



PHP-FPM

# Table des matières

1. Généralités .....	2
2. Installation .....	3
2.1. Debian 8 .....	3
2.2. Arrêt et relance du service .....	3
3. Configuration .....	4
3.1. Configuration statique ou dynamique .....	5
3.2. Configuration avancée .....	6
Status du processus .....	6
Journaliser les requêtes longues .....	6
3.3. Configuration avec nginx .....	6

**PHP-FPM (FastCGI Process Manager)** est intégré à PHP depuis sa version 5.3.3. La version FastCGI de php apporte des fonctionnalités complémentaires.

# Chapitre 1. Généralités

**CGI (Common Gateway Interface)** et **FastCGI** permettent la communication entre le serveur Web (Apache, Nginx) et un langage de développement (Php, Python, Java) :

- Dans le cas du **CGI**, chaque requête entraîne la création d'un **nouveau processus**, ce qui n'est pas efficace en terme de performance.
- **FastCGI** s'appuie, quant à lui, sur un **certain nombre de processus** pour le traitement de ses requêtes clientes.

PHP-FPM, apporte **en plus des meilleurs performances** :

- La possibilité de **mieux cloisonner les applications** : lancement des processus avec des uid/gid différents, avec des fichiers php.ini personnalisés,
- La gestion des statistiques,
- Gestion des journaux,
- Gestion dynamique des processus et redémarrage sans coupure de service ('graceful').



Apache possédant un module php, l'utilisation de php-fpm est moins intéressante que pour le serveur Nginx.

## Chapitre 2. Installation

### 2.1. Debian 8

L'installation de php-fpm s'effectue depuis les dépôts apt :

```
$ sudo apt-get install php5-fpm
...
Les paquets supplémentaires suivants seront installés :
  libapparmor1 libonig2 libqdbm14 php5-cli php5-common php5-json php5-readline
Paquets suggérés :
  php-pear php5-user-cache
...
```

### 2.2. Arrêt et relance du service

Via systemd, la commande suivante stoppe le service :

```
$ sudo systemctl stop php5-fpm
```

La commande suivante relance le service :

```
$ sudo systemctl restart php5-fpm
```

Pour simplement recharger la configuration et prendre les modifications effectuées en compte :

```
$ sudo systemctl reload php5-fpm
```

## Chapitre 3. Configuration

Les fichiers de configuration de php-fpm se situent sous */etc/php5/fpm*.

php-fpm utilise la syntaxe de php.ini pour ses fichiers de configuration (php-fpm.conf et fichier de configuration des pools).

Le fichier */etc/php5/fpm/php-fpm.conf*, dans sa version minimale, contient :

```
[global]
pid = /run/php5-fpm.pid
error_log = /var/log/php5-fpm.log

include=/etc/php5/fpm/pool.d/*.conf
```

Le fichier */etc/php5/fpm/pool.d/www.conf* contient, quant à lui, les quelques directives suivantes :

```
[www]
user = www-data
group = www-data
listen = /var/run/php5-fpm.sock
listen.owner = www-data
listen.group = www-data

pm = dynamic
pm.max_children = 5
pm.start_servers = 2
pm.min_spare_servers = 1
pm.max_spare_servers = 3

chdir = /
```

Table 1. Directives de la configuration par défaut

Directives	Observations
[pool]	Nom du pool de processus. Le fichier de configuration peut être composé de plusieurs pools de processus (le nom du pool entre crochet commence une nouvelle section)

Directives	Observations
listen	Définit l'interface d'écoute ou le socket unix utilisé.  Exemple : <b>listen = 127.0.0.1:9000</b>  Ou via une socket Unix : <b>listen = /var/run/php5-fpm.sock.</b>  L'utilisation d'une socket lorsque le serveur web et le serveur php sont sur la même machine permet de s'affranchir de la couche TCP/IP.
<b>Pour une interface :</b> listen.owner, listen.group, listen.mode	Spécifier le propriétaire, le groupe propriétaire et les droits de la socket Unix. Attention : les deux serveurs (web et php) doivent disposer des droits d'accès sur la socket.
<b>Pour une socket :</b> listen.allowed_clients	restreindre l'accès au serveur php à certaines adresses IP.  Exemple : <b>listen.allowed_clients = 127.0.0.1</b>

### 3.1. Configuration statique ou dynamique

Les processus de php-fpm peuvent être gérés de manière statique ou dynamique :

- En mode **static** : le nombre de processus fils est fixé par la valeur de pm.max\_children ;

*Configuration de php-fpm en mode static*

```
pm = static
pm.max_children = 10
```

Cette configuration lancera 10 processus.

