

Documentation Easy Bee 3

Sommaire

Table des matières

Mariadb.....	2
Sauvegarde de la base de données.....	3
Mise en Place d’un GLPI.....	4
Mise en place du serveur de mail.....	5
Installation de Postfix.....	5
Fichier de configuration.....	5

Maël Poisson

Mariadb

Pour commencer on met à jours les packages

```
root@debian:/home/adminpoisson# sudo apt update
```

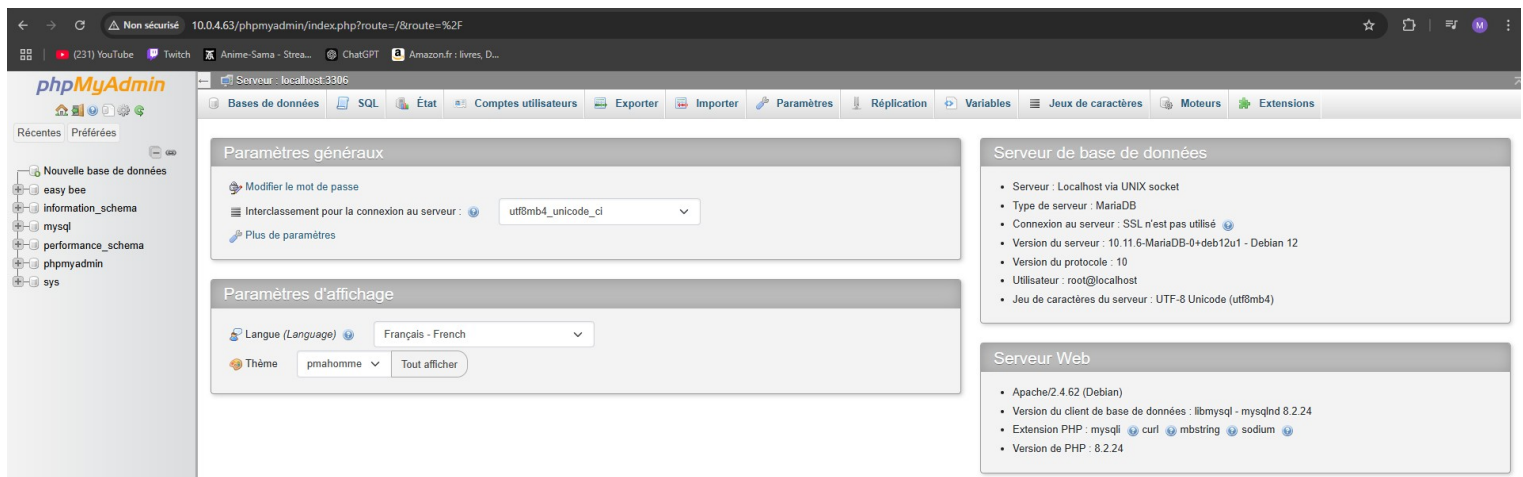
```
root@debian:/home/adminpoisson# sudo apt upgrade
```

puis on vérifie si il est démarré et actif

```
root@debian:/home/adminpoisson# systemctl status mariadb
● mariadb.service - MariaDB 10.11.6 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-11-20 14:28:27 CET; 57min ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 635 (mariabdb)
    Status: "Taking your SQL requests now..."
     Tasks: 8 (Limit: 2284)
    Memory: 95.5M
       CPU: 448ms
    CGroup: /system.slice/mariadb.service
            └─635 /usr/sbin/mariabdb

nov. 20 14:28:26 debian mariabdb[635]: 2024-11-20 14:28:26 0 [Note] Server socket created on IP: '127.0.0.1'.
nov. 20 14:28:26 debian mariabdb[635]: 2024-11-20 14:28:26 0 [Note] InnoDB: Buffer pool(s) load completed at 241120 14:28:26
nov. 20 14:28:27 debian mariabdb[635]: 2024-11-20 14:28:27 0 [Note] /usr/sbin/mariabdb: ready for connections.
nov. 20 14:28:27 debian mariabdb[635]: Version: '10.11.6-MariaDB-0+deb12u1' socket: '/run/mysqld/mysqld.sock' port: 3306 Debian 12
nov. 20 14:28:27 debian systemd[1]: Started mariadb.service - MariaDB 10.11.6 database server.
nov. 20 14:28:27 debian /etc/mysql/debian-start[758]: Upgrading MySQL tables if necessary.
nov. 20 14:28:27 debian mariabdb[635]: 2024-11-20 14:28:27 3 [Warning] Access denied for user 'root'@'localhost' (using password: NO)
nov. 20 14:28:27 debian /etc/mysql/debian-start[767]: Checking for insecure root accounts.
nov. 20 14:28:27 debian mariabdb[635]: 2024-11-20 14:28:27 4 [Warning] Access denied for user 'root'@'localhost' (using password: NO)
nov. 20 14:28:27 debian debian-start[770]: ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
root@debian:/home/adminpoisson#
```

