

**ASSURMER AP n°5**

PRESENTATION DE GLPI DANS UN ENVIRONEMENT DOCKER

Réalisation AP n°5 - 16/05/2024 v1.0 - AUTEURS :

RODRIGUES Antoine, BOUSSAHA Elijah, Aymeric PERRET DU CRAY

Validateur : Nassim MATOUK (maitre d’apprentissage Antoine)

**TABLES DES MATIERES**

[Le contexte en résumé : 2](#_Toc1066271575)

 [Pourquoi un environement Docker ? 3](#_Toc2054145367)

 [Pourquoi GLPI ? 5](#_Toc390283609)

 [x Pourquoi avoir choisi comme solution GLPI dans un environement Docker ? 6](#_Toc968446963)

[Pourquoi utilisons-nous Docker avec GLPI au lieu d’une installation classique ? 7](#_Toc291912993)

# **Le contexte en résumé :**

La Direction des Systèmes d'Information (DSI) d’ASSURMER souhaite désormais simplifier les fonctions de support de l'entreprise.

Avec l'approche imminente des Jeux Olympiques de 2024, il est impératif d'optimiser et de simplifier nos services au maximum, y compris le support informatique.

Actuellement, notre support est basé sur un système de courriels et de service téléphonique. La DSI prévoit de passer à un outil de gestion de tickets plus efficace et convivial, tant pour les utilisateurs que pour les techniciens du support. La solution retenue est GLPI.

GLPI ne se limite pas à la gestion de tickets ; il offre également des fonctionnalités d'inventaire et de FAQ, le rendant ainsi polyvalent.

Contrairement à l'utilisation conventionnelle de GLPI, notre déploiement reposera sur la conteneurisation avec Docker.

**Ce projet comprendra les documents suivants :**

1. Présentation de GLPI + Docker *(ce document*).

2. Topologie réseau intégrant GLPI à notre infrastructure.

3. Procédure d'installation de GLPI et Docker.

4. Procédure de configuration de GLPI.

5. Schéma du cycle de vie d’un ticket.

6. Les Différents Tests.

7. Procédure de création de ticket pour un utilisateur.

*Nous inclurons également notre planning et la répartition des tâches sur cette AP5.*

#  **Pourquoi un environement Docker ?**

De nos jours dans les systèmes d’informations d’une entreprise nous avons globalement 3 solutions pour mettre en production une application ; Proxmox, ESXI, XCP-NG (les hyperviseurs), Windows ou Linux. Mais en réalité il en éxiste une quatrième qui réunis ces trois environement. Docker peut se mettre en place aussi bien dans un environement Windows ainsi que linux et se mettre sur un hyperviseur...

De plus docker est un environement qui est axé sur la flexibilité grace à son sytème de containerisation et sa solution de déployment d’application. Dans notre cas nous souhaitons faire en sorte de répondre à plusieurs critères au niveau de la mise en place et la continuité des services de cette application et d’autres par la suite.

**1. Le déployement :** Docker utilise un principe de containerisation qui n’est pas nouveau. Proxmox ou encore ESXI avais déja essayer cette méthode via LXC ou encore CTNR mais n’ont jamais été plus loin dans cette démarche. Ce qui change avec docker c’est l’environement qui tourne autour des containers : Le déployement d’une application sur docker peux s’éfféctuer en une seule ligne de commande.

**2. La sauvegrade des Data :** En ajoutant une variable d’environement tel que “**--volume /home/glpi:/var/www/html/glpi”** sert à faire en sorte que toutes les données se trouvant dans le container soit répliqué sur notre machine qui heberge docker ce qui permet donc par la suite d’éfféctuer des sauvgarde automatisé et redondé grace à un script bash ou rSync.

3. **écologie et resources :** Docker est un environement très léger et différent d’une installation sur Windows ou Linux. Lorsque par exemple glpi est installé sur un serveur Windows en production, toutes une surcouche de services et d’applications par défault de Windows tourne en tache de fond et ce qui aura donc pour conséquences de consommer environs 5 à 6gb de Ram et 100% de 2 coeurs de procésseurs (idem pour Linux). La différence avec un container docker c’est qu’il n’y à pas OS, seulement un “kernel” en bash ce qui donne un environement de base très très léger et qui ne consome casiment rien (~180mb de Ram et 10% d’un seul coeur de procésseur.) Capture d’écran:

**Docker avec GLPI en production :**



De plus, Docker simplifie la création d'environnements de développement uniformes pour tous les membres de l'équipe IT. Finies les situations où "ça marche que sur ma machine" ! Avec Docker, chaque développeur / administrateur travaille dans un environnement identique à la production, réduisant ainsi les erreurs dues aux différences de configuration.

