

Mise en place de l'outil de Ticketing GLPI

Configuration initiale Docker et GLPI

ASSURMER

Montpellier, Occitanie, France

Maxence Martin-Parent

Ezequiel VARELA-MONTEIRO

Kévin BOULIER

SISR 1B



Version	Date version	Auteur	Validateur et date	Destinataires	Diffusion document	Nbr. de pages	Commentaires
1	20/05/24	Maxence MARTIN-PARENT	Aucun	Service DSI	Interne via Teams	13	Document entier

Table des matières

Introduction.....	3
Installation de Docker	4
Installation de Portainer	6
Installation de MariaDB, phpMyAdmin, GLPI.....	8
Configuration préliminaire de GLPI.....	10

Introduction

Prérequis nécessaires à l'installation de GLPI sur Docker.

Pour mettre en place GLPI, nous avons demandé au service infrastructure de nous fournir une machine virtuelle sur l'hyperviseur **Proxmox** d'ASSURMER.

La machine sera l'hôte de **Docker**. Docker ne nécessitant pas de grandes performances, nous avons demandé une machine aux performances légères.

Ainsi, nous possédons la machine suivante, nommée ASSURDOCKER :

- 2 vCPU
- 4 Go de RAM
- 32 Go de stockage
- Système d'exploitation Debian 12 Bullseye
- Connexion réseau au réseau interne seulement, étant une machine de dev. La machine de production sera située dans la DMZ pour permettre un accès externe.

La connexion au réseau interne est obligatoire également pour effectuer la jonction à l'Active Directory et permettre l'accès à GLPI aux collaborateurs par leur compte personnel AD.

Installation de Docker

Logiciel permettant l'exécution de containers.

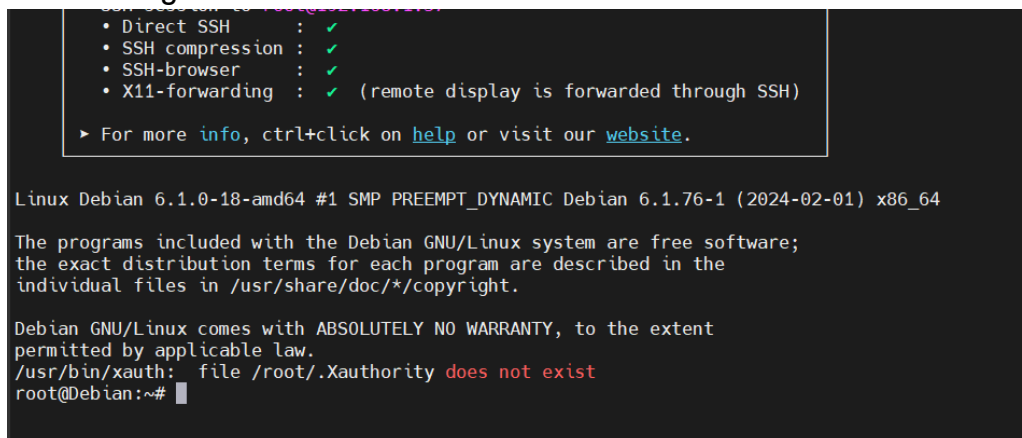
On accède à la machine par le centre d'administration Proxmox de développement, dont l'IP est 192.168.100.10.

On se connecte, la machine est déjà prête sous l'ID 602, avec l'IP 192.168.100.6.

1. Appuyer sur « Démarrer »



2. La machine dispose du SSH incorporé. On va donc se connecter au serveur en SSH via le logiciel MobaXTerm.



3. Il va falloir ensuite incorporer ses commandes pour pouvoir se connecter aux serveurs possédant les packages de Docker :

- a. `apt-get update` (MàJ d'apt-get)

```
root@Debian:~# apt-get update
```

- b. `apt-get install apt-transport-https ca-certificates curl software-properties-common` (installe des paquets obligatoires pour installer docker)

```
root@Debian:~# apt-get install apt-transport-https ca-certificates curl software-properties-common
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
ca-certificates est déjà la version la plus récente (20230311).
software-properties-common est déjà la version la plus récente (0.99.30-4).
software-properties-common passé en « installé manuellement ».
Les NOUVEAUX paquets suivants seront installés :
  apt-transport-https curl
```

