

NOUVEAU

TUTORIEL PRATIQUE



EXPOSER GLPI DERRIERE TRAEFIK 3.5 (DOCKER COMPOSE)

De l'installation de base aux premières manipulations

POINTS CLES DU TUTORIEL



Installation pas à pas de GLPI avec Docker Compose



Exposition de GLPI derrière Traefik 3.5



1^{ère} connexion à GLPI et manipulations POST-INSTALLATION

Note :

Ce tutoriel suppose que vous avez une machine Debian 13 ou un conteneur LXC Proxmox à disposition et que vous avez mis en place le reverse proxy © Traefik 3.5, sur une autre machine, conformément au tutoriel présent sur notre site : <https://tutos-info.fr>.

L'objectif de ce tutoriel est d'exposer l'helpdesk © GLPI derrière le reverse proxy © Traefik. GLPI est un logiciel libre de gestion de services d'assistance (issue tracking system et ServiceDesk) et de services informatiques (ITSM). Le code de cet outil est sous licence GPL.

Ce logiciel aide les entreprises à visualiser et gérer leurs systèmes d'information de manière complète et efficace. Il permet de créer un inventaire complet de toutes les ressources de l'organisation et de gérer les opérations administratives et financières.

Il s'agit d'une solution complète et adaptée à la plupart des parcs informatiques qu'ils soient grands, moyens ou petits. L'interface web du logiciel est claire, simple et intuitive. Elle est éditée en PHP/AJAX sur une base de données MySQL/MariaDB.

Pour réaliser ce tutoriel il vous faut :

- Une machine Debian 13 fonctionnelle avec 1 Go de RAM et 20 Go d'espace disque
- Ou un conteneur LXC Debian avec les mêmes caractéristiques
- Le reverse proxy © Traefik 3.5.0 installé sur une machine (voir tutoriel sur notre site : <https://tutos-info.fr>)

1 – MISE A JOUR DES PAQUETS

- Connectez-vous à votre nouvelle machine Debian (ou conteneur LXC) avec un utilisateur disposant des droits « sudo » et mettez à jour les paquets avec la commande suivante :

```
sudo apt update && apt upgrade -y
```

2 – INSTALLATION DE DOCKER ET DOCKER-COMPOSE AVEC LE SCRIPT OFFICIEL – SAUF SI DÉJÀ FAIT

En « mode laboratoire », il est possible d'installer Docker et docker Compose avec le script officiel. L'utilisation du script rend l'installation rapide et simple. Pour cela, saisissez les commandes suivantes :

```
apt install curl -y
curl -fsSL https://get.docker.com | sh
```

Note : Si vous travaillez en « production », il est recommandé d'ajouter votre utilisateur système au groupe docker. Ainsi, il ne sera plus nécessaire de saisir "sudo" avant toute commande. Pour cela, réalisez les étapes suivantes :

```
# Création du groupe "docker" :
groupadd docker
```

```
# Attribution du groupe à un utilisateur :
usermod -aG docker votre_nom_user
```

Déconnectez votre session Debian et reconnectez-vous.

Note : vous pouvez également installer Docker en vous aidant du site officiel et des sources officielles. Lien vers le site officiel : [Debian](#) | [Docker Docs](#)

Pour vérifier que Docker est bien installé sur votre machine Debian, saisissez la commande suivante :

sudo docker version

L'environnement Docker installé s'affiche :

```
debian@CT-DEBIAN-DOCKER:~$ sudo docker version
[sudo] password for debian:
Client: Docker Engine - Community
 Version:           28.3.3
 API version:       1.51
 Go version:        go1.24.5
 Git commit:        980b856
 Built:             Fri Jul 25 11:34:00 2025
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:           28.3.3
  API version:       1.51 (minimum version 1.24)
  Go version:        go1.24.5
  Git commit:        bea959c
  Built:             Fri Jul 25 11:34:00 2025
  OS/Arch:           linux/amd64
  Experimental:      false
 containerd:
  Version:           1.7.27
  GitCommit:         05044ec0a9a75232cad458027ca83437aae3f4da
 runc:
  Version:           1.2.5
  GitCommit:         v1.2.5-0-g59923ef
 docker-init:
  Version:           0.19.0
  GitCommit:         de40ad0
```

