

Procédure d'installation du CMS Magento 2.4.5 sur Ubuntu

Table des matières :

1. [Prérequis](#)
 - a. [Apache2](#)
 - b. [MySQL](#)
 - c. [PHP](#)
 - d. [ElasticSearch](#)
 - e. [Composer](#)
2. [Magento](#)
 - a. [Téléchargement et Installation](#)
 - b. [Configuration](#)
3. [Lancement](#)

1. Prérequis

a. Apache2

Il s'agit de configurer un serveur web pour héberger les sites web de Magento.

- Mettre à jour les paquets d'Ubuntu : `sudo apt update`
- Installer **apache2** : `sudo apt install apache2`
- Se rendre sur l'url `localhost` via le navigateur et vérifier que la page d'Apache s'affiche.
- Permettre aux services d'apache de se lancer automatiquement au démarrage de l'ordinateur :
`systemctl enable apache2.service --now`
- Vérifier que `enabled` est retourné par la commande suivante : `systemctl is-enabled apache2`

b. MySQL

Il s'agit d'installer un système de gestion de base de données (SGBD) et de créer une base de données pour Magento.

Si MySQL est déjà installé, se rendre à la deuxième partie (ii).

i. Installation de MySQL

- Installer MySQL : `sudo apt install mysql-server`
- Lancer MySQL : `sudo mysql`

- Modifier le mot de passe du compte **root** au besoin : `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'exemple_de_mot_de_passe';`
- Quitter le service de MySQL : `exit`

ii. Création d'un nouvel utilisateur

- Se connecter avec le compte **root** et entrer le mot de passe : `mysql -u root -p`
- Créer un nouvel utilisateur et configurer un mot de passe (lettres, chiffres compris): `CREATE USER 'nom_utilisateur_de_magento'@'localhost' IDENTIFIED BY 'mot_de_passe_du_compte';`
- Donner tous les privilèges au nouvel utilisateur : `GRANT ALL PRIVILEGES ON *.* TO 'nom_utilisateur_de_magento'@'localhost' WITH GRANT OPTION;`
- Quitter le service de MySQL : `exit`

iii. Création d'une base de données

- Se connecter avec le compte **nom_utilisateur_de_magento** et entrer le mot de passe : `mysql -u magento2user -p`
- Créer une base de données : `CREATE DATABASE nom_de_la_base_de_donnees;`
- Quitter le service de MySQL : `exit`

c. PHP

Il s'agit d'installer un interpréteur PHP récent, mais compatible (8.1) car Magento 2.4.x ne supporte pas les versions plus récentes que 8.1.

i. Installer PHP 8.1

- Mettre à jour les paquets : `sudo apt update`
- Installer PHP 8.1 et certains paquets (*pour le traitement des requêtes web dynamiques et le module de connexion à mysql*) : `sudo apt install php8.1 libapache2-mod-php php-mysql`
- Vérifier que cette commande retourne une version 8.1.x : `php -v`
 - Si ce n'est pas le cas, configurer la version 8.1 de PHP via la commande :
 - Afficher les versions PHP installées sur le système : `sudo update-alternatives --config php`
 - Choisir la version PHP 8.1 en entrant le numéro correspondant.
- Se rendre dans le fichier : `/etc/apache2/mods-enabled/`
- Ouvrir le fichier `dir.conf` avec : `nano dir.conf` et permuter la position des noms de fichier : `index.html` et `index.php` et sauvegarder le fichier.
- Installer l'extension **mbstring** (*Pour la prise en charge des chaînes de caractères multioctets*) : `sudo apt install php8.1-mbstring`
- Activer l'extension **mbstring** avec : `sudo phpenmod mbstring`
- Activer le module **mod_rewrite** d'apache essentiel pour manipulation des URL de façon plus conviviale : `sudo a2enmod rewrite`

ii. Installer les extensions propre à Magento 2.4.x

- Installer les modules de PHP pour Magento 2.4.x : `sudo apt install php8.1-bcmath php8.1-intl php8.1-soap php8.1-zip php8.1-gd php8.1-curl php8.1-cli php8.1-xml php8.1-xmlrpc php8.1-gmp php8.1-common`
- Relancer apache : `sudo systemctl reload apache2`
- Ouvrir le fichier `/etc/php/8.1/cli/php.ini`
- Rechercher ces lignes et modifier les valeurs associées comme suit, et sauvegarder le fichier :

```
max_execution_time=21000
max_input_time=3600
memory_limit=4G
```