puis on se connecte dessus pour être sur l'interface graphique (<http://ip/phpmyadmin>)



Ensuite il faut importer la base de données

Maël Poisson

Sauvegarde de la base de données

Créer un fichier script

```
root@debian:/sauvegarde# nano sauvegarde_complete.sh
```

Voici ce que le script doit contenir pour qu'il sauvegarde au niveaux du serveur de fichier, qu'il soit supprimer au bout de 14 jours et qu'il sauvegarde tout les 7 jours.

```
GNU nano 7.2 sauvegarde_complete.sh
#!/bin/bash

# Définir les variables
DB_NAME="easy bee"
DB_USER="root" # Remplacez par votre utilisateur MySQL
DB_PASSWORD="Admin1!" # Remplacez par votre mot de passe MySQL
BACKUP_DIR="/mnt/windows_backup"
DATE=$(date +%Y-%m-%d) # Date actuelle au format AAAA-MM-JJ

# Créer le dossier de sauvegarde si il n'existe pas
if [ ! -d "$BACKUP_DIR" ]; then
    mkdir -p "$BACKUP_DIR"
fi

# Effectuer la sauvegarde de la base de données
mysqldump -u $DB_USER -p$DB_PASSWORD "$DB_NAME" > "$BACKUP_DIR/${DB_NAME}backup$DATE.sql"

# Vérifier si la sauvegarde a réussi
if [ $? -eq 0 ]; then
    echo "Sauvegarde de la base de données $DB_NAME réussie le $DATE" >> "$BACKUP_DIR/backup.log"
else
    echo "Erreur de sauvegarde de la base de données $DB_NAME le $DATE" >> "$BACKUP_DIR/backup.log"
    exit 1
fi

# Supprimer les fichiers de sauvegarde de plus de 14 jours
DELETED_FILES=$(find "$BACKUP_DIR" -name "${DB_NAME}backup*.sql" -type f -mtime +14 -exec rm -f {} \;)
echo "Fichiers supprimés : $DELETED_FILES" >> "$BACKUP_DIR/backup.log"

# Ajouter une entrée de journal pour la suppression
echo "Suppression des sauvegardes de plus de 14 jours terminée le $(date +%Y-%m-%d %H:%M:%S)" >> "$BACKUP_DIR/backup.log"

exit 0
```

Modifier les privilège du fichier

chmod +x sauvegarde_complete.sh

```
adminpoisson@debian:/sauvegarde$ ls -l
total 16
drwxr-xr-x 4 root root 4096 16 oct. 2024 backup
-rwxr-xr-x 1 root root 1233 4 déc. 13:39 sauvegarde_complete.sh
-rwxr-xr-x 1 root root 1512 6 nov. 15:16 sauvegarde_incrementiel.sh
drwxr-xr-x 2 root root 4096 16 oct. 2024 suivi
adminpoisson@debian:/sauvegarde$ |
```

on automatise le script avec crontab -e

Maël Poisson

```
GNU nano 7.2 /tmp/crontab.DH1n06/crontab
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 0 * * 0 /sauvegarde/sauvegarde_complete.sh
#0 * * * * /usr/bin/python3 /mail/envoi_email.py
* * * * * /usr/local/bin/send_reservation_email.sh
```

Pour faire la sauvegarde il faut créer un partage de fichier sur le serveur de fichier.

