


CONFIGURATION D'UN SERVEUR DE BASE DE DONNEES MYSQL PARTAGEE

PARCOURS	SISR <input checked="" type="checkbox"/>	SLAM <input type="checkbox"/>
-----------------	---	--------------------------------------

Lieu de réalisation	Sikiwis UERP	 ERP By SIKIWIS
Période de réalisation	Du : 01.10.2023	Au : XX.XX.XXXX
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

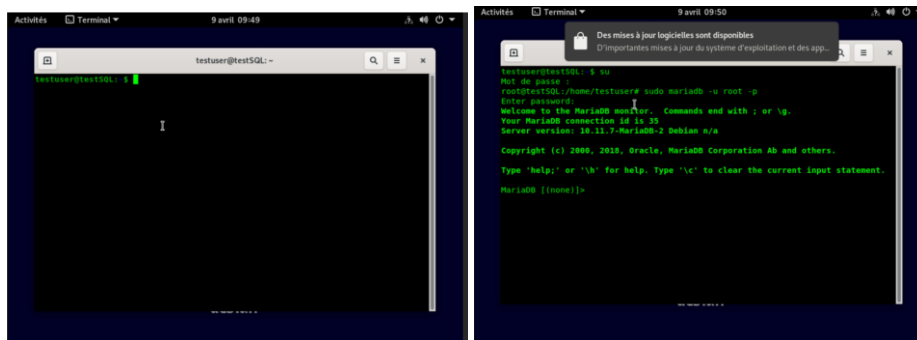
Modalités d'accès aux productions	Identifiants, mots de passe, URL d'un espace de stockage et présentation de l'organisation du stockage

CONFIGURATION D'UN SERVEUR DE BASE DE DONNEES MYSQL PARTAGEE

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.