


## CONFIGURATION D'UN SERVEUR DE BASE DE DONNEES MYSQL PARTAGEE

### PARCOURS

SISR ☒

SLAM ☐

Lieu de réalisation	Sikiwis UERP	 <b>ERP</b> By SIKIWIS
Période de réalisation	Du : 01.10.2023	Au : 01.02.2024
Modalité de réalisation	SEUL <input type="checkbox"/>	EN EQUIPE <input checked="" type="checkbox"/>

Intitulé de la mission	Mise en place d'un serveur de base de données MySQL partagée
Description du contexte de la mission	Mise en place d'un serveur de base de données MySQL partagé afin que chaque développeur puisse se mettre à jour facilement

Contraintes & Résultat	Ressources fournies / contraintes techniques / Résultats attendu
	VM Debian, MariaDB, MySQL, GitLab
Productions associées	Liste des documents produits et description

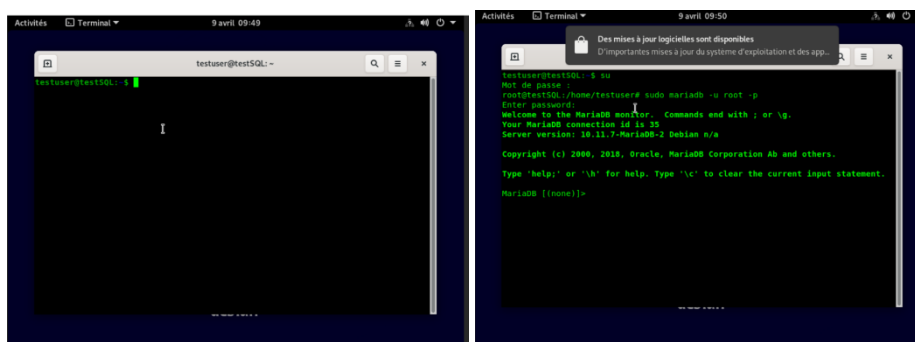
Modalités d'accès aux productions	Identifiants, mots de passe, URL d'un espace de stockage et présentation de l'organisation du stockage

## Configuration d'un serveur de base de données MySQL partagée

Originellement, la structure de réseau et de développement est agencée tel que :

- les développeurs possèdent sur leurs postes la solution en “*local*”, sur laquelle ils peuvent faire des tests et développer des modifications et répondre aux tickets de bugs des chefs de projet.
- lorsque les développements ont été acceptés par le “*lead dev*”, ceux-ci sont *push* sur le Git de l'entreprise et pris par l'ensemble des autres développeurs lorsque ceux-ci se mettent à jour en début de journée quand ils “*fetch*” les changements.
- Toutefois, il peut y avoir des problèmes, d'une part avec le manque de coordination entre développeurs qui agissent simultanément sur les mêmes pages dans leurs solutions en local et pushent des modifications qui créent des conflits.
- D'autre part, en dehors du local de chaque développeur, la solution est directement accueillie sur un serveur distant en production, donc lorsque des modifications ont été acceptées, elles sont directement implémentées sur le serveur de production, avec de possibles bugs qui n'ont pas été “*recettés*” en amont par les chefs de projet.

Ainsi, il y a eu besoin d'un nouveau serveur, de test, pour accueillir une base de données commune. En utilisant le serveur acheté précédemment (*synology*), en lui allouant une partie du stockage, on crée une machine virtuelle (VM) Linux (Debian 11), sur laquelle on installe et configure MariaDB. Par la suite, on déploie les bases de données et les données de test, et on restreint l'accès par adresse MAC à l'aide du domaine Active Directory pour les postes des développeurs.



Connexion sur la VM Debian. Ouverture du terminal. Connexion à la base de données MariaDB en ligne de commande.

Par la suite, on crée une seconde VM, avec Windows server 2022, sur laquelle on installe et configure un serveur web IIS, sur lequel on déploie la solution de l'entreprise, et on régleme également l'accès à cette VM aux seuls utilisateurs développeurs de l'AD.

