Liaison d'un site PHP avec une base de données distante

Introduction

Cette procédure vise à lier un site web développé en PHP avec une base de données distante hébergée sur un autre serveur. Elle s'appuie sur un exemple simple de formulaire de prise de rendez-vous.

Prérequis

- Un serveur web avec Apache2 et PHP installé (ex : 192.168.1.10)
- Un serveur de base de données MariaDB ou MySQL (ex : 192.168.1.20)
- Un poste client pour accéder au site via navigateur web

Partie 1: Configuration du serveur web

1. Installation de PHP et des modules MySQL

```
apt update
apt install php libapache2-mod-php php-mysql -y
```

2. Création de la page index.php

Créer le fichier contenant le formulaire HTML et le code PHP pour l'enregistrement dans la base :

```
nano /var/www/html/index.php
```

Puis coller le contenu suivant :

```
<!DOCTYPE html>
<html>
<head>
  <title>Prise de rendez-vous</title>
```

```
</head>
<body>
  <h2>Prendre un rendez-vous</h2>
  <form action="index.php" method="post">
    Nom: <input type="text" name="nom"><br><br>
    Date: <input type="date" name="date"><br><br>
    Heure: <input type="time" name="heure"><br><br>
    <input type="submit" value="Réserver">
  </form>
  <?php
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nom = $_POST['nom'];
    $date = $_POST['date'];
    $heure = $_POST['heure'];
    $conn = new mysqli("192.168.1.20", "web_user", "MotDePasseSecurise
1@", "rdv_db");
    if ($conn → connect_error) {
      die("Connexion échouée : " . $conn → connect_error);
    }
    $sql = "INSERT INTO rdv (nom, date, heure) VALUES ('$nom', '$date',
'$heure')";
    if (\$conn \rightarrow query(\$sql) === TRUE) \{
      echo "Rendez-vous enregistré !";
    } else {
      echo "Erreur : " . $conn → error . "";
    }
    $conn → close();
  }
  ?>
</body>
</html>
```

Paramètres à adapter dans le code PHP :

- 192.168.1.20 : adresse IP du serveur de base de données
- web_user : nom d'utilisateur SQL autorisé à écrire dans la base
- MotDePasseSecurise1@: mot de passe fort pour l'utilisateur
- rdv_db : nom de la base de données contenant la table rdv

Ces paramètres doivent correspondre exactement à ceux créés lors de la configuration du serveur BDD.

Ensuite, redémarrer Apache pour prendre en compte le nouveau fichier PHP:

systemctl restart apache2

Prendre un rendez-vous

Nom:
Date: jj/mm/aaaa 🗂
Heure ::
Réserver

Simple page de réservation

Partie 2 : Configuration du serveur base de données

1. Installation de MariaDB

apt update && apt install mariadb-server -y

2. Autoriser les connexions distantes

Par défaut, MariaDB n'accepte que les connexions locales (depuis 127.0.0.1). Il faut modifier ce comportement pour permettre les connexions distantes (comme depuis le serveur web).

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

Remplacer:

```
bind-address = 127.0.0.1
```

par:

bind-address = 0.0.0.0

★ Pourquoi 0.0.0.0?

Cela permet à MariaDB d'écouter sur toutes les interfaces réseau, donc d'accepter les connexions depuis d'autres machines du réseau (comme le serveur web). C'est indispensable pour une architecture distribuée.

Redémarrer MariaDB:

systemctl restart mariadb

3. Création de la base, de l'utilisateur et de la table

Connexion à MariaDB:

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

Créer une base de données, un utilisateur avec un mot de passe sécurisé et la table des rendez-vous :

```
CREATE DATABASE rdv_db;
CREATE USER 'web_user'@'%' IDENTIFIED BY 'MotDePasseSecurise1@';
GRANT ALL PRIVILEGES ON rdv_db.* TO 'web_user'@'%';
FLUSH PRIVILEGES;

USE rdv_db;
CREATE TABLE rdv (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nom VARCHAR(50) NOT NULL,
```

```
date DATE NOT NULL,
heure TIME NOT NULL
);
```

Explications :

- rdv_db est la base de données que le site web va utiliser
- web_user est l'utilisateur qui se connectera depuis le serveur web
- "" permet l'accès depuis n'importe quelle IP (à restreindre si besoin)
- rdv est la table qui stocke les RDV (nom, date, heure)

Partie 3: Test de la liaison Web <-> BDD

Depuis le serveur web (ou tout autre hôte autorisé) :

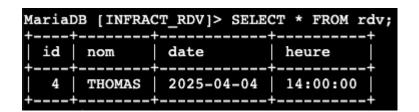
```
mysql -h 192.168.1.20 -u web_user -p
```

Mot de passe: MotDePasseSecurise1@

Vérifiez que vous pouvez accéder à la base et que des données sont enregistrées :

```
USE rdv_db;
SELECT * FROM rdv;
```

Vous devriez voir les rendez-vous insérés via le formulaire PHP.



Conclusion

Vous avez maintenant un site PHP capable de se connecter à une base de données distante, dans une architecture professionnelle séparant services web

et données. Cela facilite la maintenance, améliore la sécurité et offre une meilleure évolutivité à l'infrastructure.