Tout d'abord il faut installer samba sur le serveur de base de données. Voici comment on fait,

On fait d'abord les mises à jours

```
sudo apt update
```

Puis on installe samba

```
sudo apt install samba -y
```

installation du services accès au partage depuis linux

```
sudo apt-get install cifs-utils
```

Création d'un point de montage

```
sudo mkdir -p /mnt/windows_backup
```

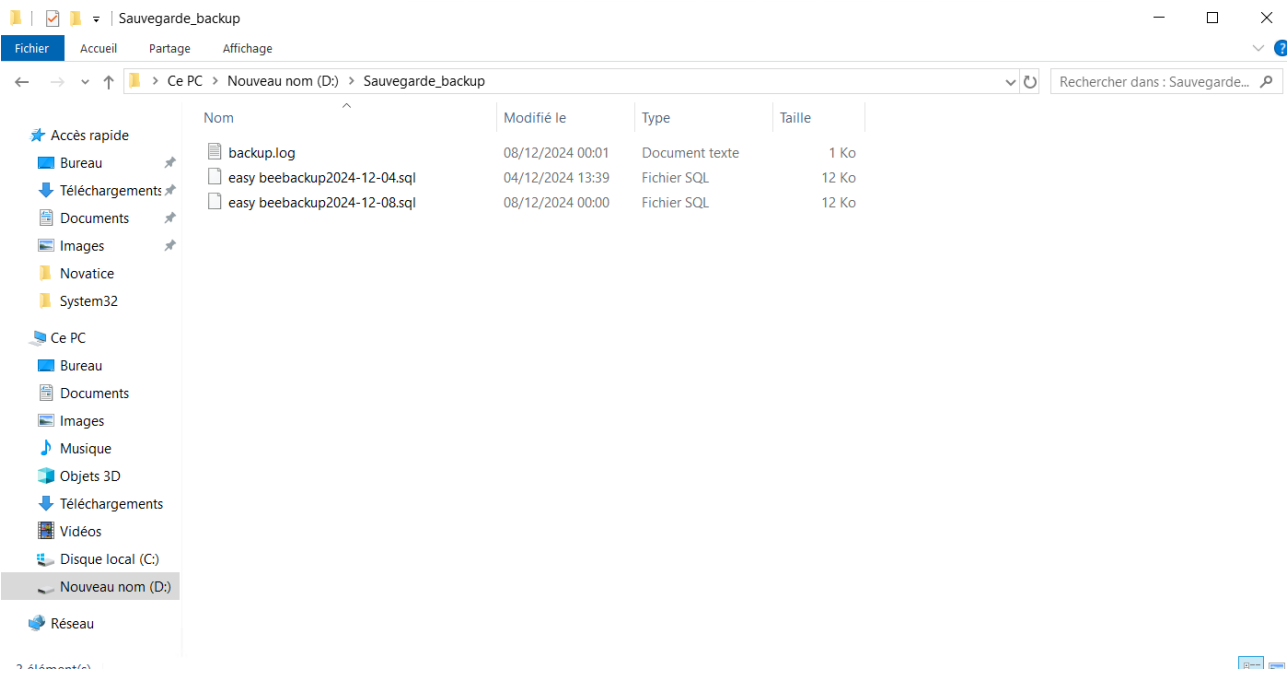
Montage du point de partage windows sur le dossier :

```
sudo mount.cifs //10.0.4.11/Sauvegarde_backup /mnt/windows_backup -o username=Administrateur,password='Admin1!',domain=10.0.4.11
```

