

# TP MariaDB X Apache2

## Objectif : Déploiement **Maria DB**

J'ai configuré un environnement web complet permettant de collecter, stocker et afficher des données utilisateur à partir d'un formulaire en ligne, depuis un site web que j'avais créé sur Apache2. Ce rapport détaille les étapes suivies pour installer et configurer les serveurs nécessaires, ainsi que les scripts PHP utilisés pour interagir avec une base de données MariaDB. Voici un aperçu des étapes réalisées :

J'ai commencé par préparer les serveurs nécessaires. Sur le serveur de base de données, j'ai installé MariaDB et modifié le fichier de configuration pour permettre les connexions distantes. Ensuite, j'ai créé un utilisateur avec les permissions appropriées et une base de données nommée batdata. Dans cette base de données, j'ai créé une table users pour stocker les informations des utilisateurs.

Parallèlement, sur le serveur web, j'ai installé Apache et PHP, puis redémarré le service Apache pour appliquer les modifications. Une fois les serveurs configurés, j'ai déployé les fichiers nécessaires sur le serveur web. J'ai créé un fichier HTML, formation.html (le nom que je voulais donner), contenant un formulaire pour collecter les noms et adresses email des utilisateurs. Ensuite, j'ai développé un script PHP, submit\_data.php, pour traiter les données soumises et les insérer dans la base de données MariaDB.

Pour afficher les données collectées, j'ai créé un autre script PHP, display\_data.php, qui récupère les informations de la table users et les affiche dans un tableau HTML. J'ai ensuite testé la procédure en accédant au formulaire via un navigateur web, en soumettant des données et en vérifiant leur affichage sur la page dédiée.

Cette procédure a permis de mettre en place une solution complète de gestion de données utilisateur, de la collecte à l'affichage, en passant par le stockage sécurisé dans une base de données distante.

# Installation de Maria DB

Premièrement nous allons faire les mises à jour du système et ensuite

Installer **Maria DB**: `sudo apt install mariadb mariadb-server`

## Étape 1 : Installer MariaDB sur la deuxième VM

1. Mettre à jour le système :

```
sudo apt update  
sudo apt upgrade
```

2. Installer MariaDB :

```
sudo apt install mariadb-server
```

Ensuite nous allons activer (mettre en service)

3. Démarrer et activer MariaDB :

```
sudo systemctl start mariadb  
sudo systemctl enable mariadb
```

4. Sécuriser l'installation de MariaDB :

```
sudo mysql_secure_installation
```

**Place à la configuration :**

Se rendre dans le répertoire : `Mariadb.conf`

Ajouter la commande **`bind-address = 0.0.0.0`**

## Étape 2 : Configurer MariaDB pour accepter les connexions à distance

1. Éditer le fichier de configuration MariaDB :

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Trouvez la ligne `bind-address` et changez-la pour :

```
bind-address = 0.0.0.0
```



Ensuite restart la machine

```
sudo systemctl restart mariadb
```

# Création USER & DATABASE

Se connecter à l'interface **Maria DB** :

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

Ensuite reprendre le code ci-dessous et le mettre dans maria db pour crée la base et les utilisateurs (*pensez à remplacer les « users » et « ip »*)

```
CREATE DATABASE batdata;  
CREATE USER 'votre_utilisateur'@'adresse_ip_du_serveur_web' IDENTIFIED BY  
'votre_mot_de_passe';  
GRANT ALL PRIVILEGES ON batdata.* TO  
'votre_utilisateur'@'adresse_ip_du_serveur_web';  
FLUSH PRIVILEGES;  
EXIT;
```

Se reconnecter : (en **root** de préférence)

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

Et dans **Maria DB** ajouter cette commande pour crée un user dans la base

```
USE batdata;  
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL  
);  
EXIT;
```

## Configuration du serveur WEB

### ***Etape 1 :***

Reprendre notre serveur web Apache2, et venir installer ces modules de librairie PHP

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

Ensuite restart le serveur apache2 : **sudo systemctl restart apache2**

### ***Etape 2 :***

Déploiement des fichiers sur le serveur WEB

Ajouter au code HTML un formulaire qui sera ensuite envoyé à la base de données du deuxième serveur 'Data Base' dans site2 :

dans le répertoire `/var/www/site2` :