- En mode **dynamic** : php-fpm lancera au maximum le nombre de processus spécifié par la valeur de pm.max\_children, en commençant par lancer un nombre de processus correspondant à pm.start\_servers, et en gardant au minimum la valeur de pm.min\_spare\_servers de processus inactifs et au maximum pm.max\_spare\_servers processus inactifs.

Exemple :

```
pm                = dynamic
pm.max_children   = 5
pm.start_servers  = 2
pm.min_spare_servers = 1
pm.max_spare_servers = 3
```



Php-fpm créera un nouveau processus en remplacement d'un processus qui aura traité un nombre de requêtes équivalent à `pm.max_requests`.

Par défaut, la valeur de `pm.max_requests` est à 0, ce qui signifie que les processus ne sont jamais recyclés. Utiliser l'option `pm.max_requests` peut être intéressant pour des applications présentant des fuites mémoires.

## 3.2. Configuration avancée

### Status du processus

Php-fpm propose, à l'instar de Apache et de son module `mod_status`, une page indiquant l'état du processus.

Pour activer la page, il faudra fournir à nginx son chemin d'accès via la directive `pm.status_path` :

```
pm.status_path = /status
```

### Journaliser les requêtes longues

La directive `slowlog` indique le fichier recevant la journalisation des requêtes trop longues (dont le temps dépasse la valeur de la directive `request_slowlog_timeout`).

Le fichier généré se situe par défaut *`/var/log/php5-fpm.log.slow`*.

```
request_slowlog_timeout = 30  
slowlog = /var/log/php5-fpm.log.slow
```

Une valeur à 0 de `request_slowlog_timeout` désactive la journalisation.

## 3.3. Configuration avec nginx

Le paramétrage par défaut de nginx intègre déjà la configuration nécessaire pour faire fonctionner php avec php-fpm.

Le fichier de configuration `fastcgi.conf` (ou `fastcgi_params`) se situe sous *`/etc/nginx/`* :

```

fastcgi_param SCRIPT_FILENAME    $document_root$fastcgi_script_name;
fastcgi_param QUERY_STRING      $query_string;
fastcgi_param REQUEST_METHOD    $request_method;
fastcgi_param CONTENT_TYPE      $content_type;
fastcgi_param CONTENT_LENGTH    $content_length;

fastcgi_param SCRIPT_NAME       $fastcgi_script_name;
fastcgi_param REQUEST_URI       $request_uri;
fastcgi_param DOCUMENT_URI      $document_uri;
fastcgi_param DOCUMENT_ROOT     $document_root;
fastcgi_param SERVER_PROTOCOL   $server_protocol;
fastcgi_param HTTPS             $https if_not_empty;

fastcgi_param GATEWAY_INTERFACE CGI/1.1;
fastcgi_param SERVER_SOFTWARE   nginx/$nginx_version;

fastcgi_param REMOTE_ADDR        $remote_addr;
fastcgi_param REMOTE_PORT        $remote_port;
fastcgi_param SERVER_ADDR        $server_addr;
fastcgi_param SERVER_PORT        $server_port;
fastcgi_param SERVER_NAME        $server_name;

# PHP only, required if PHP was built with --enable-force-cgi-redirect
fastcgi_param REDIRECT_STATUS    200;

```

Pour que nginx traite les fichiers .php, les directives suivantes doivent être ajoutées au fichier de configuration du site :

- Si php-fpm écoute sur le port 9000 :

```

location ~ /\.php$ {
    include /etc/nginx/fastcgi_params;
    fastcgi_pass 127.0.0.1:9000;
}

```

- Si php-fpm écoute sur une socket unix :

```

location ~ /\.php$ {
    include /etc/nginx/fastcgi_params;
    fastcgi_pass unix:/var/run/php5-fpm.sock;
}

```