

**UNIVERSITÉ DE KINSHASA**  
**FACULTÉS DE SCIENCES ET TECHNOLOGIES**  
**MENTION MATHÉMATIQUES STATISTIQUES ET INFORMATIQUES**  
**B.P. 190 KINSHASA XI**



MINISTÈRE  
DE L'ENSEIGNEMENT  
SUPÉRIEUR, UNIVERSITAIRE,  
RECHERCHE SCIENTIFIQUE ET  
INNOVATIONS



**COURS DE SYSTÈME D'EXPLOITATION**

**GROUPE DE TRAVAIL N° 23**

**SUJET : DÉPLOIEMENT D'UN SERVEUR DE BASES DONNÉES**  
**MySQL/MariaDB avec RÉPLICATION.**

Prof : KASENGEDIA MOTUMBE Pierre  
Collaborateurs : - Doctorant Junior  
- Assistant Ferdinand

**MEMBRES DU GROUPE :**

1. NGALAMULUME TSHIMANGA Jonas
2. MPONGO ILEMBA Peter
3. LOKALEMA LOMENGO Uriel
4. KABANGU EYATO Bradel
5. KABENGELE MUDINGAY Sylvain
6. MUTOTODI SAKUAKU Agree
7. MANDI MABUNGA Jedidia
8. MANGAYA NLANDU Emdy
9. MBONGO TSUKA Geoffrey
10. MUKENDI MUKENGESHAYI Sunamite

**ANNÉE ACADÉMIQUE : 2025 - 2026**

## INTRODUCTION.

Dans le cadre de l'enseignement des Systèmes d'Exploitation, ce travail pratique vise à valider l'acquisition des compétences liées à l'administration de serveurs sous Linux. Nous abordons ici le déploiement d'un SGBD MariaDB et l'implémentation d'une stratégie de réplication pour prévenir la perte de données et assurer la haute disponibilité

Le déploiement d'un serveur de base de données revient à configurer un système de gestion de base de données pour qu'il soit opérationnel sur un ordinateur (cas de notre travail). Nous avons suivis toutes les étapes nécessaires enseignées pendant le cours pour installer, initialiser et configurer notre serveur avec MySQL via son extension MariaDB. La réplication revient à utiliser un autre pc pour pouvoir sauvegarder les données du serveur principal sur les serveurs auxiliaires ou de secours avec toutes les autorisations d'administration d'un serveur de base de données.

Nous illustrons dans ce document les différentes étapes qui nous ont conduits jusqu'à la réalisation de notre projet académique et les différents tests effectués sur nos serveurs afin de nous assurer du bon fonctionnement de la réplication du serveur principal sur les serveurs de secours. Parmi les tests nous avons effectué des ajouts et des suppressions des éléments dans une base de données de notre serveur.

Sans oublier qu'aucune œuvre humaine n'est parfaite, nous tenons à remercier ceux qui nous ont aidés et nous tenons rigueur par rapport à la façon dont le travail a été élaboré. Merci beaucoup.

## 1. INSTALLATION DU SERVEUR.

Nous avons ici installer notre serveur depuis le terminal Ubuntu, voici les différentes procédures suivies pour y aboutir :

### 1. Ouverture du terminal Bash

### 2. Commandes utilisées pour configurer le Pc maître ou serveur principal

#### - Rafraîchissement du bash

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

#### - Installation du sereveur MariaDB

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

#### - Verification de la version du sereveur installer

```
mariadb -version
```

#### - Activation du pare-feu

```
ufw enable
```

#### - Modification du fichier nano serveur

// ici c'était pour changer des parametres afin de rendre le pc maitre visible par les pcs que nous utiliserons pour stocker les serveurs de secours.

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

// Nous avons redemarrer le serveur afin d'appliquer les modifications apporter au fichiers nano de mariadb

```
sudo systemctl restart mariadb
```

#### - Lancement de mariadb

```
sudo mariadb
```

#### - Configuration du serveur

```
> CREATE USER 'replique'@'%' IDENTIFIED BY 'Password';
```

```
> GRANT REPLICATION SLAVE ON *.* TO 'replique'@'%' ;
```

```
> FLUSH PRIVILEGES ;
```

// Nous avons ici configurer tous les paramètres afin que le maître puisse répliquer sur un autre serveur

```
> SHOW MASTER STATUS ;
```

```
MariaDB [(none)]> SHOW MASTER STATUS;
```

```
.....+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB |
|-----+-----+
| mariadb-bin.000001 | 330 | | |
|-----+-----+
1 row in set (0,000 sec)
```

// La commande qui nous a permis d'avoir le nom du fichier que les serveurs de secours(esclaves) doivent utiliser pour avoir une réplique du serveur maître.

### 3. Commandes utilisées pour configurer le Pc esclave ou serveur de secours.

#### - Rafraîchissement du bash

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

#### - Installation du sereveur MariaDB

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

#### - Verification de la version du sereveur installer

```
mariadb -version
```

- Modification du fichier nano serveur

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

- Lancement de mariadb

```
sudo mariadb
```

- Configuration du serveur

```
> STOP SLAVE;  
> CHANGE MASTER TO  
→ MASTER_HOST='IP adress serveur maître',  
→ MASTER_USER='replique',  
→ MASTER_PASSWORD=' le mot de passe définit par le maître',  
→ MASTER_LOG_FILE=' le nom du fichier de connexion du maître',  
→ MASTER_LOG_POS='la position sur laquelle est connecter le  
maître' ;  
> START SLAVE;  
  
// Vérification du statut de connexion  
> SHOW SLAVE STATUS\G;
```

Nous avons donc ici créer une connexion du serveur principal avec le serveur de réplication et maintenant voici les commandes qui nous permettent une synchronisation :

```
sudo systemctl enable mariadb  
> STOP SLAVE ;  
> MASTER_CONNECT_RETRY=2 ;  
// C'est fait pour avoir une synchronisation toutes les deux secondes sur le serveur de secours.  
> START SLAVE ;
```

## 2. PROBLÉMATIQUE.

Nous avons au cours de ce travail rencontre certains problèmes en voulant déployer notre serveur, nous avons eu la plus grande difficulté en voulant connecter le pc maître avec le pc esclave suite au fait que le pare-feu du pc avec le serveur maître était désactivé et cela nous a pousser a approfondir nos recherches afin de trouver une solution pour assurer une réplication avec succès.

La réplication consiste a créer une copie de notre serveur sur un pc différent tout en restant synchroniser chaque fois pour permettre une mise à jour instantané des données du serveur. Hormis le grand problème qui était la connexion entre maître et esclave il y a eu un problème de synchronisation quand nous avons voulu utiliser deux esclaves mais nous avons tout de même continuer a tenter une réplication sur deux esclaves et notre ambition était de faire un test avec plusieurs esclave et observer comment ils se comporteraient avec tous les droits leurs étant assigner.

## CONCLUSION

Ce projet nous a permis de passer de la théorie à la pratique en utilisant le terminal Ubuntu pour configurer un système distribué. Nous avons démontré qu'une réplication est fonctionnelle par des tests d'intégrité, tout en identifiant l'importance critique de la configuration réseau et de la sécurité (pare-feu).