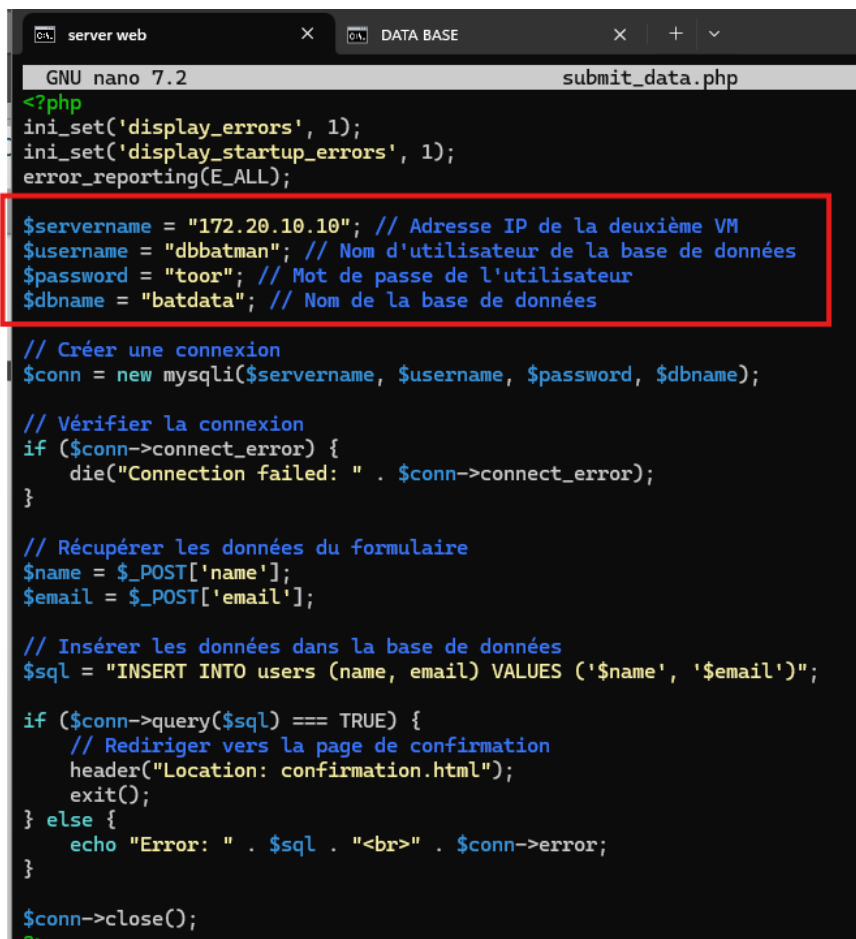
## Visualisation du code :

Quand je fais ls, j'ai l'intégralité des mes fichiers indispensable :

- **Formation.html** (code HTML principale du site)
  - o Dans le code **HTML** ajouter le **FORMULAIRE**
  - o Ajoute d'un « **confirmation.html** » pour redirection formulaire
- **Submit\_data.php** (code de mise en relation avec le serveur data base)
  - o Traitement des données du **FORMULAIRE** HTML
- **Display\_data.php** (code d'affichage 'tableau' des données envoyer)
  - o Ajout du tableau pour visualisation des données envoyer à la DB

```
abstract.jpg confirmation.html displa_data.php formation.html submit_data.php
root@batman:/var/www/site2#
```