La dernière étape consiste en l'intégration/développement continu (CI-CD).

On met à jour automatiquement, d'une part la base de données de test lorsqu'une nouvelle modification est “push” sur le Git de l'entreprise, et d'autre part la solution déployée sur le serveur IIS. Cette mise à jour est automatiquement extraite sous la forme d'un fichier texte, dans un dossier partagé accessible aux seuls utilisateurs AD développeurs. L'intégration continue s'opère en mettant en place des séries de tests unitaires sur les fonctionnalités principales du logiciel et s'exécutant sur l'environnement de test après le déploiement.

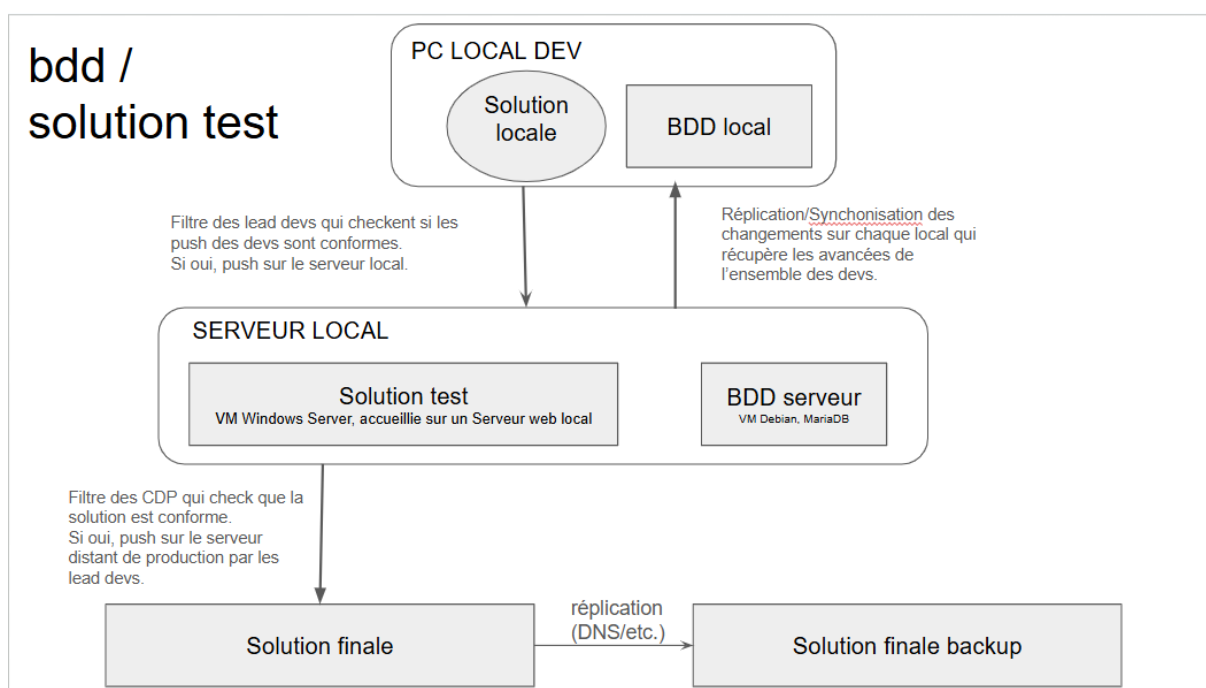
## Configuration d'un serveur de base de données MySQL partagée

### Introduction

Dans le cadre de mon stage en entreprise, j'ai participé à une mission visant à améliorer les processus de développement, de test et de déploiement d'une solution web. Cette mission s'inscrit dans un contexte où l'entreprise souhaitait résoudre plusieurs problèmes liés à la gestion des modifications sur ses solutions logicielles.