- c. `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -` (permet de vérifier que les paquets téléchargés sont bien les officiels Docker)

```
root@Debian:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

- d. `add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"` (Ajout du dépôt Docker dans apt-get)

```
root@Debian:~# add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/debian bookworm stable'
Description:
Archive for codename: bookworm components: stable
More info: https://download.docker.com/linux/debian
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
```

- e. `apt-get update` (MàJ d'apt-get)

```
root@Debian:~# apt-get update
```

- f. `apt-get install docker-ce` (installation de Docker)

```
root@Debian:~# apt-get install docker-ce
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man iptables liberror-perl libip6tc2
  libslirp0 patch pigz slirp4netns
Paquets suggérés :
```

- g. `systemctl status docker` (Vérifie que Docker s'est bien installé)

```
root@Debian:~# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-05-02 19:07:33 CEST; 6s ago
 TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 9405 (dockerd)
     Tasks: 8
    Memory: 33.4M
       CPU: 129ms
    CGroup: /system.slice/docker.service
            └─9405 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Docker est bien installé. On passe maintenant à Portainer.

Installation de Portainer

Conteneur permettant lui-même de gérer les autres conteneurs

Avant de commencer, il convient de présenter Portainer.

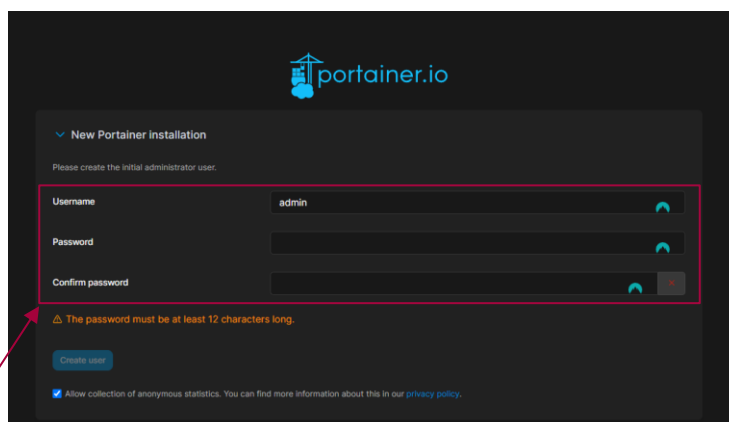
Portainer est lui même un container, permettant de gérer les container que l'on a installé sur une machine avec Docker. Cela permet une gestion plus simple, à l'aide d'une interface utilisateur accessible sur un navigateur web.

1. En SSH, lancer cette commande : `docker run -d -p 9000:9000 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest`

Cela va nous permettre d'installer le container "portainer" sur le port 9000.

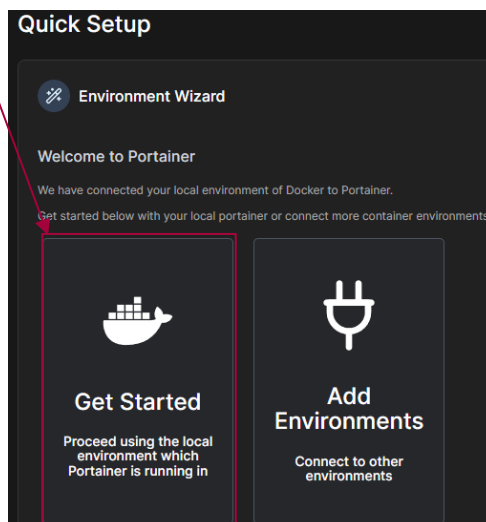
```
root@debian:~# docker run -d -p 9000:9000 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
root@debian:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED       STATUS       PORTS
11479acfab99   portainer/portainer-ce:latest       -                       10 minutes ago Up 10 minutes 9000->9000
root@debian:~#
```

