Installation Docker



Cette approche Docker présente un avantage majeur par rapport à l'installation classique LAMP : pas besoin de gérer manuellement Apache, PHP-FPM et leurs extensions — tout est packagé dans l'image. C'est aussi l'approche la plus cohérente avec ton infra Rakouns, où GLPI tourne déjà sous Docker.

Prérequis

Avant de commencer, voici ce dont tu as besoin :

- **Docker Engine** et **Docker Compose** (plugin `docker compose`, pas l'ancien binaire `docker-compose` en Python)
- Un nom de domaine ou une entrée DNS pointant vers ton hôte
- Suffisamment d'espace disque pour les volumes (fichiers GLPI + base de données)

Vérifie rapidement que Docker est bien installé :

```
docker --version
docker compose version
```

Si ce n'est pas le cas : Installation de Docker

>

> **Note** : le dépôt GitHub officiel `glpi-project/docker-images` fournit des fichiers prêts à l'emploi `docker-compose.example.yml` et `.env.example`, que nous allons utiliser comme base.

Préparer l'arborescence

Comme pour tes autres stacks Rakouns, nous allons tout ranger sous `/Docker/` :

```
sudo mkdir -p /Docker/glpi
cd /Docker/glpi
```

Récupérer les fichiers officiels

Plutôt que de tout écrire à la main, le plus fiable est de récupérer directement les fichiers d'exemple maintenus par le projet 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
```

Le fichier `.env`

Ouvre le fichier `.env` :

```
nano .env
```

Il contient les variables minimales nécessaires à la connexion à la base de données :

```
GLPI_DB_HOST=db
GLPI_DB_PORT=3306
GLPI_DB_NAME=glpi
GLPI_DB_USER=glpi
GLPI_DB_PASSWORD=glpi
```

Personnalise impérativement le mot de passe (`GLPI_DB_PASSWORD`). Ici, génère un mot de passe robuste de type 40 caractères et évite les caractères `$` et `#` qui peuvent poser des soucis d'échappement dans les fichiers `.env` Docker Compose.

```
GLPI_DB_HOST=db
GLPI_DB_PORT=3306
GLPI_DB_NAME=glpi
GLPI_DB_USER=glpi_adm
GLPI_DB_PASSWORD=MotDePasseRobuste40Caracteres
```

Le fichier `docker-compose.yml`

Voici le contenu officiel récupéré, à adapter légèrement :

```
name: glpi

services:
  glpi:
    image: "glpi/glpi:latest"
    restart: "unless-stopped"
    volumes:
      # Volume nommé : évite les soucis de permissions sur l'hôte
      - glpi_data:/var/glpi
    env_file: .env
    depends_on:
      - db
    ports:
      - "80:80"

  db:
    image: "mysql"
    restart: "unless-stopped"
    volumes:
      - db_data:/var/lib/mysql
    environment:
      MYSQL_RANDOM_ROOT_PASSWORD: "yes"
      MYSQL_DATABASE: ${GLPI_DB_NAME}
      MYSQL_USER: ${GLPI_DB_USER}
      MYSQL_PASSWORD: ${GLPI_DB_PASSWORD}

volumes:
  glpi_data:
  db_data:
```

Quelques remarques importantes sur cette stack :

- **Le volume** `/var/glpi` : l'image glpi/glpi fournit par défaut un volume contenant les répertoires config, marketplace et files. C'est l'équivalent Docker des répertoires `/etc/glpi`, `/var/lib/glpi` et `/var/log/glpi` qu'on créait manuellement en installation classique — ici, tout est regroupé sous `/var/glpi` dans le conteneur.
- `MYSQL_RANDOM_ROOT_PASSWORD: "yes"` : le mot de passe root de MySQL est généré aléatoirement et stocké dans les logs du conteneur. Comme tu n'as pas besoin du compte root pour GLPI (seulement de l'utilisateur dédié), c'est volontairement opaque.
- **Le port** `80:80` : à adapter selon ton organisation. Tu pourrais vouloir mapper sur un port interne dédié (ex. `8089:80`) plutôt que d'exposer le 80 directement sur l'hôte.

Si tu prévois d'utiliser des plugins du marketplace en GLPI 10.x, il faudrait ajouter un volume supplémentaire — mais en GLPI 11, ce n'est plus nécessaire, le répertoire marketplace étant déjà inclus dans le volume principal `/var/glpi`.

Lancer la stack

Une fois les fichiers prêts, démarre les conteneurs :