Montage automatique du partage

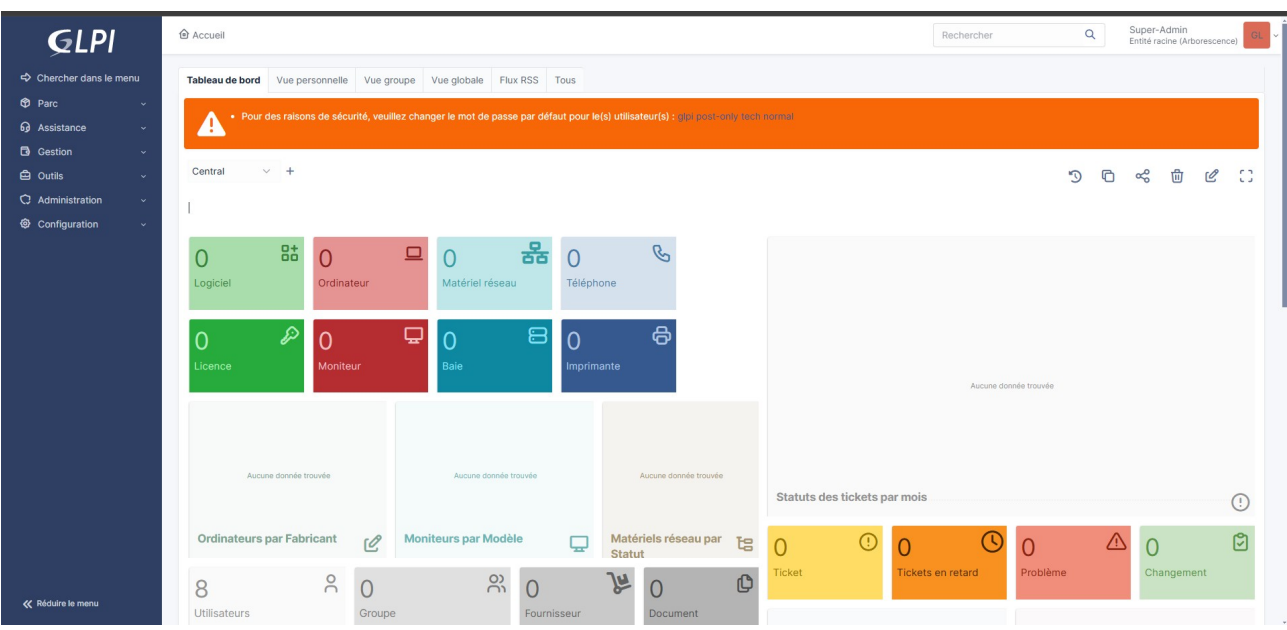
Voici les sauvegarde de la base de données

Maël Poisson



Mise en Place d'un GLPI

J'ai déjà un glpi pour cela donc je vous envoie vers un site qui permet de faire l'installation de glpi



Mise en place du serveur de mail

Installation de Postfix

Faire cette commande pour installer :

```
root@debian:/home/adminpoisson# sudo apt install postfix mailutils mariadb-client -y
```

- Postfix : serveur de mail
- mailutils : pour envoyer les mails en ligne de commande
- mariadb-client : pour interagir avec la base de données

Fichier de configuration

Fichier PHP

```
<?php
// Récupérer les arguments passés
$id = $argv[1];
$email = $argv[2];
$email_from = $argv[3];

// Contenu de l'email
$subject = "Confirmation de votre réservation";
$message = "Bonjour,\n\nVotre réservation a bien été enregistrée.\n\nCordialement,\n\nL'équipe";

// Envoi de l'email
mail($email, $subject, $message, "From: $email_from");

// Logique pour la connexion à la base de données si nécessaire
$host = 'localhost'; // Adapte selon ta configuration
$user = 'root';
$pass = 'Admin!';
$dbname = 'EasyBee P2';

$conn = new mysqli($host, $user, $pass, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Exemple de traitement avec la base de données si nécessaire
// Connexion à la base et mises à jour si nécessaires...
?>
```

