/([^/]+)/design/([^/]+)/ (stylesheets|images|javascripts|javascript|
        flash|lib?)/(.*)$" "/extension/$1/design/$2/$3/$4" break;
        rewrite "^/packages/styles/(+)/ (stylesheets|images|javascript)/([/]+)/ (.*)$"
        "/packages/styles/$1/$2/$3/$4" break;
        rewrite "^/packages/styles/(+)/thumbnail/(.*)$" "/packages/styles/$1/thumbnail/$2"
        break;
        rewrite "^/favicon\.ico$" "/favicon.ico" break;
        rewrite "^/robots\.txt$" "/robots.txt" break;
        rewrite "^/var/cache/debug.html(.*)" "/var/cache/debug.html$1" break;
        rewrite "^/var/(([/]+/)?cache/public/(.*))$" "/var/$1cache/public/$3" break;
        rewrite "^/var/([^/]+)/cache/debug\.html(.*)" "/var/$1/cache/debug.html$2" break;
        rewrite "content/treemenu/?$" "/index_treemenu.php" break;
        rewrite "ezjscore/call/?$" "/index_ajax.php" break;
        rewrite "(.*)$" "/index.php?$1" last;
    }

    location ~* ^.+.(jpg|jpeg|gif|css|png|js|ico|swf|flv|eot|ttf|woff|svg)$ {
        access_log off;
        expires 30d;
    }

    location ~ "[^/]*\.php$" {
        set $script "index.php";
        if ( $uri ~ "[^/]*(.*)\.php" ) {
            set $script $1;
        }

        fastcgi_pass unix:/tmp/php-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$script;
    }
}
```



```
include    fastcgi_params;
}
}
```

Replace YOUR.INTERNET.HOSTNAME and YOUR_EZPUBLISH_ROOT as necessary.

You can also enter these settings in a file with any name in /etc/nginx/sites-available and create a symlink to it in /etc/nginx/sites-enabled if you want to run multiple vhosts. The principles are the same as under Apache 2.

Special thanks to Boris Huisgen for providing the basic rewrite rules for eZ Publish under Nginx.

Step 5: Reload PHP and Nginx

```
/etc/init.d/php5-fpm restart
/etc/init.d/nginx restart
```

Step 6: Install or move eZ Publish

Download and unpack the eZ Publish distribution to the chosen web root directory (YOUR_EZPUBLISH_ROOT) and install as normal. You can also move an existing installation, but it should be upgraded to eZ Publish 4.5 or 4.2011 Community Edition because of issues with fastcgi in earlier versions.

See <http://doc.ez.no> for instructions on how to install or upgrade eZ Publish.

Step 7: Update eZ Publish site settings

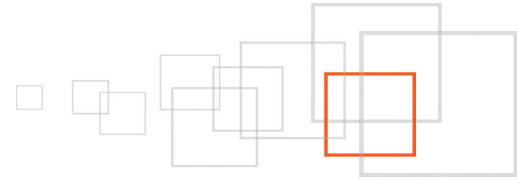
eZ Publish doesn't automatically recognise virtual host mode on Nginx as opposed to Apache.

When you are done installing/moving, in settings/override/site.ini.append.php, add the following under the [SiteAccessSettings] block:

```
ForceVirtualHost=true
```

Step 8: Flush caches

```
su - www-data
cd YOUR_EZ_PUBLISH_ROOT
php bin/php/ezcache.php --clear-all --purge
```



Conclusion

At this point you should have a working eZ Publish installation under Nginx, behaving exactly the same as under Apache but with even more speed and stability, especially under high-load situations. The next natural steps are of course to experiment with the settings to optimise them for your own environments, and carry out some stress testing.

This is just the tip of the iceberg of Nginx and server constellations, and in future tutorials I hope to provide examples of clustered setups, load balancing and proxying as well as load tests.

Resources

- Nginx Wiki – <http://wiki.nginx.org>
- A more generic howto - <http://www.howtoforge.com/installing-php-5.3-nginx-and-php-fpm-on-ubuntu-debian>
- Another howto - <http://interfacelab.com/nginx-php-fpm-apc-awesome/>

About the author : Daniel Arnrup-Øien



Daniel Arnrup-Øien is Board Director and developer at Open Concept SA, a Norwegian cooperative corporation dedicated to providing complete open source web solutions with quality and performance. He has been developing in eZ Publish since 2005. He can be reached at daniel.oien@openconcept.no.

License choice



This work is licensed under the Creative Commons Attribution-ShareAlike (CC BY-SA) license.