

Mise en place d'un outil de gestion de projet (Redmine)



Table des matières / Sommaire

1. Cahier des charges – Expression des besoins	3
2. Ressources.....	4
3. Analyse	5
4. Mise en place.....	6
5. Bilan.....	7

1. Cahier des charges – Expression des besoins

Descriptif de l'existant

Dans le cadre de la création d'un nouveau SI (Système d'Information) pour une structure en pleine croissance, il n'existe actuellement aucun outil de suivi technique centralisée. L'infrastructure est en phase de construction : les services de base (système d'exploitation et serveur Web) viennent d'être déployés, mais la couche applicative dédiée à la gestion des opérations reste à définir. Actuellement, la coordination des tâches se fait par échanges verbaux, ce qui est incompatible avec une démarche de qualité et de traçabilité professionnelle.

Besoin(s)

Le projet consiste à implémenter une solution "tout-en-un" pour piloter cette nouvelle infrastructure. Les besoins prioritaires sont :

- **Structuration du SI** : Mettre en place un outil de gestion de projet capable de supporter l'ensemble du cycle de vie des services informatiques.
- **Ticketing** : Disposer d'un centre de services pour enregistrer, prioriser et résoudre les incidents ou les demandes de changement.
- **Documentation collaborative** : Créer une base de connaissances (Wiki) pour archiver les schémas réseau et les procédures d'installation.
- **Reporting** : Permettre à la direction technique de visualiser la charge de travail et le respect des délais via des outils de pilotage (Gantt, feuilles de temps).

Contrainte(s)

Pour cette nouvelle installation, les contraintes retenues sont :

- **Intégration native** : La solution doit s'appuyer sur la pile logicielle standard (LAMP) installée sur nos nouveaux serveurs Linux.
- **Évolutivité** : L'outil doit être capable d'absorber une montée en charge (augmentation du nombre de projets ou d'utilisateurs) sans remise en cause de l'architecture.
- **Coût de possession (TCO)** : Privilégier une solution Open Source robuste pour éviter les coûts de licence récurrents dès le lancement de l'activité.
- **Confidentialité** : Hébergement local obligatoire pour garantir que les données stratégiques de l'entreprise restent au sein du périmètre de sécurité interne.

2. Ressources

Ressources mises à disposition

- **Système d'exploitation** : Un serveur fraîchement déployé sous la distribution **Debian 13 (Trixie)**, configuré avec un accès administrateur (Sudo) pour l'installation des composants applicatifs.
- **Service Web** : Une instance du serveur HTTP **Apache2** opérationnelle, destinée à servir d'interface pour l'application de gestion de projet.
- **Accès réseau** : Une connexion internet haut débit permettant la communication avec les serveurs de dépôts officiels pour le téléchargement des dépendances Ruby on Rails et des paquets de l'application.
- **Environnement de stockage** : Un espace disque dédié sur un SSD performant pour garantir la rapidité d'accès aux pièces jointes et à la base de données documentaire.

Ressources nécessaires

- **Moteur de base de données** : Le SGBD **MariaDB** (ou PostgreSQL) pour le stockage structuré des tickets, des utilisateurs et des configurations de projets.
- **Langages et dépendances** : L'environnement d'exécution **Ruby** ainsi que le framework **Ruby on Rails**, indispensables au fonctionnement interne de l'outil.
- **Connecteur d'application** : Le module **Passenger** (ou FastCGI) pour permettre la communication fluide entre le serveur Web Apache2 et le code de l'application.
- **Postes clients** : Des stations de travail équipées de navigateurs Web modernes pour effectuer les tests de recette et la configuration initiale de l'interface d'administration.

Gestion des ressources

La gestion des ressources logicielles est entièrement confiée au gestionnaire de paquets **APT**, assurant ainsi la cohérence des versions et la facilité des futures mises à jour de sécurité. Pour la partie matérielle, les ressources de calcul (CPU et RAM) font l'objet d'une surveillance particulière lors de la phase d'installation, les processus de compilation des modules Ruby pouvant être momentanément gourmands. Enfin, l'intégrité des données sera gérée par la mise en place de scripts de sauvegarde automatique de la base de données SQL.

3. Analyse

Descriptifs des solutions

Pour répondre au besoin de gestion de projet et de ticketing, plusieurs types de solutions ont été étudiés :

- **Solutions SaaS (Cloud)** : Des outils comme Jira ou Trello, hébergés par des tiers, offrant une mise en place immédiate mais avec des coûts d'abonnement récurrents et une dépendance vis-à-vis d'un prestataire externe.
- **Solutions On-Premise (Auto-hébergées)** : Des logiciels comme **Redmine**, installés directement sur nos propres serveurs Debian. Cette approche garantit la souveraineté des données et une personnalisation totale de l'outil.

Comparaison des solutions

Critère	Trello (SaaS)	Jira (SaaS/Cloud)	Redmine (Choisi)
Coût	Gratuit limité / Payant	Élevé (Abonnement)	Gratuit (Open Source)
Hébergement	Cloud externe	Cloud externe	Local (Souveraineté)
Ticketing	Basique (Cartes)	Très complet	Complet et flexible
Documentation	Non (nécessite tiers)	Via Confluence (Payant)	Wiki intégré nativement
Installation	Aucune	Aucune	Nécessite une configuration

Choix d'une solution

Le choix s'est porté sur **Redmine**. Malgré une installation plus technique que les solutions Cloud, il répond à 100% à nos contraintes de souveraineté et de budget. Son intégration parfaite avec le serveur **Apache2** déjà en place et sa capacité à centraliser à la fois les tickets et la documentation (Wiki) en font l'outil idéal pour piloter une infrastructure neuve sans multiplier les abonnements logiciels.