Fichier sh

```
GNU nano 7.2 send_reservation_email.sh
#!/bin/bash

DB_USER="root"
DB_PASS="Admin!"
DB_NAME="EasyBee P2"
EMAIL_FROM="noreply@easybee.com"

# Récupérer les réservations en attente
RESERVATIONS=$(mariadb -u"$DB_USER" -p"$DB_PASS" -D"$DB_NAME" -sN -e "SELECT id, email FROM reservation_logs WHERE statut='en attente';")

# Vérifier s'il y a des réservations à traiter
if [ -n "$RESERVATIONS" ]; then
    IFS=$'\t' # Définir le séparateur comme une tabulation
    while read -r ID EMAIL; do
        # Appeler le fichier PHP pour envoyer l'email
        php send_reservation_email.php "$ID" "$EMAIL" "$EMAIL_FROM"

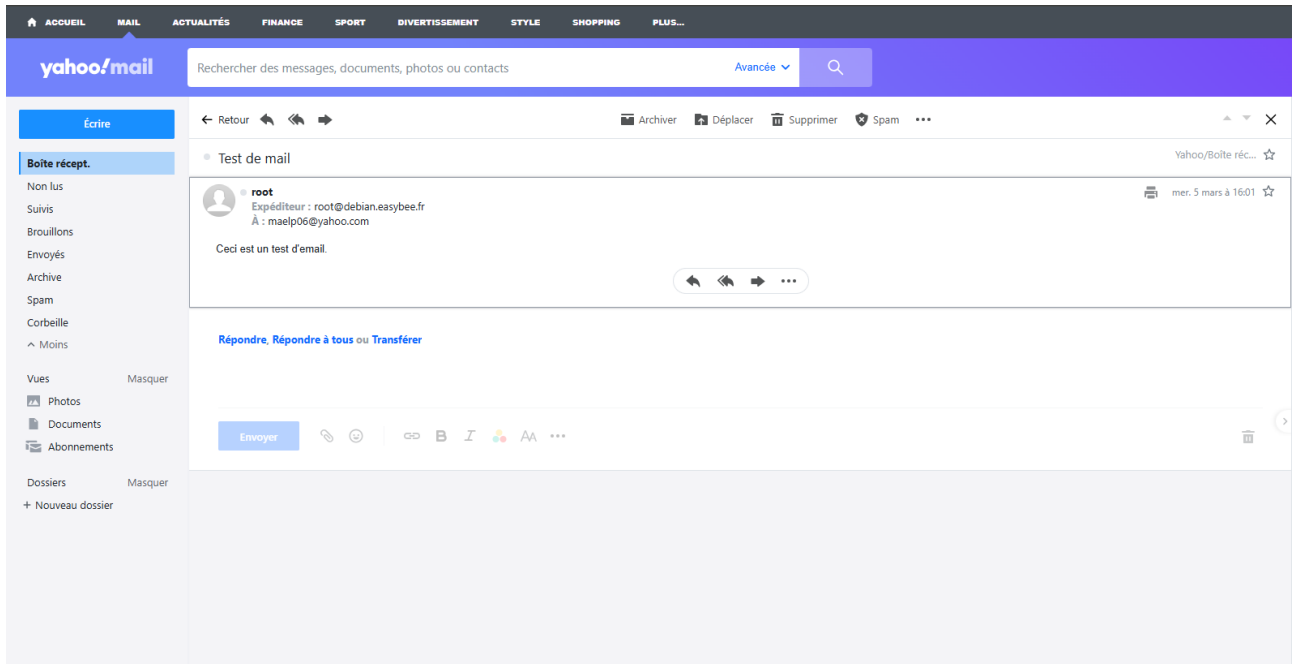
        # Marquer l'email comme envoyé
        mariadb -u"$DB_USER" -p"$DB_PASS" -D"$DB_NAME" -e "UPDATE reservation_logs SET statut='envoyé' WHERE id=$ID;"
    done <<< "$RESERVATIONS"
fi
```

Maël Poisson

Voici une commande manuel de teste mail

```
root@debian:/usr/local/bin# echo "Ceci est un test d'email." | mail -s "Test de mail" maelp06@yahoo.com
```

Voici le résultat



En revanche je n'ai pas réussi à faire le mail de réception dès qu'une personne s'inscrit à une formation