2. On peut maintenant accéder à l'interface Web de Portainer, à l'adresse **192.168.100.6:9000**

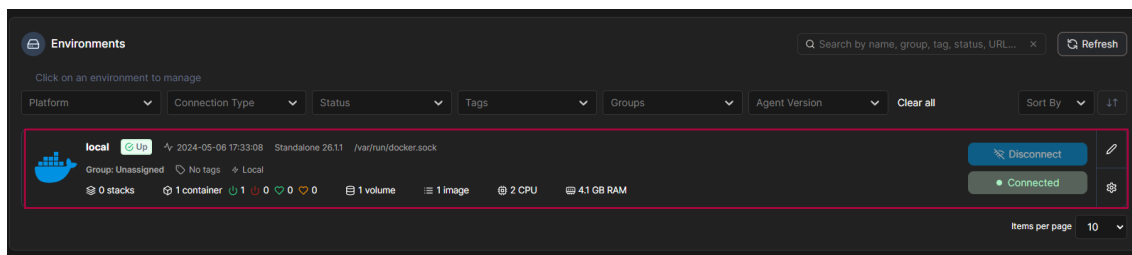


On créé les **identifiants**, et on continue.

3. Cliquer sur « **Get Started** »



Portainer a ainsi reconnu l'environnement Docker que nous avons installé plus tôt. Nous allons pouvoir passer à la suite.



Installation de MariaDB, phpMyAdmin, GLPI

Installation de toutes les conteneurs nécessaires pour GLPI

GLPI est un logiciel open-source. Aucune solution officielle n'est donnée pour utiliser GLPI pour Docker, mais la communauté très active de GLPI s'en est déjà chargé pour nous.

Nous allons utiliser le fork (version dérivée de l'initial) de GLPI créé par diouxx (<https://github.com/Diouxx/docker-glpi>). Ce fork est très renommé et régulièrement mis à jour. Il est lui même open-source (pour être en accord avec la licence de GLPI), et il est donc possible de vérifier très rapidement si besoin sa fiabilité.

Nous allons nous baser sur l'installation avec une base de donnée (MariaDB) et données persistantes, vouée à l'utilisation en environnement de production.

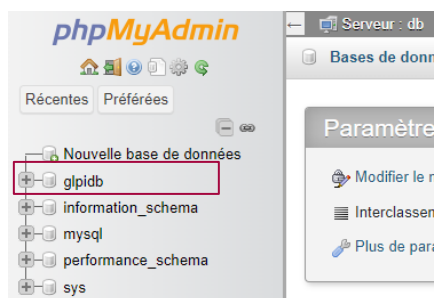
Nous allons également rajouter un container PhPMyAdmin, nous permettant de gérer les bases de données plus efficacement. Ce n'est pas nécessaire, mais un ajout bienvenu.

1. On commence par créer la base de données MariaDB avec la commande : `docker run --name mariadb -e MARIADB_ROOT_PASSWORD=<mot_de_passe_root> -e MARIADB_DATABASE=glpidb -e MARIADB_USER=glpi_user -e MARIADB_PASSWORD=<mot_de_passe_user> --volume /var/lib/mysql:/var/lib/mysql -d mariadb:10.7`

```
root@debian:~# docker run --name mariadb -e MARIADB_ROOT_PASSWORD= -e MARIADB_DATABASE=glpidb -e MARIADB_USER=glpi_user -e MARIADB_PASSWORD= --volume /var/lib/mysql:/var/lib/mysql -d mariadb:10.7
```

2. On installe ensuite PhPMyAdmin.

```
docker run --name phpmyadmin --link mariadb:db -p 8080:80 -e PMA_HOST=db -d phpmyadmin/phpmyadmin
```



La base de données précédemment créée est bien visible.

3. On installe maintenant GLPI.

```
docker run --name glpi --link mariadb:mariadb --volume /var/www/html/glpi:/var/www/html/glpi -p 80:80 -d diouxx/glpi
```

On le met en HTTP seulement car nous ne quittons pas l'environnement de développement. La version finale devra être effectuée en HTTPS avec un certificat approprié. Il faudra également le lier au reverse proxy qui permettra de lier (<http://support.assurmer-assurances.fr>) au serveur Docker.