- Relancer apache2 : `sudo systemctl reload apache2`

d. Elasticsearch

Il s'agit de disposer d'un moteur de recherche et d'analyse pour le traitement de grandes quantités de données .

i. Installer Elasticsearch

- Installer openjdk17 (Java) : `sudo apt install openjdk-18-jdk`
- Vérifier que la version de jdk configurée est 17 avec : `java --version`
 - Si ce n'est pas le cas, configurer la version 17 du jdk via la commande :
 - Afficher les versions Java installées sur le système : `sudo update-alternatives --config java`
 - Choisir la version java-17-openjdk en entrant le numéro correspondant.
- Installer un outil de transfert de données via des protocoles réseau : `sudo apt install curl` (*cURL : ClientURL*)
- Importer la clé GPG (GNU Private Guard) pour le package Elasticsearch : `sudo curl -sSfL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --no-default-keyring --keyring=gnupg-ring:/etc/apt/trusted.gpg.d/magento.gpg --import`
- Ajouter la référence du package Elasticsearch : `sudo sh -c 'echo "deb https://artifacts.elastic.co/packages/7.x/apt/stable/main" > /etc/apt/sources.list.d/elastic-7.x.list'`
- Donner les droits de lecture et d'écriture à tout le monde sur le fichier de la clé de Magento : `sudo chmod 666 /etc/apt/trusted.gpg.d/magento.gpg`
- Mettre à jour les paquets : `sudo apt update`
- Installer le paquet Elasticsearch : `sudo apt install elasticsearch`
- Recharger les fichiers de configuration avec : `sudo systemctl daemon-reload`
- Activer l'auto-démarrage du service elasticsearch : `sudo systemctl enable elasticsearch.service`
- Démarrer le précédent service : `sudo systemctl start elasticsearch.service`

ii. Configurer Elasticsearch

Ouvrir chacun des fichiers suivants et faire les configurations demandées et les sauvegarder :

- **/etc/elasticsearch/elasticsearch.yml :**
 - Décommenter la ligne : *cluster.name*
 - Décommenter la ligne : *node.name*
 - Décommenter la ligne : *network.host* et remplacer l'adresse présente par **127.0.0.1**
 - Décommenter la ligne : *http.port*
- **/etc/elasticsearch/jvm.options :**
 - Remplacer la ligne **## -Xms4g** par **# -Xms256m**
 - Remplacer la ligne **## -Xmx4g** par **# -Xmx256m**
 - Entrer l'url localhost:9200 dans le navigateur et vérifier qu'un texte semblable s'affiche :

```
{
  "name" : "node-1",
  "cluster_name" : "my-application",
  "cluster_uuid" : "F0k4q3_7Q66kDqJtNJUTKg",
  "version" : {
    "number" : "7.17.25",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "f9b6b57d1d0f76e2d14291c04fb50abeb642cfbf",
    "build_date" : "2024-10-16T22:06:36.904732810Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.3",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

e. Composer

Il s'agit d'installer un gestionnaire de dépendances pour PHP. Il est utilisé pour gérer les bibliothèques et les paquets nécessaires à une application PHP en l'occurrence Magento.

- Se rendre dans le dossier racine **root** : **cd ~**
- Télécharger et installer **Composer** : **curl -sS https://getcomposer.org/installer -o composer-setup.php**
- **sudo php composer-setup.php --install-dir=/usr/bin --filename=composer**
- Vérifier que la commande **composer** est reconnue.

2. Magento

a. Téléchargement et Installation

- Se rendre dans le dossier qui contient les fichiers des sites servis par apache : **cd /var/www/html**
- Créer un projet Magento : **sudo composer create-project --repository-url=https://repo.magento.com/ magento/project-community-edition=2.4.5**

magento2

La commande précédente débute le processus d'installation de Composer. Puis, des identifiants sont demandés (*username* et *password*). Il faut donc suivre les étapes suivantes pour continuer :

- Se rendre sur le site d'Adobe via l'url repo.magento.com
- Créer un compte si ce n'est pas encore fait
- Se rendre sur **Mon Profile > Clés d'accès**
- Créer une nouvelle clé d'accès
- Copier la clé publique au niveau de username
- Copier la clé privée au niveau de password. (*Le mot de passe ne s'affichera pas*)
- Valider avec **ENTREE**

Après l'installation, donner les droits d'accès à certains fichiers de Magento :

- Se rendre le dossier `/var/www/html/` avec : `cd /var/www/html/<dossier_de_magento_cree>`
(Le nom de ce dossier peut être connu avec `ls` dans le dossier `/var/www/html/`. Il débute par **magento**)
- `sudo find var generated vendor pub/static pub/media app/etc -type f -exec chmod g+w {} +`
- `sudo find var generated vendor pub/static pub/media app/etc -type d -exec chmod g+ws {} +`
- `sudo chown -R user_name:www-data .`
- `sudo chmod u+x bin/magento`

b. Configuration

- Se rendre dans le dossier Magento : `cd /var/www/html/magento2`
- Exécuter la ligne suivante pour configurer les variables internes de Magento :

