

MISE EN PLACE D'UN SERVEUR GLPI



I/ Cahier des charges	2
1- Descriptif de l'existant	2
2- Besoins.....	2
3- Contraintes	2
II/ Ressources.....	2
1- Ressources mises à disposition.....	2
2- Ressource nécessaire à la mise en place	3
3- Gestion des ressources	3
III/ Analyse.....	3
1- Descriptifs des solutions	3
2- Comparaisons des solutions	4
3- Choix d'une solution	4
4- Cartographie du réseau	5
5- Etude de l'impact sur le SI existant.....	5
6- Phasage de l'intervention	5
7- Prévision des tests de validation.....	6
IV/ Mise en place.....	6
1- Réalisation et rapport de tests.....	6
2- Rapport de déploiement.....	6
V/ Bilan	7
1- Conclusion	7
2- Auto-évaluation	7

I/ Cahier des charges

1- Descriptif de l'existant

Le projet consiste à mettre en place un serveur GLPI dans un environnement entièrement virtualisé afin de simuler l'infrastructure d'une petite entreprise. L'infrastructure repose sur un seul poste physique hôte équipé d'un hyperviseur, sur lequel sont créées trois machines virtuelles :

- **Debian 13** : serveur GLPI et pile LAMP,
- **pfSense** : pare-feu et passerelle réseau,
- **Windows 10 (ou 11)** : poste client pour les tests et l'agent GLPI.

Cette organisation permet de reproduire un réseau d'entreprise avec un serveur applicatif, une sécurité périmétrique et un poste utilisateur, tout en restant sur une seule machine physique.

2- Besoins

L'installation de GLPI répond à plusieurs besoins :

- Inventorier et centraliser le parc informatique (machines, périphériques, logiciels, licences),
- Offrir un outil de ticketing pour la gestion des incidents et demandes des utilisateurs,
- Faciliter le suivi des interventions et des équipements via un tableau de bord unique,
- Améliorer la traçabilité et la maintenance du système d'information.

Ces besoins s'inscrivent dans une logique d'optimisation de la gestion du SI en rendant les informations facilement accessibles et en réduisant le temps de traitement des incidents.

3- Contraintes

Le projet s'est déroulé dans un cadre pédagogique, impliquant plusieurs contraintes :

- **Techniques** : ressources limitées du poste hôte (RAM, CPU, espace disque), absence d'environnement graphique sur Debian pour économiser les ressources,
- **Organisationnelles** : temps limité pour le déploiement (2h à partir de notes prises précédemment),
- **Réseau** : absence d'accès direct à un véritable SI d'entreprise ; nécessité de simuler le réseau via pfSense et de travailler dans un environnement fermé,
- **Sécurité** : mise en œuvre obligatoire du protocole HTTPS pour chiffrer les échanges entre le client et le serveur GLPI.

II/ Ressources

1- Ressources mises à disposition

Les ressources matérielles et logicielles utilisées pour ce projet reposent sur un ordinateur physique hôte équipé d'un hyperviseur (VMWare) permettant de créer plusieurs machines virtuelles. Les images ISO de Debian 13, de Windows 10 et de pfSense ont été fournies pour l'installation des différents systèmes. L'accès au réseau WAN a été indispensable afin de pouvoir télécharger les paquets nécessaires à Debian et récupérer l'archive d'installation de GLPI et sa documentation.

2- Ressource nécessaire à la mise en place

La mise en œuvre a nécessité la création de trois machines virtuelles distinctes. Le serveur Debian 13 héberge la pile LAMP et GLPI, pfSense joue le rôle de pare-feu et de passerelle réseau entre le LAN simulé et l'accès WAN, tandis que le poste client Windows 10 permet d'effectuer les tests d'accès web et de vérifier la remontée de l'inventaire via l'agent GLPI. Un plan d'adressage IP fixe a été défini afin de garantir la communication entre ces différentes machines et avec Internet via pfSense (cf. partie III/4.)

3- Gestion des ressources

L'ensemble du projet a été déployé sur un PC physique unique hébergeant plusieurs machines virtuelles. Deux de ces VM ont été utilisées de manière complémentaire : la première exécute Debian 13, qui héberge tous les services (Apache, MariaDB, PHP, GLPI et HTTPS), tandis que la seconde exécute Windows, destinée aux tests fonctionnels, à la connexion au serveur et à l'installation de l'agent GLPI. Pour simuler un environnement réseau plus réaliste, nous avons également mis en place une machine virtuelle pfSense faisant office de pare-feu et de passerelle réseau. Cette configuration permet de reproduire la topologie d'un réseau d'entreprise, d'assurer un routage correct entre les VM et de tester les communications dans des conditions proches de la production. Une attention particulière a été portée à l'attribution des droits d'accès sur le répertoire web `/var/www/html/glpi` afin de garantir le bon fonctionnement d'Apache. Les configurations réseau et les règles du pare-feu pfSense ont été soigneusement documentées afin de pouvoir reproduire la mise en place et faciliter le dépannage.

