B. Mise en place

- Téléchargement des sources
- Installation du serveur web
- Installation de MySQL ou MariaDb
- Installation de PHP
- Configuration de PHP
- Installation de Vips
- Création des dossiers nécessaires

Téléchargement des sources

Le projet Archimage est publié sous la licence AGPL 3.0 et est hébergé dans la forge logicielle GitHub.

Vous pouvez télécharger la dernière version de l'applicationen cliquant sur ce lien

Vous pouvez également consulter le code source sur la page du projet :

https://github.com/EFAthenes/archimage

Facultatif : installer unzip

```
sudo apt install unzip
```

Téléchargeons et plaçons le dossier de l'application dans le répertoire /var/www/ avec les droits par défaut.

```
wget https://github.com/EFAthenes/archimage/releases/download/26.5/archimage_26.5.zip

sudo mkdir -p /var/www/html

sudo unzip archimage_26.5.zip -d /var/www/html/

sudo chown www-data:www-data /var/www/html/archimage_26.5/ -Rf
```

Installation du serveur web

Vous pouvez installer le serveur de votre choix néanmoins seul Apache2 et NginX ont été correctement testé.

Préambule : Télécharger le zip contenant les sources de l'application.

Apache 2

```
sudo apt install apache2
```

Nous vous invitons également à installer les modules suivants

```
sudo a2enmod ssl

sudo a2enmod rewrite

sudo a2enmod headers
```

Configurer le host de apache en fonction du dossier d'installation :

*Faites pointer le dossier sur le répertoire **/var/www/archimage_26.5/Archimage2/www/**

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Exemple de fichier de conf Apache2 :

```
<VirtualHost *:443>
	
	ServerName site-archimage.com
	ServerAdmin admin@archimage.com

	ErrorLog ${APACHE_LOG_DIR}/archimage-error.log
	CustomLog ${APACHE_LOG_DIR}/archimage-acces.log combined
```

```
    # ALOWW CORS
    <IfModule mod_headers.c>
  Header always unset Access-Control-Allow-Origin
Header add Access-Control-Allow-Origin "*"
    </IfModule>

DocumentRoot /var/www/html/archimage_26.5/Archimage2/www/
<Directory /var/www/html/archimage_26.5/Archimage2/www/>
        AllowOverride All
        require all granted
  LimitRequestBody 40048576000
</Directory>

    RewriteEngine on

    ### /iiif/id_document/info.json
    #RewriteRule ^/iiif/([^/]+)/info\.json$ /action.php?kroute=iiif_json_public&id=$1
[L]

    ### /iiif/id_document/full/800,/0/default.jpg
    #RewriteRule ^/iiif/([^/]+)/([^/]+)/([^/]+)/([^/]+)/([^/.]+)\.([^/]+)$
/index.php?kroute=image_request_iiif&id=$1&region=$2&size=$3&rotation=$4&quality=$5&for
mat=$6 [L]

    #### /iiif/id_document
    #RewriteRule ^/iiif/([0-9]+)$ /action.php?kroute=iiif_json_public&id=$1 [L]



ErrorDocument 404 /index.php

# Possible values include: debug, info, notice, warn, error, crit,
LogLevel warn

</VirtualHost>
```

Non recommandé : Utiliser seulement Apache2 et donc rajouter le module Apache PHP

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

puis le recharger :

```
sudo systemctl reload apache2
```

Recommandé rajouter PHP-FPM

```
sudo apt install php8.5-fpm
sudo a2enmod proxy_fcgi setenvif
sudo a2enconf php8.5-fpm
```

*si vous avez déjà activé Apache2 :

```
sudo a2dismod php8.5
```

Puis relancer les services :

```
sudo systemctl restart php8.5-fpm
sudo systemctl restart apache2
```

NginX

```
sudo apt install nginx -y


sudo apt install php8.5-fpm
```

Modifier les fichiers par défaut :

```
sudo nano /etc/php/8.5/fpm/pool.d/www.conf
```

```
sudo nano /etc/nginx/sites-available/default
```

