

Démonstration complète : Hébergement d'applications PHP avec PHP-FPM et Apache

Cette démonstration explique comment configurer Apache pour utiliser **PHP-FPM** via **FastCGI**, permettant d'exécuter des scripts PHP avec des versions différentes sur le même serveur.

Contexte

Vous hébergez deux applications PHP :

1. Une application utilisant **PHP 7.4**.
2. Une autre application utilisant **PHP 8.1**.

Apache sera configuré pour utiliser **PHP-FPM**, avec des Virtual Hosts distincts pour chaque application.

Étape 1 : Installer les prérequis

1. Installer Apache et ajouter le dépôt PHP

Ajoutez le dépôt **Ondřej Surý** pour accéder aux versions spécifiques de PHP :

```
sudo apt update
sudo apt install software-properties-common -y
sudo add-apt-repository ppa:ondrej/php
sudo apt update
sudo apt install apache2 -y
```

2. Installer PHP 7.4 et PHP 8.1 avec leurs extensions FPM

Installez les versions requises de PHP et leurs gestionnaires FPM :

```
sudo apt install php7.4 php7.4-fpm php8.1 php8.1-fpm -y
```

3. Activer les modules Apache nécessaires

Activez les modules indispensables pour PHP-FPM :

```
sudo a2enmod proxy_fcgi setenvif
sudo a2enmod actions
sudo systemctl restart apache2
```

Étape 2 : Configurer PHP-FPM

1. Configurer les pools pour PHP 7.4 et PHP 8.1

Assurez-vous que chaque version de PHP-FPM écoute sur un socket unique :

Pour PHP 7.4 (`/etc/php/7.4/fpm/pool.d/www.conf`) :

```
[www]
listen = /run/php/php7.4-fpm.sock
```

Pour PHP 8.1 (`/etc/php/8.1/fpm/pool.d/www.conf`) :

```
[www]
listen = /run/php/php8.1-fpm.sock
```

2. Redémarrer les services PHP-FPM

Appliquez les changements en redémarrant PHP-FPM :

```
sudo systemctl restart php7.4-fpm
sudo systemctl restart php8.1-fpm
```

Étape 3 : Configurer Apache avec des Virtual Hosts

1. Créer un Virtual Host pour l'application PHP 7.4

Ajoutez cette configuration dans `/etc/apache2/sites-available/php74.example.com.conf` :

```
<VirtualHost *:80>
    ServerName php74.example.com
    DocumentRoot /var/www/php74

    <Directory /var/www/php74>
        AllowOverride All
        Require all granted
    </Directory>

    <FilesMatch \.php$>
        SetHandler "proxy:unix:/run/php/php7.4-fpm.sock|fcgi://localhost"
    </FilesMatch>

    ErrorLog ${APACHE_LOG_DIR}/php74-error.log
    CustomLog ${APACHE_LOG_DIR}/php74-access.log combined
</VirtualHost>
```

2. Créer un Virtual Host pour l'application PHP 8.1

Ajoutez cette configuration dans `/etc/apache2/sites-available/php81.example.com.conf` :

```
<VirtualHost *:80>
    ServerName php81.example.com
    DocumentRoot /var/www/php81

    <Directory /var/www/php81>
        AllowOverride All
        Require all granted
    </Directory>

    <FilesMatch \.php$>
        SetHandler "proxy:unix:/run/php/php8.1-fpm.sock|fcgi://localhost"
    </FilesMatch>

    ErrorLog ${APACHE_LOG_DIR}/php81-error.log
    CustomLog ${APACHE_LOG_DIR}/php81-access.log combined
</VirtualHost>
```

3. Activer les Virtual Hosts

Activez les Virtual Hosts et rechargez Apache :

```
sudo a2ensite php74.example.com.conf
sudo a2ensite php81.example.com.conf
sudo systemctl reload apache2
```

Étape 4 : Créer les répertoires des applications

1. Créer les dossiers

Créez les répertoires des applications :

```
sudo mkdir -p /var/www/php74
sudo mkdir -p /var/www/php81
```

2. Ajouter un fichier de test

Ajoutez un fichier `index.php` pour tester les configurations PHP :

- Pour PHP 7.4 :

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/php74/index.php
```

- Pour PHP 8.1 :

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/php81/index.php
```

3. Ajuster les permissions

Assurez-vous qu'Apache peut accéder aux fichiers :

```
sudo chown -R www-data:www-data /var/www/php74 /var/www/php81
sudo chmod -R 755 /var/www/php74 /var/www/php81
```

Étape 5 : Tester la configuration

1. Modifier le fichier `hosts`

Ajoutez ces entrées dans `/etc/hosts` pour simuler les noms de domaine :

```
127.0.0.1 php74.example.com
127.0.0.1 php81.example.com
```

2. Accéder aux Virtual Hosts

Ouvrez les URLs dans votre navigateur :

- <http://php74.example.com> : Vous devriez voir **PHP 7.4**.
- <http://php81.example.com> : Vous devriez voir **PHP 8.1**.

Résumé des configurations

1. **PHP-FPM** :
 - Deux versions de PHP configurées avec des sockets distincts.
2. **Apache** :
 - Virtual Hosts distincts, chacun lié à une version de PHP.
3. **Résultat attendu** :
 - Chaque application utilise la version PHP requise, isolée et performante.

Avantages de cette approche

- **Isolation** : Les applications utilisent des versions PHP spécifiques sans conflit.
- **Performance** : PHP-FPM gère efficacement les requêtes grâce à des processus persistants.

- **Extensibilité** : Vous pouvez facilement ajouter de nouvelles applications avec d'autres versions PHP.