III/ Analyse

1- Descriptifs des solutions

L'objet de ce projet est de trouver une solution logicielle afin de répondre au besoin de ticketing dans une infrastructure mais également de la gestion du parc informatique.

Pour cela il existe plusieurs solutions pour y répondre. Nous n'allons pas tous les citer mais nous pouvons au moins présenter les plus pertinents :

- **GLPI** : GLPI (Gestionnaire Libre de Parc Informatique) est une solution open source écrite en PHP, sous licence GPL, qui combine la gestion des actifs informatiques (hardware, software, réseaux), la gestion des tickets (helpdesk, incidents/demandes) et la gestion financière/contractuelle. Il permet également de faire de la CMDB¹ (base de données de configuration), de gérer les licences, contrats, fournisseurs, etc., et dispose d'un vaste écosystème de plugins pour étendre les fonctionnalités
- **ITop** : iTop est un outil open-source conçu pour la gestion des services IT et des demandes de support. Il est plus léger que GLPI et permet une gestion centralisée des tickets et des incidents. Il permet de gérer les incidents, les changements, les services, et de modéliser les éléments de configuration (CI), c'est-à-dire les matériels, les logiciels, les dépendances entre services. Bon pour les organisations qui veulent un outil structuré et évolutif.

¹ **Configuration Management Database** : C'est un référentiel central qui contient toutes les informations sur les éléments constitutifs du système d'information (appelés CI, pour Configuration Items) et leurs relations. Ces éléments peuvent être des serveurs, postes de travail, logiciels, périphériques réseau, services IT, contrats, etc.

- **Jira** : Jira est une solution professionnelle de gestion des services et du support IT développée par Atlassian. Initialement conçu pour le suivi de projets et de bugs, Jira a évolué en un outil ITSM (IT Service Management) complet. Il permet de gérer efficacement les tickets d'incidents, les demandes de service, les changements et de suivre les SLA² grâce à des workflows³ automatisés et personnalisables. Disponible en SaaS (cloud), il est très apprécié dans les grandes organisations pour sa souplesse et ses nombreuses intégrations, mais il est payant et peut être plus complexe à configurer qu'une solution open source comme GLPI.

2- Comparaisons des solutions

Critère	GLPI	Jira Service Management	iTop
Gestion des tickets / helpdesk	Intégré : incidents, demandes, SLA.	Avancé : workflows puissants, SLA, portail self-service.	Conforme ITIL : incidents, demandes, changements.
Inventaire matériel / logiciels / CI	Complet : matériels, logiciels, licences.	Bon via module Assets, pas natif.	Basé CMDB : nécessite outils externes pour inventaire auto.
CMDB / relations d'impact	Présente mais basique.	Via Assets, dépend du paramétrage.	Très avancée : relations, dépendances et analyse d'impact.
Extensibilité / Plugins	Nombreux plugins (FusionInventory, docs, reporting).	Large marketplace, surtout pour cloud payant.	Personnalisable : modèles de données, workflows.
Sécurité	Gestion HTTPS facile, plugins pour audit et rôles.	Conforme entreprise : SSO, MFA, mais réservé aux versions payantes.	Gestion des accès et rôles, nécessite configuration.
Coût / licence	Gratuit et open source.	Payant, abonnement mensuel.	Mise en œuvre plus longue : nécessite réflexion CMDB.
Facilité pédagogique / petit projet	Simple, interface claire, idéal pour petits projets.	Plus complexe, nécessite formation et paramétrage.	Simple, interface claire, idéal pour petits projets.

