

GLPI avec Docker — Installation simple et expliquée

Un mini-déploiement réel : GLPI + MySQL + accès Web

Le but du TP

Installer GLPI rapidement dans une VM Debian avec Docker, sans installer manuellement Apache, PHP, MySQL et GLPI sur le système. Docker Compose va créer les conteneurs nécessaires et les faire travailler ensemble.

1) Qu'est-ce qu'on crée exactement ?

Question — GLPI tient dans un seul conteneur ?

Non, dans l'exemple officiel utilisé ici, Docker Compose lance deux services principaux : un conteneur GLPI et un conteneur MySQL.

- Conteneur GLPI = l'application Web que tu ouvres dans le navigateur.
- Conteneur MySQL = la base de données qui stocke les tickets, utilisateurs, matériels, paramètres, etc.
- GLPI communique avec MySQL sur le réseau interne créé par Docker Compose.

Architecture très simple

PC Windows / navigateur
↓
http://192.168.190.134:80
↓
VM Debian + Docker
↓



Question — Pourquoi plusieurs conteneurs ?

Parce que GLPI et la base de données sont deux rôles différents. Les séparer permet de gérer chaque service indépendamment, tout en les reliant avec Docker Compose.

2) À quoi servent les deux fichiers téléchargés ?

Question — docker-compose.yml, c'est quoi ?

C'est la recette de l'application multi-conteneurs. Elle indique à Docker Compose quelles images utiliser, quels volumes créer, quels services démarrer et quels ports publier.

Question — .env, c'est quoi ?

C'est un petit fichier de variables. Dans l'exemple officiel, il contient notamment l'adresse du service de base de données, le nom de la base, l'utilisateur MySQL et son mot de passe.

Très important — ne pas confondre les identifiants

Les identifiants du fichier .env servent à la connexion entre GLPI et MySQL. Ils ne sont pas les identifiants de connexion à l'interface Web GLPI.

3) Installation complète — dans l'ordre

Étape	Commande / action	Pourquoi ?
1	<code>mkdir -p /root/glpi</code>	Créer le dossier du projet.
2	<code>cd /root/glpi</code>	Se placer dans ce dossier.
3	<code>curl ...docker-compose.example.y ml --output docker-compose.yml</code>	Télécharger la recette Docker Compose officielle.
4	<code>curlenv.example --output .env</code>	Télécharger les variables de configuration de l'exemple.
5	<code>docker compose up -d</code>	Télécharger les images si besoin, créer les conteneurs et les démarrer en arrière-plan.
6	<code>docker compose ps</code>	Vérifier que les services sont démarrés.
7	<code>ip -br a</code>	Trouver l'adresse IP de la VM Debian.
8	<code>http://IP_VM:80</code>	Ouvrir GLPI depuis le navigateur.

4) Les commandes exactes utilisées

```
mkdir -p /root/glpi
cd /root/glpi

curl --fail
https://raw.githubusercontent.com/glpi-project/docker-images/main/docker-compose.example.yml --output docker-compose.yml

curl --fail
https://raw.githubusercontent.com/glpi-project/docker-images/main/.env.example --output .env

docker compose up -d
docker compose ps
ip -br a
```

Dans ton installation

L'adresse utilisée depuis ton navigateur est : <http://192.168.190.134:80>
Le « :80 » correspond au port HTTP publié par Docker pour GLPI.

5) Que fait réellement docker compose up -d ?

Question — Pourquoi cette seule commande fait autant de choses ?

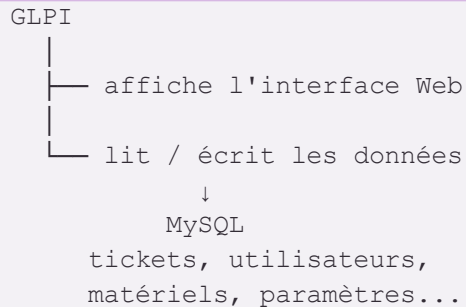
Parce que Docker Compose lit le fichier docker-compose.yml et orchestre l'ensemble pour toi.

- Il récupère l'image officielle GLPI si elle n'est pas déjà présente.
- Il récupère l'image MySQL si nécessaire.
- Il crée le conteneur GLPI.
- Il crée le conteneur MySQL.
- Il crée le réseau permettant aux deux conteneurs de communiquer.
- Il crée les volumes nommés pour conserver les données.
- Il publie le port 80 de GLPI sur le port 80 de la VM.
- L'option -d signifie : démarrer en arrière-plan.

