

SAÉ 2.03 partie 3

Thomas BAERT, Robin CERISIER, Louis ROUBAUD S5

1. Introduction

Objectif du rapport

Durant ce rapport nous allons vous présenter le prototype de l'environnement à installer pour l'entreprise de coworking. Ce prototype utilisera donc la configuration retenue : Nginx, PostgreSQL, Ubuntu et PHP. Du fait de l'utilisation de Nginx nous installeront aussi PHP-FPM pour gérer l'exécution des scripts PHP.

2. Installation des services

Nginx

Après avoir mis à jour le système en utilisant :

```
thomasbaert@ubuntu:~$ sudo apt update && sudo apt upgrade
```

Installez Nginx en utilisant :

```
thomasbaert@ubuntu:~$ sudo apt install nginx
```

Vérifiez ensuite l'état de Nginx avec :

```
thomasbaert@ubuntu:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Drop-In: /run/systemd/system/service.d
            └─zzz-lxc-service.conf
   Active: active (running) since Fri 2025-06-06 22:44:18 CEST; 47s ago
     Docs: man:nginx(8)
    Process: 3434 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 3435 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 3437 (nginx)
      Tasks: 9 (limit: 4772)
     Memory: 5.1M (peak: 5.9M)
        CPU: 10ms
    CGroup: /system.slice/nginx.service
            └─3437 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              └─3438 "nginx: worker process"
                └─3439 "nginx: worker process"
                  └─3440 "nginx: worker process"
                    └─3441 "nginx: worker process"
                      └─3442 "nginx: worker process"
                        └─3443 "nginx: worker process"
                          └─3444 "nginx: worker process"
                            └─3445 "nginx: worker process"

Jun 06 22:44:18 ubuntu systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Jun 06 22:44:18 ubuntu systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
```

Si le retour affiche une erreur c'est que l'installation s'est mal passée, autrement si le service est juste en *inactive* utilisez :

```
thomasbaert@ubuntu:~$ sudo systemctl start nginx
```

Vous pouvez également utiliser *curl* ou votre navigateur pour vérifier l'état de la page. Cette commande devrait théoriquement vous retourner le code html de la page par défaut de Nginx.

```
thomasbaert@ubuntu:~$ curl http://localhost
```

PHP et PHP-FPM et PHP-PGSQL

Dans cette section nous allons aborder l'installation de PHP et de ses différentes extensions et modules.

Cette commande installera PHP qui interprètera les scripts PHP et PHP-FPM pour faire le lien entre Nginx et PHP. PHP-PGSQL sert PHP à relier la base de données PostgreSQL :

```
thomasbaert@ubuntu:~$ sudo apt install php php-fpm php-pgsql
```

On vérifie ensuite l'installation de PHP avec :

```
thomasbaert@ubuntu:~$ php -v
PHP 8.3.6 (cli) (built: Mar 19 2025 10:08:38) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
```

Si cette commande renvoie une erreur essayez de réinstaller PHP.

Cette commande permet de vérifier l'état de l'installation de PHP-FPM et s'assure que tout fonctionne correctement :

```
thomasbaert@ubuntu:~$ systemctl status php8.3-fpm
● php8.3-fpm.service - The PHP 8.3 FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php8.3-fpm.service; enabled; preset: enabled)
   Drop-In: /run/systemd/system/service.d
            └─zzz-lxc-service.conf
   Active: active (running) since Fri 2025-06-06 22:55:49 CEST; 3min 19s ago
     Docs: man:php-fpm8.3(8)
  Process: 13959 ExecStartPost=/usr/lib/php/php-fpm-socket-helper install /run/php/php-fpm.sock /etc/php/8.3/fpm/pool.d/www.conf 83 (code=e
Main PID: 13956 (php-fpm8.3)
   Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0req/sec"
    Tasks: 3 (limit: 4772)
  Memory: 7.5M (peak: 12.7M)
     CPU: 43ms
   CGroup: /system.slice/php8.3-fpm.service
           └─13956 "php-fpm: master process (/etc/php/8.3/fpm/php-fpm.conf)"
             └─13957 "php-fpm: pool www"
               └─13958 "php-fpm: pool www"

Jun 06 22:55:49 ubuntu systemd[1]: Starting php8.3-fpm.service - The PHP 8.3 FastCGI Process Manager...
Jun 06 22:55:49 ubuntu systemd[1]: Started php8.3-fpm.service - The PHP 8.3 FastCGI Process Manager.
```

Configuration de Nginx pour gérer PHP avec PHP-FPM

On utilise cette commande pour pouvoir modifier le fichier de configuration de Nginx :

```
thomasbaert@ubuntu:~$ nano /etc/nginx/sites-available/default
```

On rajoute ensuite ces lignes dans le fichier de configuration :

```
# Configuration du lien vers php-fpm
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/run/php/php8.3-fpm.sock;
}
```

Le bloc `location ~ \.php$ { ... }` indique à NGINX de traiter les fichiers PHP via FastCGI. La directive `include snippets/fastcgi-php.conf;` charge la configuration standard pour FastCGI. `fastcgi_pass unix:/run/php/php8.1-fpm.sock;` envoie les requêtes PHP au service PHP-FPM via le socket Unix.