Pour afficher la version Docker Compose installée, saisissez la commande suivante :

sudo docker compose version

```
debian@CT-DEBIAN-DOCKER:~$ sudo docker compose version
Docker Compose version v2.39.1
```

En cas de problème, vous pouvez désinstaller Docker et Docker Compose avec les commandes suivantes :

systemctl stop docker

apt purge docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

1^{ère} étape : création des dossiers et sous-dossiers

Dans cette étape, nous créons les dossiers et sous-dossiers utiles à la création de notre « stack » GLPI avec les commandes suivantes (nous mettons les dossiers dans « /opt/containers » :

```
sudo mkdir -p /opt/containers/glpi/{data/glpi,data/marketplace,data/mysql}
```

On se place ensuite dans le dossier adéquat avec la commande suivante :

```
cd /opt/containers/glpi
```

2^{ème} étape : création d'un fichier « .env » (fichier caché avec les variables d'environnement)

- Créez le fichier « .env » dans « /opt/containers/glpi » à l'aide de la commande suivante :

```
sudo nano /opt/containers/glpi/.env
```

- Collez ou saisissez le contenu suivant :

```
# DB GLPI
GLPI_DB_HOST=db
GLPI_DB_PORT=3306
GLPI_DB_NAME=glpi
GLPI_DB_USER=glpi
GLPI_DB_PASSWORD=ChangeMe-GlpiDB! # à personnaliser

# DB root (pour opérations ponctuelles)
DB_ROOT_PASSWORD=ChangeMe-RootDB!

# Traefik
TRAEFIK_CERTRESOLVER=letsencrypt # nom de notre resolver

# Fuseau
TZ=Europe/Paris
```

- Quittez et sauvegardez le fichier avec **CTRL + O** et pressez la touche **Entrée**.

3^{ème} étape : création du fichier « docker-compose.yml » avec la dernière version de GLPI (10.0.19)

- Créez le fichier « **docker-compose.yml** » dans « /opt/containers/glpi » à l'aide de la commande suivante :

```
sudo nano /opt/containers/glpi/docker-compose.yml
```

- Collez ou saisissez le contenu suivant (**le fichier commence par 3 tirets**) :

```
---
```

```
services:
```

```
db:
```

```
image: mariadb:10.11
```

```
command: ["--character-set-server=utf8mb4","--collation-server=utf8mb4_unicode_ci"]
```

```
environment:
```

```
MARIADB_ROOT_PASSWORD: "${DB_ROOT_PASSWORD}"
```

```
MARIADB_DATABASE: "${GLPI_DB_NAME}"
```

```

MARIADB_USER: "${GLPI_DB_USER}"
MARIADB_PASSWORD: "${GLPI_DB_PASSWORD}"
TZ: "${TZ}"
volumes:
- ./data/mysql:/var/lib/mysql
healthcheck:
  test: ["CMD","mysqladmin","ping","-h","127.0.0.1","-u","${GLPI_DB_USER}","--
password=${GLPI_DB_PASSWORD}"]
  interval: 5s
  timeout: 3s
  retries: 30
  start_period: 20s

glpi:
  image: glpi/glpi:10.0.19
  restart: unless-stopped
  env_file: .env
  depends_on:
    db:
      condition: service_healthy
  environment:
    TZ: "${TZ}"
  volumes:
  - ./data/glpi:/var/glpi
  - ./data/marketplace:/var/www/glpi/marketplace
  ports:
  - "8800:80" # Ici on expose, par exemple, le port 8800 pour que Traefik y accède

glpi-cron:
  image: glpi/glpi:10.0.19
  restart: unless-stopped
  depends_on:
    glpi:
      condition: service_started
  environment:
    TZ: "${TZ}"
  volumes:
  - ./data/glpi:/var/glpi
  - ./data/marketplace:/var/www/glpi/marketplace
  entrypoint: ["/bin/sh","-c","while true; do php /var/www/glpi/front/cron.php; sleep 60; done"]

```

- Quittez et sauvegardez le fichier avec **CTRL + O** et pressez la touche **Entrée**.

