



Setup Ubuntu #2

▼ Version	V1
≡ Type	Technique
📅 Date de création	@12 août 2023 10:40
📅 Dernière modification	@14 décembre 2023 15:18



Environnement Node.js

Pour développer des projets Javascript à l'aide de frameworks comme React, Vue, etc il est nécessaire d'avoir un environnement Node.js sur votre machine, pour par exemple exécuter le serveur de développement du framework.



Le Javascript est nativement interprétable par les navigateurs. Si vous souhaitez développer en Javascript pur, sans librairies, installer un environnement Node n'est pas nécessaire.

1. Installer NVM

 [Document officielle installation NVM](#)

NVM permet de gérer différentes versions de Node.js sur votre système et de basculer entre elles facilement. En utilisant NVM, vous pouvez installer différentes versions de Node.js côte à côte sur votre machine. Cela vous permet de choisir la version spécifique de Node.js requise pour chaque projet en fonction de ses

besoins ou prérequis.

Installer NVM sur votre machine.

```
curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash
```

Puis exécuter la commande suivante pour appliquer les changements :

```
source ~/.nvm/nvm.sh
```

Vérifier le bon déroulement de l'installation de NVM.

```
nvm --version
```

Installer la dernière version stable de Node.js.

```
nvm install node
```

Installer la version de Node.js de votre choix, ici nous installerons la version 19 et la version 20

```
nvm install 19  
nvm install 20
```

Dès lors vous pouvez basculer d'une version de Node.js à l'autre, ici vous pourrez basculer de la version 19 à la version 20.

```
nvm use 20
nvm use 19
```



Il n'est pas nécessaire d'installer NPM (Node Package Manager), celui-ci est livré lors de l'installation d'une version de Node.js, sa version est automatiquement adaptée.

Commandes utiles

<code>nvm install <version></code>	Installer une version de Node.js.
<code>nvm use <version></code>	Indiquer la version de Node.js à utiliser.
<code>nvm ls</code>	Lister les versions de Node.js installés.
<code>nvm ls-remote</code>	Lister les versions de Node.js disponibles.



Environnement PHP et Apache

PHP est un langage de programmation principalement utilisé pour le développement web côté serveur, il est souvent associé au serveur web Apache pour interpréter les fichiers PHP et servir des pages web dynamiques aux utilisateurs.



Le PHP n'est pas nativement interprétable par les navigateurs, il est donc nécessaire de configurer Apache de sorte à ce qu'il puisse interpréter les fichiers PHP pour afficher les pages de votre application. Si vous utilisez un framework PHP comme Symfony, la configuration de ce dernier n'est pas nécessaire puisque Symfony apporte son propre serveur de développement.



La configuration de l'environnement PHP / Apache est relativement complexe, prenez votre temps et assurez-vous de suivre le tutoriel étape par étape.

1. Installer PHP et Apache

Installer les paquets PHP et Apache et leurs indispensables.

```
sudo apt install apache2 php libapache2-mod-php php-mysql
```

Installer les paquets nécessaires à l'utilisation de PHP.

```
sudo apt install php-curl php-gd php-intl php-json php-mbstring php-xml php-zip
```

Rendez-vous sur la page web

<http://localhost> de votre navigateur, si l'installation c'est bien déroulé vous devriez arriver sur la page par défaut d'Apache.

2. Configurer Apache

Si Apache n'a pas créé de dossier `public_html` dans le dossier racine de votre utilisateur, créez le.

```
mkdir ~/public_html
```

Indiquer à Apache d'activer le mode "userdir", cela permettra d'exploiter le dossier `public_html` via le navigateur.

```
sudo a2enmod userdir
```

Relancer le service Apache.

```
sudo systemctl restart apache2
```

Créer un fichier `info.php` dans le dossier `public_html`.

```
touch ~/public_html/info.php
```

À l'aide de `nano` ou d'un éditeur de texte, ajouter le code suivant au fichier `info.php`.

```
<?php phpinfo(); ?>
```

Rendez-vous sur la page web http://localhost/~<votre_nom_d_utilisateur>/info.php (remplacer `<votre_nom_d_utilisateur>` et laisser la `~` avant celui-ci). Vous devriez arriver sur une page affichant le code `<?php phpinfo(); ?>` présent dans le fichier `info.php`.

Si vous rencontrez une erreur 403 Forbidden, exécuter les commandes suivantes :

```
cd
cd ../
sudo chmod 711 <votre_nom_d_utilisateur>
```

3. Configurer Apache pour PHP

Par défaut Apache ne permet pas d'interpréter PHP depuis le dossier `public_html`, pour cela nous devons modifier sa configuration.