Un fichier d'exemple de cpnfiguration Nginx :

```
server {
    listen 443 ssl;
    server_name site-archimage.com;

    index index.php;



    # Haerdenning
    server_tokens off;
    etag off;

    root /var/www/html/archimage_26.5/Archimage2/www;

 location / {
    try_files $uri $uri/ =404;
  }

  # pass PHP scripts on Nginx to FastCGI (PHP-FPM) server
  location ~ \.php$ {
    include snippets/fastcgi-php.conf;

    # Nginx php-fpm sock config:
    fastcgi_pass unix:/run/php/php8.5-fpm.sock;
    # Nginx php-cgi config :
    # Nginx PHP fastcgi_pass 127.0.0.1:9000;
  }



# IIIF REWRITES
#rewrite    ^/iiif/(.*)/info.json$
"https://archimage.com/action.php?kroute=iiif_json_public&id=$1";
#rewrite ^/iiif/(.*)/(.*)/(.*)/(.*)/(.*)\.(.*)?$
https://archimage.com/index.php?kroute=image_request_iiif&id=$1&region=$2&size=$3&rotat
```

```
ion=$4&quality=$5&format=$6;
#rewrite    ^/iiif/([0-9]+)$
"https://archimage.com/action.php?kroute=iiif_json_public&id=$1";




  # deny access to Apache .htaccess on Nginx with PHP,
  # if Apache and Nginx document roots concur
  location ~ /\.ht {
    deny all;
  }

#    return       404;
}
```

puis le recharger :

```
sudo nginx -t
sudo systemctl reload nginx
```

Installation de MySQL ou MariaDb

La plateforme Archimage fonctionne indifférement avec MySQL et MariaDB, il suffira de le préciser dans le fichier de configuration.

Pour installer MySQL par ici, pour MariaDB par là

MYSQL

Installation

En ligne de commande :

```
sudo apt update && sudo apt install mysql-server
```

S'assurer que le service s'est correctement installé :

```
sudo service mysql status
```

Nous recommandons d'installer la version sécurisée via

```
sudo mysql_secure_installation
```

Création d'un utilisateur

Se connecter en ligne de commande à MySQL

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

Puis créer un utilisateur, une base de données et lui affecter les droits :

```
CREATE USER 'myuser'@'localhost' IDENTIFIED BY 'StrongPassword123!';

CREATE DATABASE mydatabase CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

GRANT ALL PRIVILEGES ON mydatabase.* TO 'myuser'@'localhost';

FLUSH PRIVILEGES;
```

Veuillez bien noter ses informations ils seront demandés par l'application dans le fichier de configuration.

Puis quittez

```
exit;
```

Optionnel :

Si vous comptez avoir des centaines de milliers d'enregistrements il est conseiller de rajouter de modifier la variable **sort_buffer_size**

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

et rajouter

```
[mysqld]
sort_buffer_size=8192k
```

Et relancer le service

```
sudo service mysql restart
```

MariaDb

Installation

En ligne de commande :

```
sudo apt update && sudo apt install mariadb-server mariadb-client -y
```

S'assurer que le service s'est correctement installé :

```
sudo service mariadb status
```

Nous recommandons d'installer la version sécurisée via

```
sudo mysql_secure_installation
```

Pour avoir la version :

```
mariadb --version
```

Création d'un utilisateur

Se connecter en ligne de commande à MySQL

```
sudo mariadb -p -u root
```

Puis créer un utilisateur, une base de données et lui affecter les droits (veuillez remplacer les données génériques de l'utilisateur le mot de passe) :

```
CREATE USER 'myuser'@'localhost' IDENTIFIED BY 'StrongPassword123!';

CREATE DATABASE mydatabase CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci;

GRANT ALL PRIVILEGES ON mydatabase.* TO 'myuser'@'localhost';

FLUSH PRIVILEGES;
```

Veuillez bien noter ses informations ils seront demandés par l'application dans le fichier de configuration.

Puis quittez

```
exit;
```

Optionnel :

Si vous comptez avoir des centaines de milliers d'enregistrements il est conseiller de rajouter de modifier la variable **sort_buffer_size**

```
sudo nano /etc/mysql/mariadb.conf.d/99-custom.cnf
```

et rajouter

```
[mysqld]
sort_buffer_size=8192k
```

Et relancer le service

```
sudo service mariadb restart
```

Installation de PHP

Toujours en ligne de commande veuillez installer la version de PHP et les extensions nécessaires.