Étude de l'impact sur le SI existant

Le déploiement de Redmine a plusieurs impacts bénéfiques et structurels sur le système d'information :

- **Performance** : L'utilisation de Ruby on Rails demande une allocation de mémoire vive (RAM) stable. Le serveur Debian a été dimensionné pour supporter cette charge sans dégrader les autres services.
- **Accessibilité** : Le SI devient pilotable à distance. Un administrateur peut suivre les incidents depuis son **MacBook Air M4** ou son **PC Windows 11** via une simple interface Web sécurisée par le VPN.

- **Disponibilité** : En centralisant les procédures dans le Wiki, on réduit le "Single Point of Failure" humain : toute l'équipe possède le même niveau d'information pour intervenir en cas de panne.
- **Sécurité** : L'authentification centralisée sur Redmine permet de tracer "qui fait quoi" sur le SI, renforçant ainsi la responsabilité et l'auditabilité des interventions.

4. Mise en place

Réalisation en suivant le phasage énoncé précédemment

Étape	Description	Images
1	<p>Préparation : Mise à jour des dépôts et installation du serveur de base de données MariaDB.</p> <p>sudo apt update && sudo apt install mariadb-server -y</p>	 <pre>root@debian:~# apt install mariadb-server -y Installation de : mariadb-server</pre>
2	<p>Base de données : Connexion au shell SQL et création de la base de données pour Redmine.</p> <p>sudo mysql -u root CREATE DATABASE redmine CHARACTER SET utf8mb4; exit</p>	 <pre>MariaDB [(none)]> CREATE DATABASE redmine CHARACTER SET utf8mb4; Query OK, 1 row affected (0,001 sec)</pre>
3	<p>Installation : Installation des paquets Redmine et du module de liaison MySQL.</p> <p>sudo apt install redmine redmine-mysql -y</p>	 <p>0011 de configuration des paquets</p> <p>Configuration de redmine</p> <p>Le paquet redmine/instances/default a besoin d'une base de données installée et configurée avant de pouvoir être utilisé. Ceci peut, si nécessaire, être géré par dbconfig-common.</p> <p>Si vous êtes un administrateur de bases de données expérimenté et savez que vous voulez procéder à cette configuration vous-même, ou si votre base de données est déjà installée et configurée, vous pouvez refuser cette option. Des précisions sur la procédure se trouvent dans /usr/share/doc/redmine/instances/default.</p> <p>Autrement, vous devriez choisir cette option.</p> <p>Faut-il configurer la base de données de redmine/instances/default avec dbconfig-common ?</p> <p><input checked="" type="radio"/> Oui <input type="radio"/> Non</p>
4	<p>Lien Apache : Installation du module Passenger pour permettre à Apache de lire le code Ruby.</p> <p>sudo apt install libapache2-mod-passenger -y</p>	 <pre>root@debian:~# apt install libapache2-mod-passenger -y Installation de : libapache2-mod-passenger</pre>
5	<p>Activation : Création du lien symbolique vers le dossier web, activation du module et redémarrage final.</p> <p>sudo ln -s /usr/share/redmine/public /var/www/html/redmine</p> <p>sudo a2enmod passenger && sudo systemctl restart apache2</p>	 <pre>Module passenger already enabled root@debian:~# systemctl restart apache2 root@debian:~#</pre> <p>bitnami</p> <p>Awesome! Redmine is now installed.</p> <p>Access Redmine</p> <p>You just installed Redmine using Bitnami - the fastest, easiest and most secure way to deploy your favorite app.</p> <ul style="list-style-type: none"> • Visit the Bitnami Redmine documentation page for more information on managing your installation. • Access the built-in management tool <code>phpMyAdmin</code>. For security reasons it is only accessible when using 127.0.0.1 as the hostname. • Find answers to common questions or pose a new question on our community forum. <p>Thanks for using Bitnami!</p>

Rapport de tests

Test de conformité	Action effectuée	Résultat attendu	Résultat obtenu
Accès distant	Saisie de l'adresse IP du serveur sur le port 80	Affichage de la page d'accueil Redmine	OK
Stabilité DB	Création d'un premier compte utilisateur	Enregistrement immédiat dans la base MariaDB	OK

5. Bilan

Conclusion

La mise en place de l'outil de gestion de projet **Redmine** est un succès. L'application est désormais fonctionnelle et accessible depuis l'ensemble du réseau local via une interface Web sécurisée. Ce déploiement répond parfaitement au cahier des charges initial en offrant une solution centralisée pour le suivi des tickets, la planification des tâches et la documentation technique (Wiki), le tout hébergé sur notre propre infrastructure **Debian 13**.

Auto-critique / Auto-évaluation

- **Points forts** : L'intégration de Redmine avec le serveur **Apache2** existant permet d'économiser des ressources serveur en évitant de multiplier les machines virtuelles. L'utilisation du paquet dbconfig-common a grandement simplifié la configuration complexe de la base de données.
- **Difficultés rencontrées** : La liaison entre Apache et Ruby on Rails via le module **Passenger** est l'étape la plus délicate. Elle nécessite une configuration précise des liens symboliques et des droits d'accès pour éviter les erreurs "403 Forbidden".
- **Points de vigilance** : Redmine étant un outil critique pour l'organisation du travail, une politique de sauvegarde rigoureuse de la base de données **MariaDB** devra être automatisée rapidement (via un script Cron par exemple).

Compétence(s) SISR mobilisée(s)

Ce projet valide les compétences suivantes du référentiel BTS SIO (Option SISR) :

- **Gérer le patrimoine informatique** (Mise en place d'un outil de gestion de parc et d'incidents).
- **Répondre aux incidents et aux demandes d'assistance et d'évolution** (Installation de la solution de ticketing).
- **Développer la présence en ligne de l'organisation** (Déploiement d'un service Web collaboratif).