En somme, Docker offre une approche légère, portable et flexible pour empaqueter et déployer des applications dans des environnements modernes, particulièrement adaptés aux architectures micro-services et au cloud computing. Il répondra donc parfaitement à nos exigences futures et aux exigences de GLPI.

#  **Pourquoi GLPI ?**

GLPI (Gestionnaire Libre de Parc Informatique) est un logiciel open-source développé par Teclib, conçu pour la gestion des services et des actifs informatiques. Il permet aux entreprises de gérer efficacement leur parc informatique, de suivre les incidents et les demandes de service, et de maintenir un inventaire précis de leurs matériels informatique.

Les principales fonctionnalités de GLPI que nous utiliserons sont :

1. **Gestion des actifs** : GLPI permet de gérer tous types d'actifs informatiques (ordinateurs, serveurs, imprimantes, logiciels, licences, etc.). Il enregistre des informations détaillées sur chaque actif, telles que son modèle, son numéro de série, sa localisation et son statut.
2. **Gestion des tickets** : GLPI offre un système complet de gestion des tickets pour suivre et résoudre les incidents et les demandes de service. Les utilisateurs peuvent soumettre des tickets via une interface web, par email ou via des APIs. Les techniciens peuvent ensuite assigner, prioriser, suivre et résoudre ces tickets.
3. **Rapports et tableaux de bord** : GLPI génère des rapports et des tableaux de bord fournissant une vue d'ensemble de l'état du parc informatique, des performances de l'équipe de support, des coûts associés, etc.

De nombreuses autres fonctionnalités, telles que la gestion des contrats ou des licences et la base de connaissances, sont également disponibles, mais nous ne les déploierons pas pour le moment.

L'un des grands avantages de GLPI est sa flexibilité. Il peut être adapté aux besoins spécifiques de chaque entreprise grâce à son système de plugins. De nombreux plugins officiels et communautaires étendent les fonctionnalités de GLPI, incluant la gestion de projets, la réservation de ressources, l'intégration avec d'autres outils, etc.

En résumé, GLPI est une solution complète et flexible pour gérer les services et les actifs informatiques. Il aide les entreprises à rationaliser leurs opérations IT, à améliorer la qualité de service et à réduire les coûts. Sa nature open-source et sa large communauté active en font un choix populaire pour de nombreuses organisations.

#

# **x**  **Pourquoi avoir choisi comme solution GLPI dans un environement Docker ?**

### Pourquoi utilisons-nous Docker avec GLPI au lieu d’une installation classique ?

Déployer GLPI dans un conteneur Docker offre plusieurs avantages par rapport à un déploiement traditionnel sur un serveur Debian. Voici quelques raisons pour lesquelles Docker peut être préférable :

1. **Isolation et sécurité** : Chaque conteneur Docker est isolé et contient uniquement les dépendances nécessaires pour l'application. Cela réduit les risques de conflits avec d'autres applications ou services sur le même serveur. De plus, si un conteneur est compromis, cela n'affecte pas nécessairement les autres conteneurs ou le système hôte.
2. **Portabilité et cohérence** : Un conteneur Docker encapsule GLPI et toutes ses dépendances dans un package unique. Ce conteneur peut être exécuté de manière identique sur n'importe quel système supportant Docker, que ce soit en développement, en test ou en production. Cela élimine les problèmes liés aux différences d'environnement.
3. **Facilité de déploiement** : Avec Docker, déployer une nouvelle instance de GLPI est aussi simple que de lancer un nouveau conteneur. Il n'est plus nécessaire de configurer un serveur entier ou d'installer manuellement les dépendances, ce qui accélère et simplifie grandement le processus de déploiement.
4. **Gestion des ressources** : Docker permet de faire fonctionner plusieurs conteneurs sur le même serveur, tout en ayant une charge légère sur les performances du serveur. Cela permet de faire fonctionner de nombreux services sur un seul serveur à performances modestes.
5. **Maintenance et mises à jour** : Mettre à jour GLPI dans un conteneur Docker est généralement plus simple et plus sûr. Il suffit de construire une nouvelle image avec la nouvelle version et de remplacer les conteneurs existants. En cas de problème, il est facile de revenir à la version précédente.

Dans l'ensemble, Docker apporte de nombreux avantages en termes de portabilité, de sécurité et de facilité de gestion, ce qui en fait souvent le meilleur choix pour déployer des applications comme GLPI dans des environnements modernes.