```
docker compose up -d
```

Vérifie que tout tourne correctement :

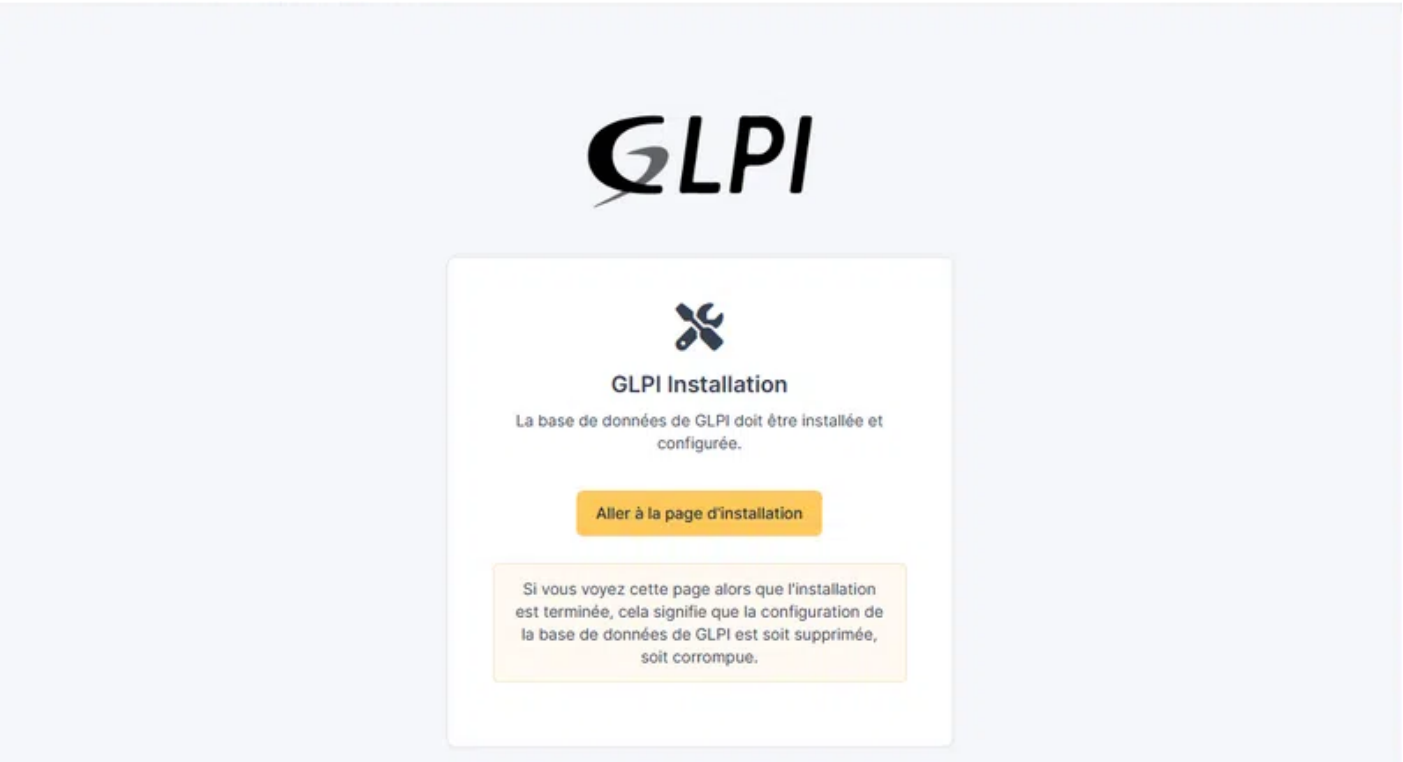
```
docker compose ps
docker compose logs -f glpi
```

Tu dois voir GLPI terminer son initialisation et le service Apache interne démarrer. La base MySQL doit également apparaître comme saine.

Installation de GLPI via le navigateur

Ouvre ton navigateur et rends-toi sur l'adresse de ton hôte Docker, sur le port que tu as mappé (par exemple `http://IP:PORT` ou via ton reverse proxy NPM si déjà configuré).

Si tout s'est bien déroulé, tu arrives directement sur l'assistant d'installation de GLPI :



Le processus d'installation est très similaire à la version classique, déjà vue ici : Installation Classique.

Revision #1
Created 2026-06-20 12:52:19 UTC by clement-derouet
Updated 2026-06-20 13:03:59 UTC by clement-derouet