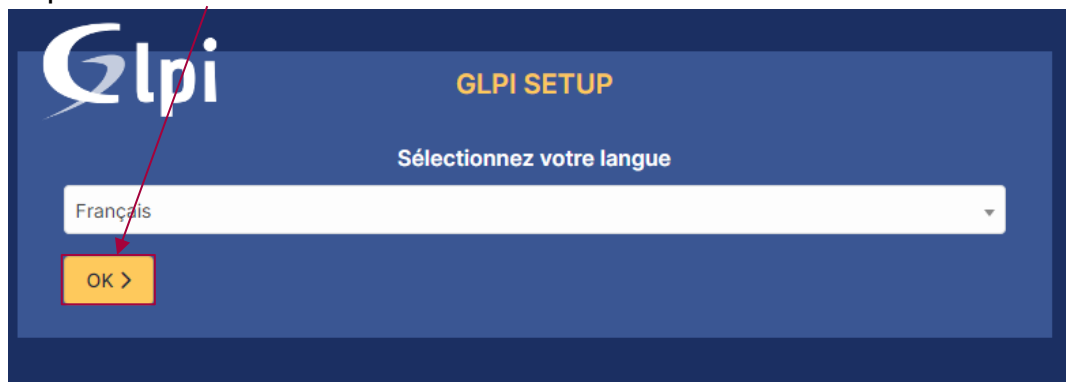
On peut vérifier sur le port 80 que GLPI est **fonctionnel** :