3- Choix d'une solution

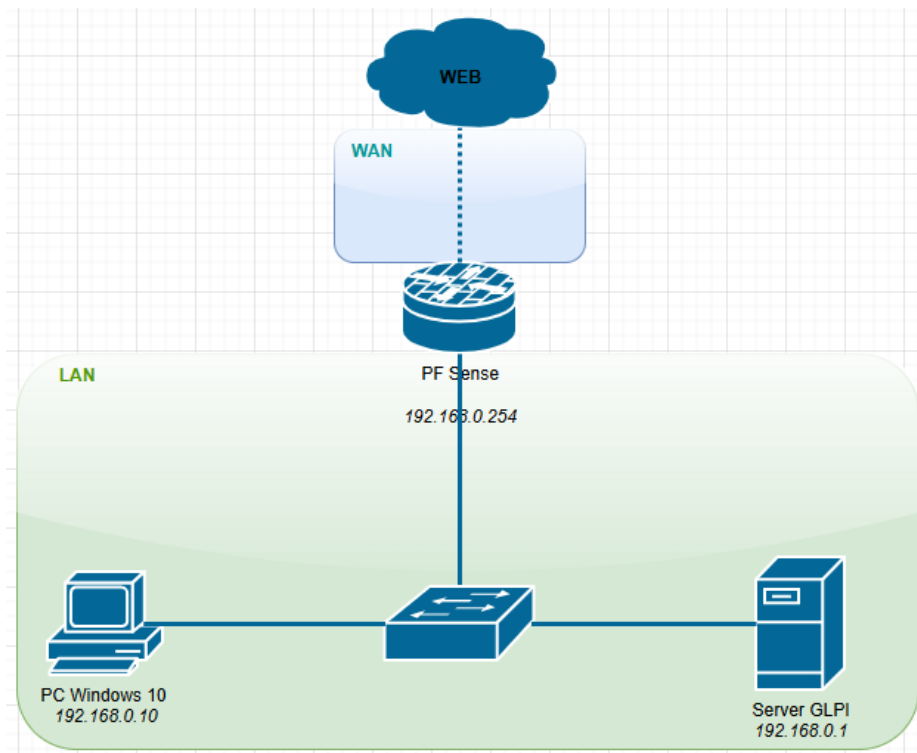
Le choix de GLPI pour ce projet s'est imposé en raison de sa gratuité et de son caractère open source, ce qui le rend parfaitement adapté à un contexte pédagogique ou à un laboratoire virtuel. GLPI offre une gestion complète du parc informatique (matériel, logiciels, licences) ainsi qu'un système de ticketing intégré pour le suivi des incidents et des demandes, le tout dans une seule interface intuitive. Comparé à Jira Service Management, qui nécessite un abonnement pour accéder à toutes les fonctionnalités, et à iTop, davantage orienté CMDB et ITSM, GLPI combine simplicité d'installation, richesse fonctionnelle et extensibilité via plugins, permettant ainsi de couvrir tous les besoins essentiels pour la gestion des services et du parc informatique dans un cadre pédagogique.

² **Service Level Agreement** ou Accord de niveau de service : C'est un contrat entre un fournisseur de services IT et un client / utilisateur, qui définit le niveau de qualité attendu pour un service. Les SLA sont essentiels pour mesurer la performance du support informatique et garantir que les utilisateurs reçoivent le service promis.

³ **Un workflow** organise et automatise le cycle de vie d'un ticket ou d'un processus, garantissant que chaque étape est suivie et que rien n'est oublié. (ex : *Ouvert* → *En cours* → *En attente* → *Résolu* → *Fermé*)

4- Cartographie du réseau

a) Schéma



b) Plan d'adressage

Réseau LAN	192.168.0.0/24
Masque	255.255.255.0
1 ^{ère} adresse	172.28.0.1/24
Dernière adresse	172.28.0.254/24
Broadcast	172.28.0.255/24

c) Tables de routage

Sous-réseau	Adresse	Interface	Passerelle
LAN	192.168.0.0/24	172.28.0.254/24	172.28.0.254/24
WAN	0.0.0.0/0	WAN	FAI

5- Etude de l'impact sur le SI existant

Ce projet est une infrastructure test réalisé sur un réseau virtualisé entre des VM. Il n'y a donc aucun impact sur le SI existant. L'objectif de ses tests est justement de permettre éventuellement un déploiement dans un environnement réel.

6- Phasage de l'intervention

- 1) Création des machines virtuelles : PF Sense ; Debian 13 ; Windows 10
- 2) Configuration réseau des machines : connectivité, adresses IP, masque, passerelle
- 3) Installation de la pile LAMP et configuration
- 4) Installation et configuration de GLPI
- 5) Génération de clés SSL pour une connexion sécurisée HTTPS

- 6) Installation de l'agent GLPI sur un client
- 7) Test de l'infrastructure

7- Prévision des tests de validation

- Essaie d'accès à la page GLPI via le navigateur web du client
- Test de connexion à GLPI
- Test d'accès en HTTPS
- Test de l'agent GLPI et de la remontée des informations sur GLPI

IV/ Mise en place

1- Réalisation et rapport de tests

a) Installation et configuration de PF Sense

- [Procédure installation et configuration de PFSense](#)

b) Installation et configuration des machines virtuelles

- [Procédure d'installation d'une machine Debian](#)
- [Procédure d'installation d'une machine Windows 10](#) ou [Procédure d'installation d'une machine Windows 11](#)

c) Installation et configuration de GLPI sur le serveur

- [Procédure d'installation et configuration de GLPI](#)

d) Sécurisation de l'accès en HTTPS

- [Procédure sécurisation HTTP,PS](#)

2- Rapport de déploiement

Le déploiement de la solution GLPI a été réalisé dans un environnement virtualisé afin de simuler l'infrastructure réseau d'une entreprise. Le matériel physique utilisé est un poste hôte unique, équipé de l'hyper-viseur VMWare sur lequel ont été créées trois machines virtuelles :

- **VM Debian 13** : serveur dédié à l'installation du LAMP (Linux, Apache, MariaDB, PHP) et de GLPI.
- **VM pfSense** : utilisée comme pare-feu et passerelle réseau, simulant l'accès WAN et assurant la sécurité.
- **VM Windows 10** : poste client destiné aux tests d'accès web et à l'installation de l'agent GLPI.