La mise en place d'un serveur de test et d'un processus d'intégration continue (CI/CD) a permis d'offrir un environnement centralisé et sécurisé, réduisant les risques de conflits entre développeurs et les erreurs en production.

Ce rapport détaille les différentes étapes de la mission, de la création des machines virtuelles (VM) à la configuration des bases de données, jusqu'à l'automatisation des déploiements et des tests.



## Configuration d'un serveur de base de données MySQL partagée

### Objectifs de la mission

- Configurer une machine virtuelle Linux (Debian 11) avec MariaDB pour accueillir une base de données de test commune.
- Déployer une machine virtuelle Windows Server 2022 avec IIS pour héberger la solution de l'entreprise.
- Restreindre l'accès aux deux serveurs aux seuls développeurs via un domaine Active Directory.
- Mettre en place un processus CI/CD pour synchroniser les données et déployer automatiquement les modifications du Git de l'entreprise.
- Valider le bon fonctionnement de l'environnement via des tests unitaires sur les fonctionnalités principales.

### Ressources mises à disposition

- Serveur physique (Synology) pour héberger les machines virtuelles.
- Système de gestion des versions (Git).
- Domaine Active Directory configuré.
- Images ISO de Debian 11 et Windows Server 2022.
- Solution logicielle de l'entreprise et jeux de données de test.

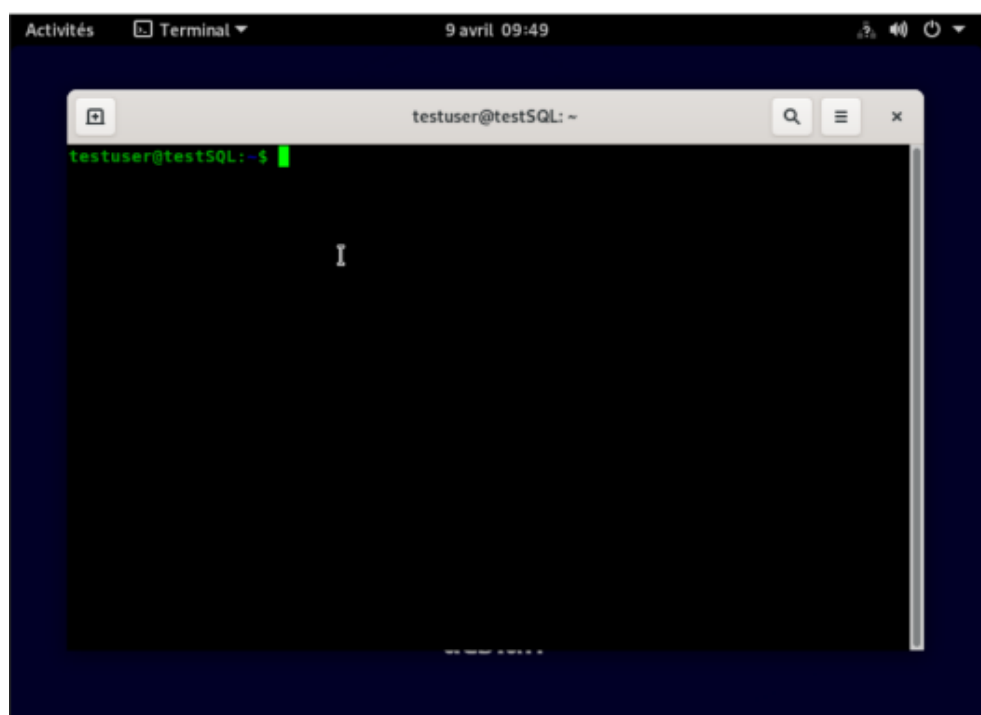
## Configuration d'un serveur de base de données MySQL partagée

### 1) Création et configuration de la VM Linux avec MariaDB

#### A) Création d'une machine virtuelle Debian 11 sur le serveur Synology

La première étape consiste à créer une machine virtuelle sur le serveur Synology. Pour ce faire, on utilise l'outil de virtualisation intégré (Virtual Machine Manager).