Rendez-vous dans le dossier `/etc/apache2/mods-enabled`.

```
cd /etc/apache2/mods-enabled
```

Repérer un fichier PHP nommé de la sorte `php<votre_version_de_php>.conf`, dans mon cas le fichier est `php8.1.conf`, ouvrir le fichier à l'aide de `nano` ou d'un éditeur de texte. Si vous avez Sublime Text d'installer sur votre machine exécuter la commande suivante :

```
sudo subl php<votre_version_de_php>.conf
```

Initialement le code présent ressemble à ceci :

```
<FilesMatch ".+\.ph(ar|p|tml)$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch ".+\.phps$">
    SetHandler application/x-httpd-php-source
    # Deny access to raw php sources by default
    # To re-enable it's recommended to enable access to the files
    # only in specific virtual host or directory
    Require all denied
</FilesMatch>
# Deny access to files without filename (e.g. '.php')
<FilesMatch "^\.ph(ar|p|ps|tml)$">
    Require all denied
</FilesMatch>

# Running PHP scripts in user directories is disabled by default
```

```
#
# To re-enable PHP in user directories comment the following lines
# (from <IfModule ...> to </IfModule>.) Do NOT set it to On as it
# prevents .htaccess files from disabling it.
<IfModule mod_userdir.c>
    <Directory /home/*/public_html>
        php_admin_flag engine Off
    </Directory>
</IfModule>
```

Ajouter le code suivant à la fin du fichier.

```
<Directory /home/*/public_html/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Votre fichier doit dorénavant ressembler à ceci :

```
<FilesMatch ".+\.ph(ar|p|tml)$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch ".+\.phps$">
    SetHandler application/x-httpd-php-source
    # Deny access to raw php sources by default
    # To re-enable it's recommended to enable access to the files
    # only in specific virtual host or directory
```

```

    Require all denied
</FilesMatch>
# Deny access to files without filename (e.g. '.php')
<FilesMatch "^\.ph(ar|p|ps|tml)$">
    Require all denied
</FilesMatch>

# Running PHP scripts in user directories is disabled by default
#
# To re-enable PHP in user directories comment the following lines
# (from <IfModule ...> to </IfModule>.) Do NOT set it to On as it
# prevents .htaccess files from disabling it.
# <IfModule mod_userdir.c>
#     <Directory /home/*/public_html>
#         php_admin_flag engine Off
#     </Directory>
# </IfModule>

<Directory /home/*/public_html/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

```

Redémarrer le service Apache.

```
sudo systemctl restart apache2
```

Retourner sur la page web http://localhost/~<votre_nom_d_utilisateur>/info.php (remplacer `<votre_nom_d_utilisateur>` et laisser la `~` avant celui-ci).

Cette fois vous ne devriez plus voir le code `<?php phpinfo(); ?>` du fichier `info.php` mais bien une page en bonne et due forme présentant des informations sur votre version de PHP. Cela veut dire que dorénavant Apache interprète correctement les fichiers PHP présents dans le dossier `public_html`.

3. Configurer PHP

Il reste une dernière étape, actuellement PHP ne donne aucune indication lorsque vous avez une erreur dans votre fichier PHP. Pour vérifier cela, éditer le fichier `public_html/info.php` et supprimer un bout du code pour provoquer une erreur, puis rafraîchissez la page http://localhost/~<votre_nom_d_utilisateur>/info.php.

```
<?php phpinfo(); ?>
```

La page n'affiche plus rien si ce n'est une erreur 500.



La configuration suivante permettra d'avoir un message d'erreur intelligible permettant de facilement debugger le fichier en erreur.



Il est conseillé d'utiliser un éditeur de texte pour manipuler le prochain fichier, en effet il va être nécessaire de faire des recherches au sein de celui-ci et l'utilisation d'un éditeur de texte avec le raccourci `Ctrl + f` facilitera grandement la tâche.

Rendez-vous dans le dossier `/etc/php/<votre_version_php>/apache2`.

```
cd /etc/php/<votre_version_php>/apache2
```

Éditer le fichier `php.ini`, si vous avez Sublime Text d'installer sur votre machine exécuter la commande suivante :

```
sudo subl php.ini
```

À l'aide du raccourci `Ctrl + f` chercher les occurrences du mot `error_reporting`. Vous devriez avoir deux résultats :

Le premier bout de code contenant notre recherche ressemble à cela :

```
; error_reporting
;   Default Value: E_ALL
;   Development Value: E_ALL
;   Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT
```

Il est nécessaire d'enlever les

`;` devant chaque ligne afin de le dé-commenter.

Le second et dernier bout de code contenant notre recherche ressemble à cela :

```
error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT
```

Si c'est le cas il n'est pas nécessaire d'y toucher, autrement veuillez à avoir la même chose.

À l'aide du raccourci `Ctrl + f` chercher les occurrences du mot `display_errors`. Vous devriez avoir trois résultats :

L'unique bout de code contenant notre recherche qui nous intéresse ressemble à cela :

```
display_errors = Off
```

Il est nécessaire de remplacer la valeur `Off` par la valeur `On`.

Redémarrer le service Apache.