Pour tester cette connexion on crée un fichier PHP de test comme suit :

```
thomasbaert@ubuntu:/var/www/html$ sudo touch info.php
```

On le modifie ensuite avec :

```
thomasbaert@ubuntu:/var/www/html$ sudo nano info.php
```

On recharge Nginx avec :


```
thomasbaert@ubuntu:/var/www/html$ sudo systemctl reload nginx
```

On peut ensuite vérifier la connexion avec :

```
thomasbaert@ubuntu:/var/www/html$ curl http://localhost/info.php
```

Qui, si tout est bien configuré, vous affichera le code HTML complet de la page d'informations de PHP.

Voici le rendu dans le navigateur :

PHP Version 8.3.6	
	
System	Linux ubuntu 6.14.8-orbstack-00288-g80b66077b748 #126 SMP Tue May 27 09:02:55 UTC 2025 aarch64
Build Date	Mar 19 2025 10:08:38
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/fpm
Loaded Configuration File	/etc/php/8.3/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/fpm/conf.d
Additional .ini files parsed	/etc/php/8.3/fpm/conf.d/10-opcache.ini, /etc/php/8.3/fpm/conf.d/10-pdo.ini, /etc/php/8.3/fpm/conf.d/20-calendar.ini, /etc/php/8.3/fpm/conf.d/20-ctype.ini, /etc/php/8.3/fpm/conf.d/20-exif.ini, /etc/php/8.3/fpm/conf.d/20-ffi.ini, /etc/php/8.3/fpm/conf.d/20-fileinfo.ini, /etc/php/8.3/fpm/conf.d/20-ftp.ini, /etc/php/8.3/fpm/conf.d/20-gettext.ini, /etc/php/8.3/fpm/conf.d/20-iconv.ini, /etc/php/8.3/fpm/conf.d/20-pdo_pgsql.ini, /etc/php/8.3/fpm/conf.d/20-pgsql.ini, /etc/php/8.3/fpm/conf.d/20-phar.ini, /etc/php/8.3/fpm/conf.d/20-posix.ini, /etc/php/8.3/fpm/conf.d/20-readline.ini, /etc/php/8.3/fpm/conf.d/20-shmop.ini, /etc/php/8.3/fpm/conf.d/20-sockets.ini, /etc/php/8.3/fpm/conf.d/20-sysvmsg.ini, /etc/php/8.3/fpm/conf.d/20-sysvsem.ini, /etc/php/8.3/fpm/conf.d/20-sysvshm.ini, /etc/php/8.3/fpm/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831,NTS
PHP Extension Build	API20230831,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled

Installation de PostgreSQL

Cette commande permet d'installer PostgreSQL :

```
thomasbaert@ubuntu:~$ sudo apt install postgresql postgresql-contrib
```

On se connecte à la base en utilisant cette commande (l'utilisateur par défaut est postgres)

```
thomasbaert@ubuntu:~$ sudo -u postgres psql
psql (16.9 (Ubuntu 16.9-0ubuntu0.24.04.1))
Type "help" for help.

postgres=#
```

J'utilise `sudo -u` car `psql` utilise ici le mode d'authentification peer, c'est-à-dire que l'utilisateur doit avoir pour nom `postgres`.

```
postgres=# CREATE DATABASE Test;
CREATE DATABASE
postgres=# CREATE USER TestUser WITH ENCRYPTED PASSWORD 'testpwd';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE Test TO TestUser;
GRANT
postgres=# █
```

Ces commandes créent une base de données puis un utilisateur avec toutes les permissions sur cette base de données.

Cette modification de `info.php` nous permet de tester la connexion à la base de données :

```
<?php
$conn = pg_connect("host=localhost port=5432 dbname=Test");
$result = pg_query($conn, "select * from testTable");
var_dump(pg_fetch_all($result));
?>
█
```

Ici ce script retourne une erreur puisque nous n'avons pas de table `testTable`. Cependant cette erreur indique qu'il parvient à atteindre la base de données.