On configure une VM Debian 11 avec les paramètres suivants : allocation de ressources matérielles (CPU, RAM et espace disque) adaptées aux besoins de la base de données, et configuration du réseau pour qu'elle soit accessible depuis le réseau local. Une fois l'installation de Debian 11 terminée, les mises à jour du système sont appliquées pour garantir la sécurité et la stabilité.



Capture d'écran d'une machine virtuelle accueillant Debian.

## Configuration d'un serveur de base de données MySQL partagée

### B) Installation de MariaDB

La prochaine étape consiste en l'installation de MariaDB sur la VM accueillant Debian 11. MariaDB est un SGBDR, système de gestion de base de données relationnelles (RDBMS en anglais, relational database management system). C'est une solution open-source dérivée de MySQL (fork). L'installation de MariaDB sur Debian 11 est réalisée via le gestionnaire de paquets APT.

Afin d'installer un serveur et un client MariaDB sur la machine, il faut lancer la commande suivante.

```
root@debian:~# apt install mariadb-server mariadb-client -y
```

On vérifie les statuts de l'installation de MariaDB. La solution a bien été installée et est fonctionnelle.

```
root@debian:~# 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 Mon 2024-11-25 09:39:31 CET; 25s ago
  Docs: man:mariadb(8)
        https://mariadb.com/kb/en/library/systemd/
  Process: 559 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql
  Process: 562 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSI
  Process: 565 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||
  Process: 636 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSI
  Process: 638 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
```

Une fois la configuration sécurisée terminée, une base de données spécifique est créée pour les tests. On importe la base de données entreprise dans la machine nouvellement créée. On appelle ça un « dump ».

```
root@debian:~# mysql -u root -p entreprise <F:\dump.sql
```

## Configuration d'un serveur de base de données MySQL partagée

### C) Sécuriser les accès au serveur de base de données

#### a) Intégration du serveur de base de données dans le domaine

La première étape consiste à intégrer le serveur au domaine. Il faut d'abord s'assurer que celui-ci fasse partie du même réseau que celui du contrôleur de domaine. Ensuite, il faut le faire intégrer le domaine, en ligne de commandes. Pour intégrer Debian au domaine AD, on doit installer certains outils.

```
root@debian:~# apt install realmd sssd adcli krb-user packagekit samba-common-bin
```

Voici à quoi correspondent ces commandes :

- realmd : facilite la découverte et l'intégration au domaine AD.
- sssd : gère l'authentification des utilisateurs AD.
- krb5-user : permet de gérer les tickets Kerberos pour l'authentification.

On cherche le domaine et on y accède.

```
root@debian:~# realm discover sikiwis.local
```

```
root@debian:~# realm join --user=admin sikiwis.local
```

Pour restreindre l'accès à des utilisateurs ou groupes spécifiques du domaine AD, on modifie les paramètres de realmd :

```
root@debian:~# realm permit -g developpeurs
```

Après l'intégration de Debian au domaine, configurez MariaDB pour permettre aux utilisateurs AD d'accéder à la base de données.

```
root@debian:~# apt install mariadb-plugin-pam
```

## Configuration d'un serveur de base de données MySQL partagée

### b) Filtrage par adresses MAC

En complément de l'authentification AD, le filtrage par adresses MAC renforce la sécurité en limitant l'accès réseau aux seules machines autorisées.

Si le filtrage MAC doit être appliqué directement sur la VM Debian, on configure iptables pour autoriser uniquement les adresses MAC spécifiées.

```
root@debian:~# iptables -A INPUT -p tcp --dport 3306 -m mac --mac-source [redacted] -j ACCEPT
```

On bloque toutes les autres connexions.

```
root@debian:~# iptables -A INPUT -p tcp --dport 3306 -j DROP
```



## Configuration d'un serveur de base de données MySQL partagée

### 2) Création et configuration de la VM Windows Server

#### A) Déploiement d'une VM Windows Server

On installe une VM Windows Server sur le serveur Synology dans le parc de machines virtuelles, mis à disposition dans les applications du serveur.