Le fichier **submit\_data.php** devrait ressembler à ça :



```
server web x DATA BASE x + v
GNU nano 7.2 submit_data.php
<?php
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

$servername = "172.20.10.10"; // Adresse IP de la deuxième VM
$username = "dbbatman"; // Nom d'utilisateur de la base de données
$password = "toor"; // Mot de passe de l'utilisateur
$dbname = "batdata"; // Nom de la base de données

// Créer une connexion
$conn = new mysqli($servername, $username, $password, $dbname);

// Vérifier la connexion
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Récupérer les données du formulaire
$name = $_POST['name'];
$email = $_POST['email'];

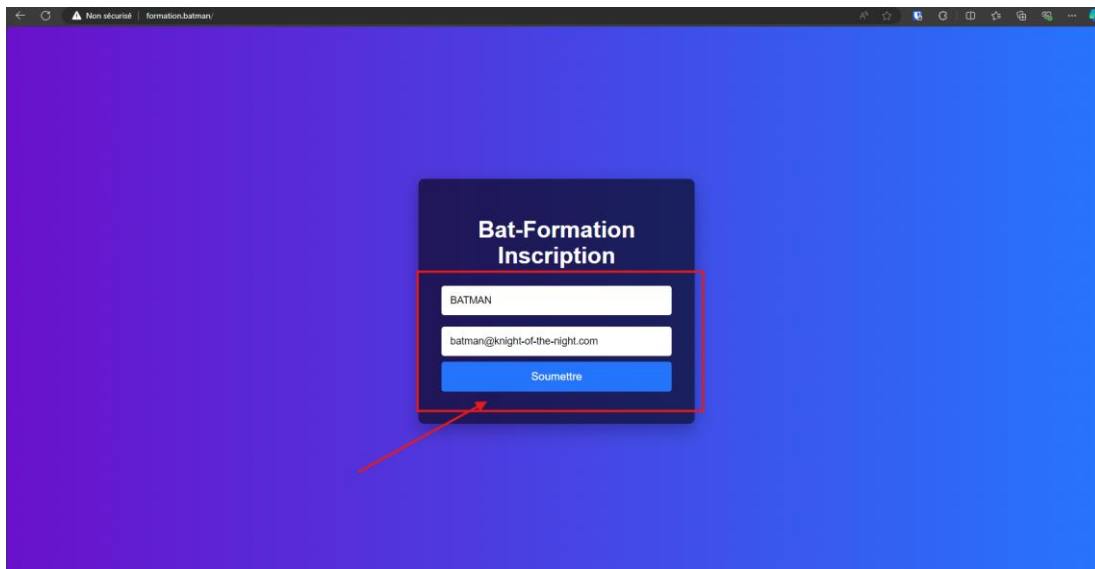
// Insérer les données dans la base de données
$sql = "INSERT INTO users (name, email) VALUES ('$name', '$email')";

if ($conn->query($sql) === TRUE) {
    // Rediriger vers la page de confirmation
    header("Location: confirmation.html");
    exit();
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

Il est important de bien remplir les informations en encadrer rouge, elles sont cruciales pour la établir la connexion avec la 2éme VM

## TEST du site + Envoie données DATA\_BASE



Une fois les données rentrer dans le formulaire, cliquer sur soumettre et elle devrait être envoyer dans le formulaire [display\\_data.php](#)

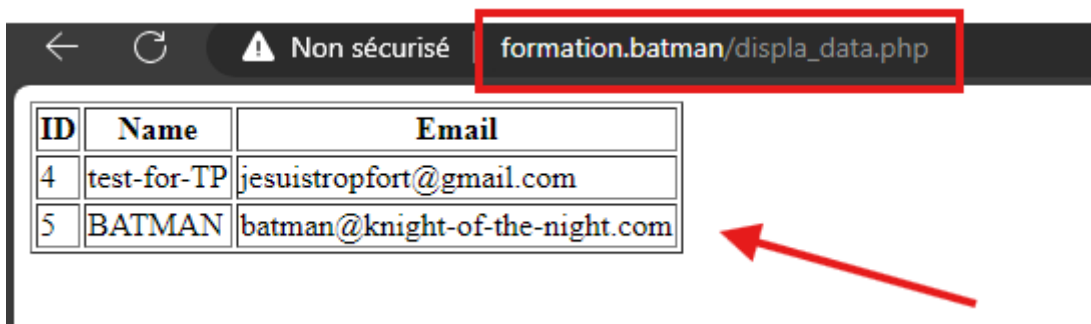
Après l'envoi du formulaire, le serveur DATA\_BASE reçoit bien les informations et les affiche sur le [display\\_data.php](#)

**Avant :** (sans le [display\\_data.php](#)) tester si on reçoit bien les infos dans la database

```
MariaDB [batdata]> SELECT * FROM users;
```

id	name	email
4	test-for-TP	jesuistropfort@gmail.com
5	BATMAN	batman@knight-of-the-night.com

**Après :** Ajout du [display\\_data.php](#), les infos sont bien transmises au site sous tableau



ID	Name	Email
4	test-for-TP	jesuistropfort@gmail.com
5	BATMAN	batman@knight-of-the-night.com

# CONCLUSION

En conclusion, j'ai pu mettre en place une solution de déploiement de Maria DB en relation avec Apache2 qui est efficace pour la gestion des données utilisateur, depuis leur collecte via un formulaire en ligne jusqu'à leur stockage sécurisé dans une base de données distante et leur affichage dynamique sur une page web. Grâce à l'installation et à la configuration de MariaDB et Apache, ainsi qu'au développement de scripts PHP adaptés, j'ai pu créer un environnement web fonctionnel et interactif. Ce travail renforcé notre compréhension des technologies web (principalement celle de l'IA). Les compétences acquises et les défis relevés au cours de ce projet seront sans aucun doute précieux pour les futurs TP.