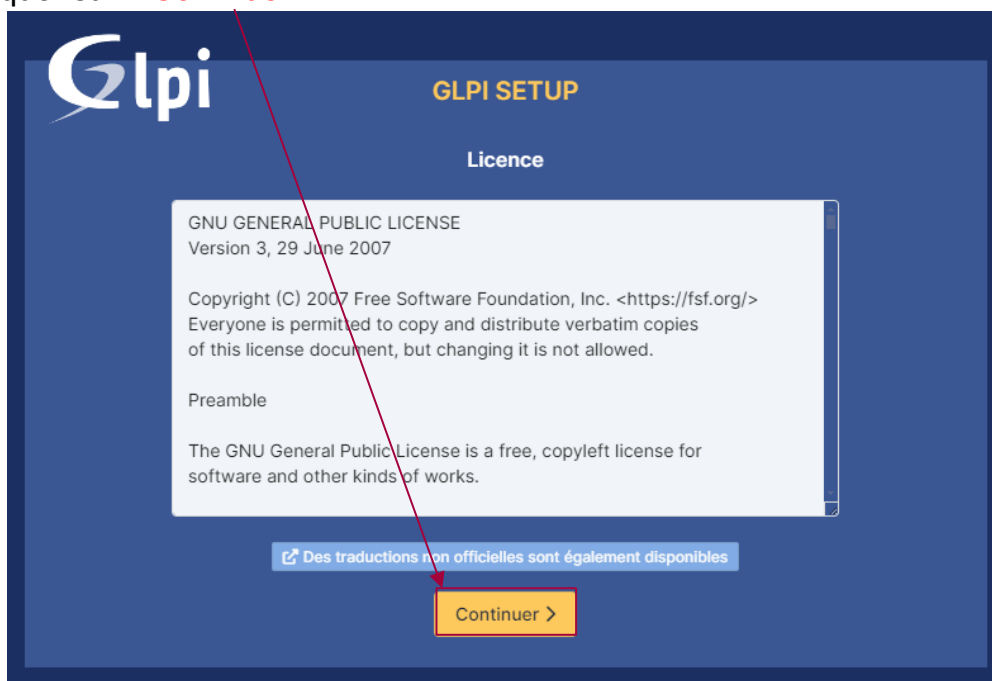
Configuration préliminaire de GLPI

Premières configurations

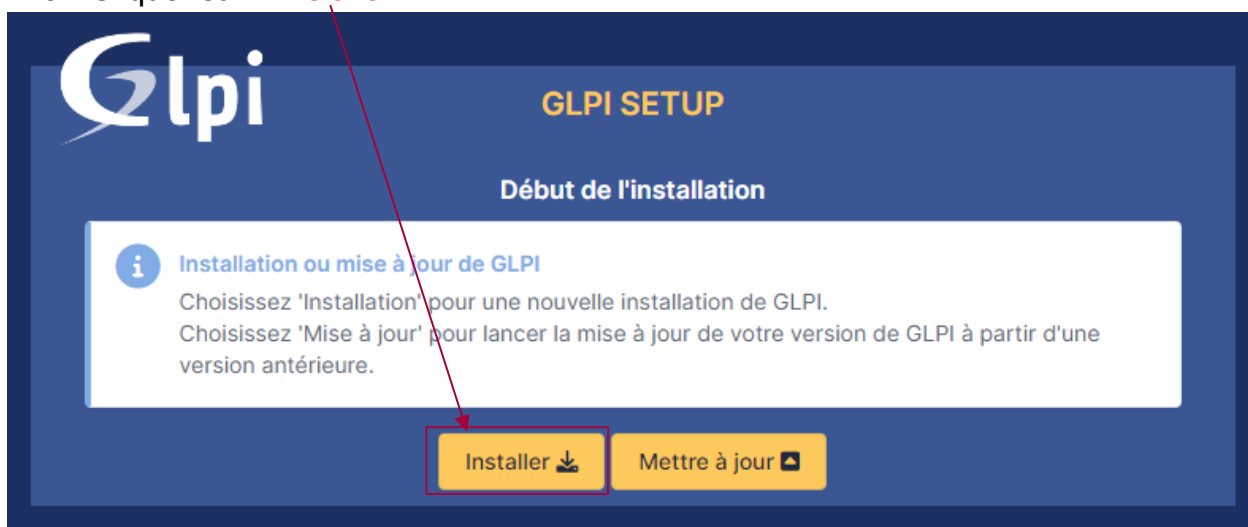
1. Cliquer sur « **OK** »



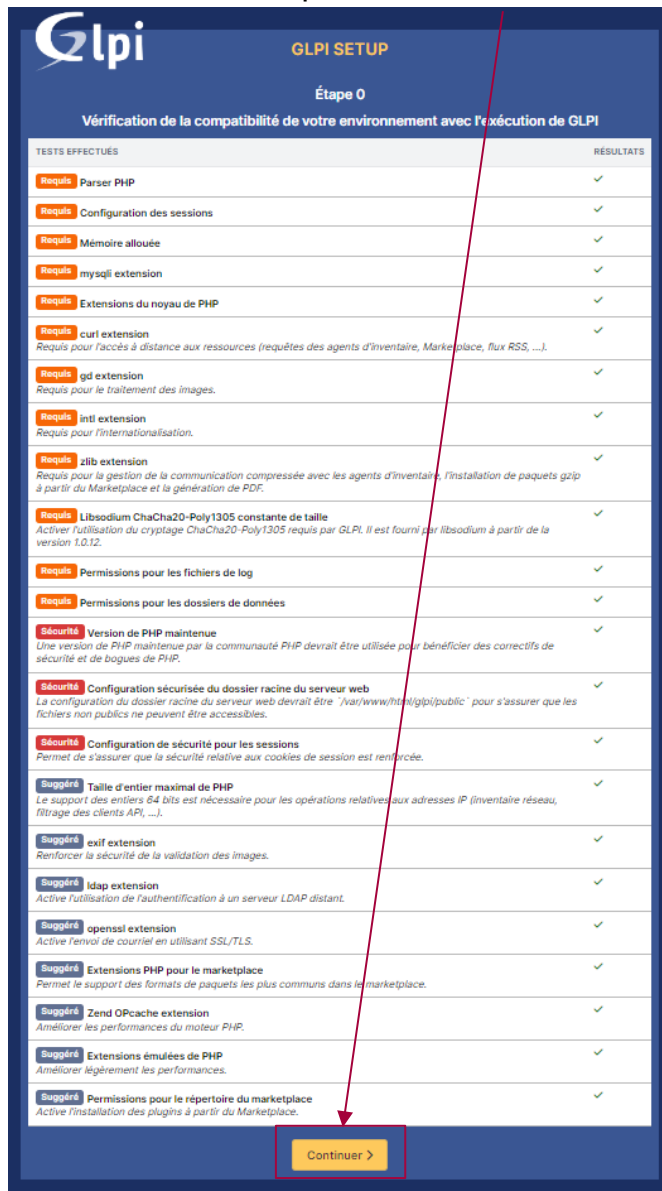
2. Cliquer sur « **Continuer** »



