

## Nginx HTTP Server + PHP5 (With fast-cgi And xcache) On Ubuntu Feisty Fawn

*By Horus*

Published: 2007-10-08 12:55

## Nginx HTTP Server + PHP5 (With fast-cgi And xcache) On Ubuntu Feisty Fawn

This HowTo describes the implementation of Nginx with php5 support (through FastCGI). The fast-cgi process will be initiated via spawn-fcgi.

What for? Nginx is a great replacement of Apache with very low memory foot print and contrary to Lighttpd, doesn't suffer from memory leak over time. You can then use all the memory left to unleash the power of mysql for instance by increasing the default query cache.

We have first to install php5 and, because we will need it later, build-essential.

```
> apt-get install php5 php5-common php5-cgi php5-xcache php5-dev build-essential
```

Note that xcache has to be implemented manually by adding the following line in the *php.ini* located in */etc/php5/cgi/*.

```
extension = xcache.so
```

You can do that right now even if your php configuration isn't loaded yet so everything will be in good order when Nginx and fcgi process will be started.

Next: we get the latest stable version of Nginx. The one proposed by Feisty is prehistoric one. Fortunately, there's a place you can get the latest stable version, or if you are adventurous, the latest dev version. So here we go:

```
> wget http://technokracy.net/Nginx/Nginx_0.5.32-grrr-1_i386.deb
```

(Note that if you are running on AMD replace i386 by amd64.)

Then:

```
> sudo dpkg -i Nginx_0.5.32~grrr-1_i386.deb
```

Nginx is now up and running on default port 8080 (just in case you already have Apache or anything else on port 80).

The default root folder is Nginx-default and is located in `/var/www/`.

To change that and to start to listen to the fast-cgi we will launch next, you have to open `/etc/Nginx/sites-available/default`.

```
> vi /etc/Nginx/sites-available/default
```

You can find there all the obvious options to change and add (or uncomment the original php paragraph):

```
location ~ \.php$ {
    include /etc/Nginx/fastcgi_params;
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME /var/www/Nginx-default$fastcgi_script_name;
}
```

We've just asked Nginx to listen to fcgi on port 9000. So we have to start now the fcgi process. I've chosen to use spawn-fcgi and to make my own init script of it (so the process will start after reboot). To have spawn-fcgi you have to get lighttpd configured but without the need to install it. Let's grab the latest version:

```
> wget http://www.lighttpd.net/download/lighttpd-1.4.18.tar.bz2
```

```
> tar -xvjf lighttpd-1.4.18.tar.bz2
```

```
> cd lighttpd-1.4.18
```

```
> ./configure
```

```
> make
```

```
> cp src/spawn-fcgi /usr/bin/spawn-fcgi
```

Note that we did not type `make install` so lighttpd is not running!

Then we create a shell script we can call `php-fastcgi` or whatever you want and place that file in `/usr/bin/` to make it simple (as `php5-cgi` and `spawn-fcgi` are already there...).

```
> touch /usr/bin/php-fastcgi
```

Then edit it:

```
> vi /usr/bin/php-fastcgi
```

and add the following:

```
#!/bin/sh
/usr/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -u www-data -f /usr/bin/php5-cgi
```

That means every time this script will be called, `fcgi` will be spawned on port 9000 for user `www-data` (default user).

To make it work at startup we need now to create an init script:

```
> touch /etc/init.d/init-fastcgi
```

Edit and add:

```
> vi /etc/init.d/init-fastcgi
```

```
#!/bin/bash
PHP_SCRIPT=/usr/bin/php-fastcgi
RETVAL=0
case "$1" in
  start)
    $PHP_SCRIPT
    RETVAL=$?
    ;;
  stop)
    killall -9 php
    RETVAL=$?
    ;;
  restart)
    killall -9 php
    $PHP_SCRIPT
    RETVAL=$?
    ;;
  *)
    echo "Usage: php-fastcgi {start|stop|restart}"
    exit 1
    ;;
esac
exit $RETVAL
```

You may have to change the permissions there by typing:

```
> chmod 755 /etc/init.d/init-fastcgi
```

Check then if it works by typing:

```
> /etc/init.d/init-fastcgi start
```

You should have an answer from spawn-fcgi attributing a PID process. To make now everything working after reboot type:

```
> update-rc.d init-fastcgi defaults
```

And we are done. To check if php is working as fast-cgi you can first type:

```
> ps ax | grep php
```

To check then if Nginx is listening to php, create an echo command in an empty php file:

```
echo phpinfo()
```

and place it in your Nginx directory. You should see then all your php conf along with the xcache module.