Facultatif : Vous pouvez ajouter le repository pour obtenir des versions alternatives de PHP pour **Ubuntu version 24**

```
sudo add-apt-repository ppa:ondrej/php


sudo apt update && sudo apt upgrade -Y
```

PHP 8.4 :

```
sudo apt install php8.4 php8.4-bcmath php8.4-bz2 php8.4-cli php8.4-common php8.4-gd php8.4-igbinary php8.4-imagick  php8.4-intl php8.4-mbstring php8.4-mysql php8.4-opcache php8.4-readline php8.4-xml php8.4-zip php8.4-curl
```

PHP 8.5 : recommandé

```
sudo apt install php8.5 php8.5-bcmath php8.5-bz2 php8.5-cli php8.5-common php8.5-gd php8.5-igbinary php8.5-imagick php8.5-intl php8.5-mbstring php8.5-mysql php8.5-readline php8.5-xml php8.5-zip php8.5-curl
```

Configuration de PHP

Nous allons paramétrer PHP averc des valeurs acceptables pour que l'application foncitonne normalement. À vous de les adapter en fonction des besoins et des possibilités du matériel.

Nous prenons pour exemple la version de PHP 8.5.

Apache :

```
sudo nano /etc/php/8.5/apache2/php.ini
```

NginX ou Apache avec FPM :

```
sudo nano /etc/php/8.5/fpm/php.ini
```

Veuillez trouver les paramètres suivants :

```
post_max_size=200M

upload_max_filesize=200M

max_file_uploads=200

memory_limit=2048M

max_input_vars=1000

date.timezone = Europe/Paris
```

Si Vips installé :

```
ffi.enable=true
zend.max_allowed_stack_size=-1
pcre.jit=0
```

Puis relancer les services :

```
sudo systemctl restart php8.5-fpm
sudo systemctl restart apache2

sudo systemctl restart php8.5-fpm
sudo systemctl restart nginx
```

Installation de Vips

Pour les instances gérant des fichiers images de plusieurs dizaines de milliers de pixels il est recommandé de ne pas utiliser les libriairies standard de traitement d'images de PHP : GD ou Imagick)

Archimage fonctionne correctement avec ces librairies mais les performances seront médiocres pour tout ce qui est gestion du IIIF.

Afin d'améliorer les performances nous vous proposons d'utiliser VIPS, cette libraiirie s'installe de nos jours relativement simplement sous Ubuntu :

```
sudo apt-get install --no-install-recommends libvips42
```

Si vous voulez vous pouvez également installer les outils afin d'avoir plus d'information ou de modifier le paramétrage de cette librairie.

```
sudo apt install libvips-tools
```

Vérifier qu'elle est bien installée :

```
vips -v
```

Rajouter dans le php.ini

```
ffi.enable=true
zend.max_allowed_stack_size=-1
pcre.jit=0
```

Création des dossiers nécessaires

Archimage nécessite d'utiliser certains dossiers pour un bon fonctionnement.

- Répertoire de chargement de fichier temporaire
- Répertoire de stockage de log
- Répertoire de stockage de mails envoyés
- Répertoire de stockage des fichiers numériques que va gérer Archimage
- Optionnel : répertoire de stockage de visuel généré par l'API IIIF

Tous ses dossiers nécéssitent d'avoir des droits similaires au server web.

Vous pouvez individuellement choisir leur emplacement dans le fichier de config mais nous allons vous proposer de créer directement cela :

```
sudo mkdir /var/www/archimage_dir/
sudo mkdir /var/www/archimage_dir/tmp_upload
sudo mkdir /var/www/archimage_dir/logs
sudo mkdir /var/www/archimage_dir/mails
sudo mkdir /var/www/archimage_dir/repository
sudo mkdir /var/www/archimage_dir/iiif_cache/

sudo chown www-data:www-data /var/www/archimage_dir/ -Rf
```