3. Cliquer sur « **Installer** »



4. Vérifier que tout est bien coché. Cliquer sur « **Continuer** ».



GLPI **GLPI SETUP**

Étape 0

Vérification de la compatibilité de votre environnement avec l'exécution de GLPI

TESTS EFFECTUÉS	RÉSULTATS
Requis Parser PHP	✓
Requis Configuration des sessions	✓
Requis Mémoire allouée	✓
Requis mysql extension	✓
Requis Extensions du noyau de PHP	✓
Requis curl extension <small>Requis pour l'accès à distance aux ressources (requêtes des agents d'inventaire, Marketplace, flux RSS, ...).</small>	✓
Requis gd extension <small>Requis pour le traitement des images.</small>	✓
Requis intl extension <small>Requis pour l'internationalisation.</small>	✓
Requis zlib extension <small>Requis pour la gestion de la communication compressée avec les agents d'inventaire, l'installation de paquets glpi à partir du Marketplace et la génération de PDF.</small>	✓
Requis Libsodium ChaCha20-Poly1305 constante de taille <small>Activer l'utilisation du cryptage ChaCha20-Poly1305 requis par GLPI. Il est fourni par Libsodium à partir de la version 1.0.12.</small>	✓
Requis Permissions pour les fichiers de log	✓
Requis Permissions pour les dossiers de données	✓
Sécurité Version de PHP maintenue <small>Une version de PHP maintenue par la communauté PHP devrait être utilisée pour bénéficier des correctifs de sécurité et de bogues de PHP.</small>	✓
Sécurité Configuration sécurisée du dossier racine du serveur web <small>La configuration du dossier racine du serveur web devrait être "/var/www/html/glpi/public" pour s'assurer que les fichiers non publics ne peuvent être accessibles.</small>	✓
Sécurité Configuration de sécurité pour les sessions <small>Permet de s'assurer que la sécurité relative aux cookies de session est renforcée.</small>	✓
Suggéré Taille d'entier maximal de PHP <small>Le support des entiers 64 bits est nécessaire pour les opérations relatives aux adresses IP (inventaire réseau, filtrage des clients API, ...).</small>	✓
Suggéré exif extension <small>Renforcer la sécurité de la validation des images.</small>	✓
Suggéré ldap extension <small>Active l'utilisation de l'authentification à un serveur LDAP distant.</small>	✓
Suggéré openssl extension <small>Active l'envoi de courriel en utilisant SSL/TLS.</small>	✓
Suggéré Extensions PHP pour le marketplace <small>Permet le support des formats de paquets les plus communs dans le marketplace.</small>	✓
Suggéré Zend OPcache extension <small>Améliorer les performances du moteur PHP.</small>	✓
Suggéré Extensions émulées de PHP <small>Améliorer légèrement les performances.</small>	✓
Suggéré Permissions pour le répertoire du marketplace <small>Active l'installation des plugins à partir du Marketplace.</small>	✓

Continuer >

5. Entrer les informations de connexion à la base de données. Cliquer sur « **Continuer** ». (L'IP de la base de données est celle interne à Docker, visible dans l'interface de Portainer)



GLPI **GLPI SETUP**

Étape 1

Configuration de la connexion à la base de données

Serveur SQL (MariaDB ou MySQL)

172.17.0.3

Utilisateur SQL


glpi_user

Mot de passe SQL

....

Continuer >

6. Sélectionner « **glpidb** », puis « **Continuer** ».



7. Cliquer sur « **Continuer** ».




8. **Décocher**, puis cliquer sur « **Continuer** ».



9. Cliquer sur « **Continuer** », puis sur « **Utiliser GLPI** ».



10. GLPI est maintenant **fonctionnel**. Il faut maintenant passer à la configuration de GLPI pour qu'il réponde à nos besoins.



Connexion à votre compte

Identifiant

Mot de passe

Source de connexion

Base interne GLPI

☒ Se souvenir de moi

Se connecter

GLPI Copyright (C) 2015-2024 Teclib' and contributors