#### B) Installation du serveur web IIS

Une fois la VM Windows Server créée, on installe le rôle de serveur web dessus.

#### C) Déploiement de la plateforme

On installe le code source de la plateforme sur le serveur web.

#### D) Ouverture de l'accès par domaine et MAC aux développeurs

L'accès au site web est permis en local par http, et restreint aux membres du groupe « développeurs » du domaine et par adresse MAC.

### 3) Gérer la communication entre serveurs

#### A) Ouvrir un flux entre WS et BDD

On ouvre un flux entre la machine accueillant le serveur Windows et celle accueillant la base de données.

## Configuration d'un serveur de base de données MySQL partagée

### 4) Déploiement continu

#### A) MAJ automatique de la BDD lorsque modif push sur le Git

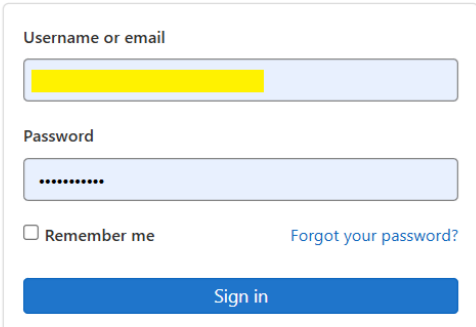
Le code source de la plateforme et de la base de données de Sikiwis sont contenus dans une instance privée Gitlab.

## GitLab

### A complete DevOps platform

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security.

This is a self-managed instance of GitLab.

A screenshot of the GitLab login interface. It features two input fields: 'Username or email' and 'Password'. The 'Username or email' field contains a yellow bar. The 'Password' field contains a series of dots. Below the password field, there is a checkbox labeled 'Remember me' and a link labeled 'Forgot your password?'. At the bottom, there is a blue button labeled 'Sign in'.

Gitlab est une plateforme permettant de partager du code de manière collaborative, en trackant les changements opérés et en permettant de « pousser » ces changements locaux vers une instance partagée. D'autres plateformes existent comme Git, Github ou Bitbucket.

La particularité de Gitlab est qu'il permet en plus de pouvoir opérer des actions de CI/CD, « continuous development/continuous integration », développement continu et intégration continue, c'est-à-dire d'automatiser des actions par rapport au code source partagé afin d'automatiser et de faciliter la tâche du développeur.

On créer un script d'automatisation qui exécute les fichiers SQL sur la base de données de test, et on l'intègre avec un fichier yml.

## Configuration d'un serveur de base de données MySQL partagée

### **B) MAJ automatique de la plateforme lorsque modif push sur le Git**

On crée un script à l'aide de Powershell pour déployer l'application sur le serveur web IIS. On ajoute un script dans un fichier yml.

### **Conclusion**

La mise en place de cet environnement de développement et de test, intégrant des solutions modernes telles que MariaDB, IIS, et un pipeline CI/CD, permet d'améliorer considérablement la collaboration et la productivité des équipes de développement. Grâce à l'automatisation des mises à jour et des déploiements, ainsi qu'à la centralisation des données de test, les développeurs disposent désormais d'un environnement fiable et représentatif pour valider leurs modifications avant leur mise en production.

Cette infrastructure réduit les risques liés aux erreurs humaines, renforce la sécurité grâce à l'intégration Active Directory et au filtrage MAC, et facilite la gestion des projets grâce à une visibilité accrue sur les étapes de développement et de test. En outre, les tests unitaires intégrés au pipeline garantissent que les fonctionnalités principales de l'application sont validées en continu, améliorant ainsi la qualité du produit final.

En centralisant les efforts autour de processus bien définis, l'entreprise se dote des outils nécessaires pour répondre aux exigences de ses clients tout en renforçant la satisfaction des équipes internes, qui gagnent en efficacité et en sérénité.