6) Où est la base de données ?

Question — Je ne vois pas MySQL dans mon navigateur. Est-ce normal ?

Oui. MySQL n'est pas une interface Web à ouvrir dans le navigateur. C'est le moteur de base de données utilisé en arrière-plan par GLPI.



Question — Comment vérifier que les deux conteneurs existent ?

Utiliser docker compose ps depuis le dossier /root/glpi.

```
cd /root/glpi
docker compose ps
```

7) Connexion à GLPI

Question — Quels identifiants utiliser sur la page de connexion ?

Pour une installation GLPI fraîche, le compte administrateur par défaut documenté est : glpi / glpi.

- Identifiant : glpi
- Mot de passe : glpi
- Après connexion, change le mot de passe par défaut.

Attention

Le compte Web glpi/glpi et l'utilisateur MySQL glpi du fichier .env peuvent avoir le même texte dans cet exemple, mais ils n'ont pas le même rôle. L'un sert à se connecter à GLPI dans le navigateur ; l'autre sert à la connexion de GLPI à la base MySQL.

8) Les commandes de gestion à retenir

Étape	Commande / action	Pourquoi ?
Voir	<code>docker compose ps</code>	Voir l'état des services du projet.
Logs	<code>docker compose logs glpi</code>	Voir les journaux du conteneur GLPI.
Arrêter	<code>docker compose stop</code>	Arrêter les conteneurs sans les supprimer.
Redémarrer	<code>docker compose start</code>	Redémarrer les conteneurs existants.
Relancer	<code>docker compose restart</code>	Redémarrer les services.
Supprimer	<code>docker compose down</code>	Supprimer les conteneurs et le réseau du projet ; les volumes nommés restent normalement présents.

Commande dangereuse à connaître

`docker compose down -v` supprime aussi les volumes du projet. Comme la base MySQL est stockée dans un volume, cela peut supprimer les données de ton installation. Ne l'utilise pas comme commande de nettoyage réflexe.

9) Si la page Web ne s'ouvre pas

Étape	Commande / action	Pourquoi ?
1	<code>ip -br a</code>	Vérifier l'IP réelle de la VM Debian.
2	<code>docker compose ps</code>	Vérifier que GLPI et MySQL tournent.
3	<code>docker compose logs glpi</code>	Chercher une erreur côté GLPI.
4	<code>http://IP_VM:80</code>	Utiliser l'IP de la VM, pas 127.0.0.1 depuis le PC Windows.

Question — Pourquoi 127.0.0.1 ne fonctionnait pas depuis Windows ?

Parce que 127.0.0.1 désigne toujours la machine sur laquelle le navigateur s'exécute. Depuis Windows, 127.0.0.1 pointe vers Windows, pas vers la VM Debian. Il faut donc utiliser l'adresse IP de la VM Debian.

10) Ce qu'il faut réellement retenir

Résumé express

1. Docker Compose lit une recette : docker-compose.yml.
2. Cette recette crée ici deux conteneurs : GLPI + MySQL.
3. GLPI est l'application Web ; MySQL stocke ses données.
4. docker compose up -d lance toute l'installation.
5. docker compose ps permet de vérifier les conteneurs.
6. On ouvre GLPI avec `http://IP_DE_LA_VM:80`.
7. Pour une installation fraîche, le compte Web administrateur par défaut est `glpi / glpi`.

11) Pourquoi ce TP est un bon exemple de Docker ?

- Tu n'as pas installé GLPI, PHP et MySQL un par un directement sur Debian.
- Tu as utilisé des images prêtes à l'emploi.
- Docker a créé des conteneurs isolés à partir de ces images.
- Docker Compose a relié les différents services ensemble.
- Tu as publié le service Web pour y accéder depuis ton navigateur.
- Les volumes permettent de conserver les données indépendamment de la vie des conteneurs.

Sources officielles vérifiées

- GLPI Help Center — Running GLPI on Docker.
- Dépôt officiel `glpi-project/docker-images` — `docker-compose.example.yml`.
- Dépôt officiel `glpi-project/docker-images` — `.env.example`.
- Documentation d'installation GLPI — comptes créés par défaut.