L'ensemble du déploiement s'est déroulé en quatre grandes étapes :

- 1- **Préparation de l'environnement** : Les ISO de Debian 13, Windows 10 et pfSense ont été téléchargés et utilisés pour créer les machines virtuelles. Un plan d'adressage IP a été défini au préalable afin de garantir la communication entre les VM et avec le réseau WAN simulé par pfSense.
- 2- **Installation du serveur GLPI** : Après l'installation de Debian, la pile LAMP a été configurée pour héberger l'application GLPI. L'archive de la dernière version stable de GLPI a été récupérée depuis GitHub, puis décompressée dans le répertoire /var/www/html/. Les droits d'accès aux répertoires ont été ajustés pour permettre au serveur Apache de servir correctement les fichiers GLPI. La base de données a été créée sous MariaDB et configurée via l'assistant web de GLPI.

- 3- **Configuration réseau et sécurité** : Le pare-feu pfSense a été configuré en tant que passerelle entre le réseau local (VM Debian et VM Windows) et le WAN. Les règles NAT et firewall ont été définies pour autoriser l'accès HTTP/HTTPS au serveur GLPI et les communications des agents clients.
- 4- **Validation et tests** : Depuis la VM Windows, l'accès à l'interface web de GLPI a été testé avec succès via l'URL interne du serveur. L'installation de l'agent GLPI sur le poste client a permis de vérifier la remontée automatique des informations matérielles et logicielles dans l'inventaire. Les tests de création de tickets et de résolution ont été concluants.

Le déploiement s'est déroulé sans incident majeur. L'ensemble des configurations a été documenté (adresses IP, règles pfSense, étapes d'installation) afin de faciliter la maintenance et d'assurer la reproductibilité du projet.

V/ Bilan

1- Conclusion

Le déploiement de la solution GLPI sur une infrastructure virtualisée s'est révélé concluant et a permis de valider la faisabilité d'une gestion complète du parc informatique et du support utilisateur dans un environnement pédagogique. L'utilisation de Debian 13 comme serveur, associée à pfSense pour la sécurisation et au poste client Windows 10 pour les tests, a permis de reproduire les conditions d'un réseau d'entreprise tout en restant sur un unique poste physique.

Cette mise en œuvre a permis d'acquérir et de consolider des compétences pratiques : installation et configuration d'un serveur LAMP, mise en place d'une application web d'entreprise, gestion d'une base de données MariaDB, configuration réseau et pare-feu, ainsi que déploiement et test de l'agent GLPI.

Grâce à ses fonctionnalités complètes de ticketing, gestion de parc et inventaire, GLPI s'est imposé comme l'outil le plus adapté à nos besoins dans un contexte open source. L'ajout de la couche HTTPS a renforcé la sécurité des échanges et mis en évidence l'importance de la protection des données dans un système d'information.

En conclusion, ce projet a permis non seulement de répondre aux objectifs fixés — déploiement d'un serveur de gestion d'inventaire et de tickets — mais aussi d'acquérir une méthodologie applicable à des environnements professionnels réels, en particulier pour la gestion et la maintenance des systèmes d'information.

2- Auto-évaluation

Ce projet m'a permis de mettre en pratique des compétences techniques variées : installation et configuration d'un serveur sous Debian, déploiement d'une pile LAMP, mise en place d'une application web professionnelle (GLPI), configuration réseau incluant un pare-feu pfSense, ainsi que gestion de la sécurité avec HTTPS.

J'ai rencontré quelques difficultés, notamment dans la gestion des droits utilisateurs pour les accès au dossier GLPI (www-data) et la résolution des problèmes liés à la sécurité. Problèmes résolus grâce à mes recherches, mes tests et ma persévérance.

Je considère avoir atteint la plupart des objectifs pédagogiques : j'ai appris à identifier et corriger des erreurs de configuration et renforcé ma capacité à travailler de manière autonome et méthodique. Cependant, je dois encore améliorer ma rapidité d'exécution et approfondir mes connaissances sur l'optimisation de la sécurité et la gestion avancée de la base de données MariaDB.

Globalement, ce projet a renforcé ma confiance en mes capacités techniques et m'a donné une meilleure vision des enjeux de la gestion d'un système d'information complet.