```
php bin/magento setup:install --base-url=http://rocktube.magento.com --db-host=localhost --db-name=magento2db --db-user=magento2user --db-password=Admin12345 --admin-firstname=Admin --admin-lastname=Admin --admin-email=admin@gmail.com --admin-user=admin --admin-password=Admin123 --language=en_US --currency=USD --timezone=America/Chicago --backend-frontname=admin --search-engine=elasticsearch7 --elasticsearch-host=localhost --elasticsearch-port=9200
```

Note : Les variables suivantes doivent être configurées.

- **--base-url** : URL du site
- **--db-name** : Nom de la base de données configurée dans MySQL
- **--db-user** : Nom de l'utilisateur de Magento configuré dans MySQL
- **--db-password** : Mot de passe de l'utilisateur de Magento configuré dans MySQL
- **--admin-user** : Nom de l'utilisateur ADMIN de Magento (au choix)
- **--admin-password** : Mot de passe de l'utilisateur ADMIN de Magento (au choix)

Créer un hôte virtuel pour la configuration multi-sites web sur Apache :

- Se rendre dans le dossier des sites d'apache : `cd /etc/apache2/sites-available`

- Créer un fichier de configuration pour le site : (Exemple : *monsite.magento.com.conf*) via **nano** **monsite.magento.com.conf**.
- Ouvrir ce fichier et y copier ce qui suit en se rassurant que **monsite.magento.com** correspond au nom du site :

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/magento2/pub
ServerName monsite.magento.com
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
<Directory "/var/www/html">
AllowOverride all
</Directory>
</VirtualHost>
```

- Ouvrir le fichier des hôtes : **nano /etc/hosts**
- Ajouter la ligne suivante à la fin du fichier et le sauvegarder : **127.0.0.1 /etc/apache2/sites-available**
- Relancer apache : **systemctl restart apache2**
- Activer le fichier de configuration d'hôte virtuel dans apache pour rendre le site configuré accessible : **sudo a2ensite /etc/apache2/sites-available**
- Met à jour la configuration de Magento et désactive l'authentification à deux facteurs de Magento : **php bin/magento indexer:reindex && php bin/magento se:up && php bin/magento se:s:d -f && php bin/magento c:f && php bin/magento module:disable Magento_TwoFactorAuth**

3. Lancement

- Entrer l'url **monsite.magento.com** dans le navigateur et vérifier qu'une page HOME s'affiche
- Entrer l'url **monsite.magento.com/admin** dans le navigateur et vérifier que l'interface d'administration s'affiche. Il suffit alors de se connecter avec ses identifiés d'administrateur (cf. **2 > b**)