```
sudo systemctl restart apache2
```

Retourner sur la page web http://localhost/~<votre_nom_d_utilisateur>/info.php (remplacer `<votre_nom_d_utilisateur>` et laisser la `~` avant celui-ci).

Cette fois vous ne devriez plus voir une page en erreur 500 mais bien une page indiquant l'erreur contenue dans le fichier `info.php` provoquant cette erreur, ce message permet de facilement intervenir dans le fichier pour en résoudre l'erreur. Si vous avez corrompu le fichier `info.php` de la manière indiquée dans ce tutoriel le message ressemble à ceci :

```
Parse error: Unmatched ')' in /home/<nom_utilisateur>/public_html/info.php on line 1
```

Pensez à remettre le fichier `info.php` en état.

Félicitation la configuration de l'environnement PHP et Apache est terminée, voici ce qui a été mis en place :

- ☐ Installation de PHP, Apache et leurs dépendances respectives.
- ☐ Configuration de Apache afin d'interpréter les fichiers PHP du dossier `public_html`.
- ☐ Configuration de PHP pour indiquer les erreurs et faciliter le débogage.

MySQL et PHPMyAdmin

1. Installer MySQL

 [Documentation Ubuntu pour MySQL](#)

MySQL est un système de gestion de base de données relationnelles populaire, largement utilisé pour gérer efficacement les données.



De nombreux logiciels de gestion de bases de données existent, pour les bases de données relationnelles nous pouvons citer : MySQL, PostgreSQL, SQLite. Pour les bases de données non relationnelles : MongoDB, Cassandra ...

Installer MySQL server.

```
sudo apt install mysql-server
```

Vérifier que le service MySQL est en cours d'exécution.

```
sudo systemctl status mysql
```

Dans un soucis de sécurité nous allons créer un utilisateur, il sera préférable d'utiliser cet utilisateur plutôt que d'exploiter MySQL avec votre super utilisateur `sudo`.

Accéder à MySQL, c'est depuis celui-ci que nous exécuterons les commandes suivantes.

```
sudo mysql -u root -p
```

Créer votre utilisateur, pensez à remplacer `<nom_utilisateur>` par le nom que vous souhaitez donner à votre utilisateur et `<un_mot_de_passe>` par le mot de passe que vous souhaitez lui attribuer.

```
CREATE USER 'nom_utilisateur'@'localhost' IDENTIFIED BY 'un_mot_de_passe';
```

Attribuer les privilèges nécessaires à l'exploitation complète de MySQL, pensez à remplacer `<nom_utilisateur>` par le nom que vous attribuer à votre utilisateur.

```
GRANT ALL PRIVILEGES ON *.* TO 'nom_utilisateur'@'localhost' WITH GRANT OPTION;
```

Persister les changements.

```
FLUSH PRIVILEGES;
```

Quitter MySQL.

```
exit;
```

Essayer de vous connecter avec l'utilisateur que vous venez de créer.

```
mysql -u <nom_utilisateur> -p
```

Le mot de passe à indiquer et celui attribuer à votre utilisateur.

Dès lors vous devriez être en capacité d'accéder à MySQL avec cet utilisateur. Si ce n'est pas le cas vous pouvez re-accéder à MySQL en tant que super-administrateur `sudo` et recommencer l'opération.

2. Installer PHPMyAdmin

 [Documentation Ubuntu pour PHPMyAdmin](#)

PHPMyAdmin est une interface web permettant de gérer facilement les bases de données MySQL à l'aide du navigateur. Cela rend la gestion de bases de données MySQL beaucoup plus simple que le CLI du logiciel MySQL.

Installer PHPMyAdmin.

```
sudo apt install phpmyadmin
```



Pour suivre le prompt de PHPMyAdmin il est hautement conseillé de suivre la documentation car celle-ci contient des images.

Après avoir suivi le prompt de PHPMyAdmin, accéder à l'URL <http://localhost/phpmyadmin> et connectez-vous à l'aide de l'utilisateur MySQL précédemment créé, vous devriez accéder à l'interface de gestion de vos bases de données MySQL.



Docker et Docker Compose

 [Documentation officielle installation Docker et Docker Compose](#)

Docker est une plateforme de virtualisation qui permet de conteneuriser les dépendances d'une application et ainsi l'exécuter dans un environnement isolé de celui de votre machine. Docker Compose est un outil complémentaire pour la gestion de plusieurs conteneurs interconnectés au sein d'une application.



Docker et Docker Compose isolent les dépendances d'une application dans des conteneurs autonomes, éliminant ainsi les soucis liés aux versions d'environnement sur la machine hôte.

1. Configurer le répertoire Docker

Installer les paquets pré-requis.

```
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
```

Ajouter la clé GPG officielle de Docker.

```
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

Configurer le répertoire.

```
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ub
```

```
"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

2. Installer Docker et Docker Compose

Installer Docker Engine, containerd et Docker Compose.

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Vérifier que le service Docker est en cours d'exécution.

```
sudo systemctl status docker
```

Si ce n'est pas le cas, démarrer le service Docker.

```
sudo systemctl start docker
```

Pour éviter d'utiliser `sudo` à chaque fois que vous exécutez une commande Docker, vous pouvez ajouter votre utilisateur au groupe Docker.

```
sudo usermod -aG docker $USER  
newgrp docker
```

Puis redémarrer le service Docker.


```
sudo systemctl restart docker
```

Couper votre terminal puis relancer en un pour que les changements soient pris en compte.

Exécuter un container de test (mis à disposition par Docker) pour vérifier que l'installation s'est bien déroulé.

```
docker run hello-world
```