

GESTIONNAIRE LIBRE DE PARC INFORMATIQUE

DOCKER-GLPI

INSTALLATION AVEC DOCKER

J'ai choisit d'utiliser docker pour l'installation de GLPI, pour découvrir et se challenger.

Docker est un outil qui peut empaqueter une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur

- On commence par installer Docker et Docker compose. Je suis sur Mac OS mais pour linux cette commande permet l'installation :

```
sudo curl -L  
"https://github.com/docker/compose/releases/download/1.23.2/d  
ocker-compose-$(uname -s)-$(uname -m)" -o /usr/bin/docker-  
compose && sudo chmod +x /usr/bin/docker-compose
```

Docker compose s'utilise en CLI (Command Line Interface)

- Pour vérifier l'installation de docker compose il faut faire :

docker-compose -v

On visualise la version de docker-compose installer sur notre machine.

- On crée un dossier « docker-glpi »
- On crée un fichiers .yml:

docker-compose :

```
version: "3.2"

services:
  mariadb:
    image: mariadb:10.7
    container_name: mariadb
    hostname: mariadb
    volumes:
      - /var/lib/mysql:/var/lib/mysql
    environment:
      - MARIADB_ROOT_PASSWORD=pass
      - MARIADB_DATABASE=glpidb
      - MARIADB_USER=glpi_user
      - MARIADB_PASSWORD=glpi
    restart: always

  glpi:
    image: diouxx/glpi
    container_name: glpi
    hostname: glpi
    ports:
      - "80:80"
    volumes:
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
      - /var/www/html/glpi:/var/www/html/glpi
    environment:
      - TIMEZONE=France/Paris
```

Dans ce fichier on retrouve :

- La version utilisé par docker compose.
- Les services servent à définir nos conteneurs, ici nous avons 2 conteneurs : mariadb et glpi.
- Image est l'image docker que nous souhaitons utiliser. Elle est composée de plusieurs couches empaquetant toutes les installations, dépendances, bibliothèques, processus et codes d'application nécessaires pour un environnement de conteneur pleinement opérationnel.
- L'argument volume nous permet de stocker l'ensemble du contenu du dossier dans un disque persistant et donc de garder les données en local.
- L'argument restart définit la politique du redémarrage du conteneur dans notre cas il redémarrera automatiquement.
- On utilise des variables d'environnement associé à notre image. Je donne au conteneur les valeurs des différents mots de passe et utilisateurs qui doivent exister sur cette base.
- Le port , permet de dire à Docker Compose qu'on veut exposer un port de notre machine hôte vers notre conteneur, et ainsi le rendre accessible depuis l'extérieur.

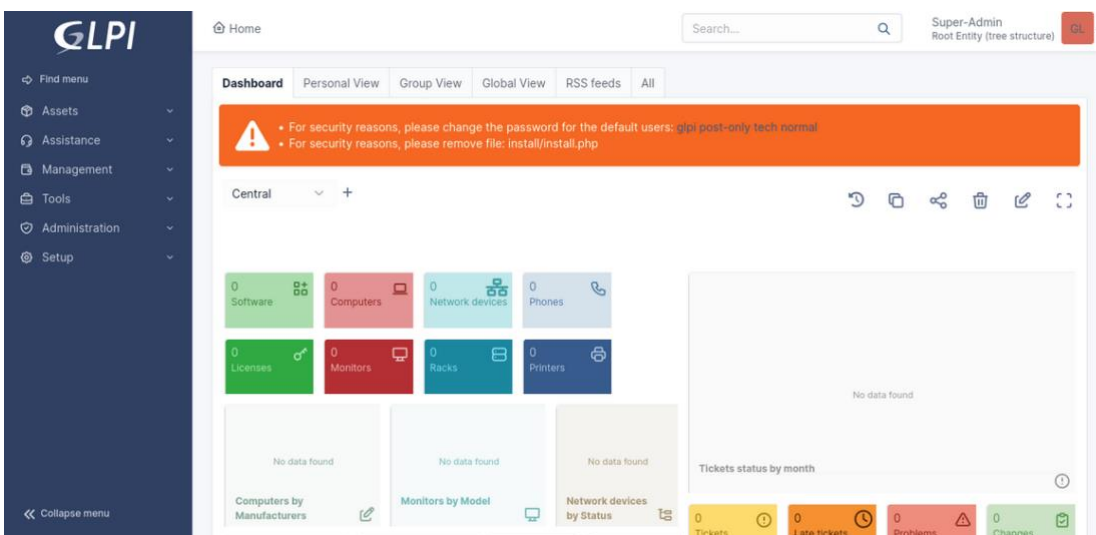
Pour lancer notre stack docker compose on utilise la commande :

```
docker compose up -d
```

Pour fermer les conteneurs on fait : *docker compose stop* .

On se connecte via notre navigateur internet à l'adresse localhost.





Vous retrouverez ce projets menée en autonomie ci-dessous :

<https://astro-portfolio-serres-nicolas.vercel.app/projet>