Note : Pourquoi ces chemins ? L'image officielle glpi/glpi persiste config/, files/ et (particularité GLPI 10.x) le marketplace doit être monté en plus sur /var/www/glpi/marketplace.

4^{ème} étape : création du fichier de configuration dynamique sur Traefik

On crée maintenant le fichier de configuration dynamique (service et routeur) dans notre machine Traefik à l'aide de la commande suivante :

```
sudo nano /etc/traefik/dynamic/glpi.yml
```

- Copiez ou saisissez le contenu suivant :

http:

routers:

glpi-https:

rule: Host(`glpi.mon\_domaine.fr`)

entryPoints: [websecure]

service: glpi-svc

tls:

certResolver: letsencrypt

middlewares: [glpi-sec]

glpi-http:

rule: Host(`glpi.mon\_domaine.fr`)

entryPoints: [web]

middlewares: [redirect-to-https]

service: glpi-svc

middlewares:

redirect-to-https:

redirectScheme:

scheme: https

glpi-sec:

headers:

stsSeconds: 63072000

stsIncludeSubdomains: true

stsPreload: true

frameDeny: true

contentTypeNosniff: true

referrerPolicy: no-referrer-when-downgrade

services:

glpi-svc:

loadBalancer:

servers:

- url: "http://ip_machine_glpi:8800"

passHostHeader: true # (par défaut true)

- Quittez et sauvegardez le fichier avec **CTRL + O** et pressez la touche **Entrée**.

5^{ème} étape : lancement de la création de la stack « GLPI »

Sur votre machine GLPI, effectuez les manipulations suivantes :

cd /opt/containers/glpi

sudo docker compose up -d

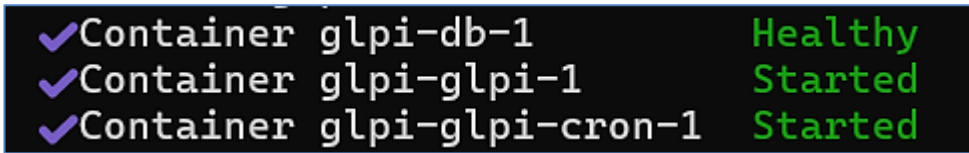
Patientez pendant la création de la stack complète GLPI :

```

debian@CT-DEBIAN-DOCKER:/opt/containers/glpi$ sudo docker compose up -d
[+] Running 1/39
glpi-cron [###] Pulling
  59e22667830b Extracting [>] 294.9kB/2...
  fe352bfe9f03 Download complete
  62c69d78aad7 Downloading [=====>] 13.36MB/...
  58b62470f55f Waiting
  a8b06518987c Waiting
  620e4c947637 Waiting
  551f0e34b27e Waiting
  0af5c19bcd88 Waiting
  afc8f7df9c61 Waiting
  f5a182ef7511 Waiting
  9a61f4dc56c8 Waiting
  83ca8d1d4995 Waiting

```

Une fois la stack créée, vous devez obtenir ceci :



Portainer affiche les nouveaux conteneurs en lien avec notre stack GLPI :

Containers									
Q Search... X									
Start Stop Kill Restart Pause Resume Remove Add container									
Name	State	Quick Actions	Stack	Image	Created	IP Address	Published Ports	Ownership	
glpi-db-1	healthy		glpi	mariadb:10.11	2025-08-16 16:27:20	172.18.0.2	-	administrators	
glpi-glpi-1	running		glpi	glpi/glpi:10.0.19	2025-08-16 16:27:20	172.18.0.3	8800:80	administrators	
glpi-glpi-cron-1	running		glpi	glpi/glpi:10.0.19	2025-08-16 16:27:20	172.18.0.4	-	administrators	

5 – TEST DE CONNEXION A GLPI

- Lancez votre navigateur et saisissez, dans la barre d’adresses, votre nom de domaine (ou votre WAN). Si vous avez tout configuré correctement, GLPI affiche la fenêtre d’authentification.
- Saisissez l’identifiant « **glpi** » et le mot de passe par défaut « **glpi** » pour cette 1^{ère} connexion :

Connexion à votre compte

Identifiant

glpi

Mot de passe

••••

Source de connexion

Base interne GLPI

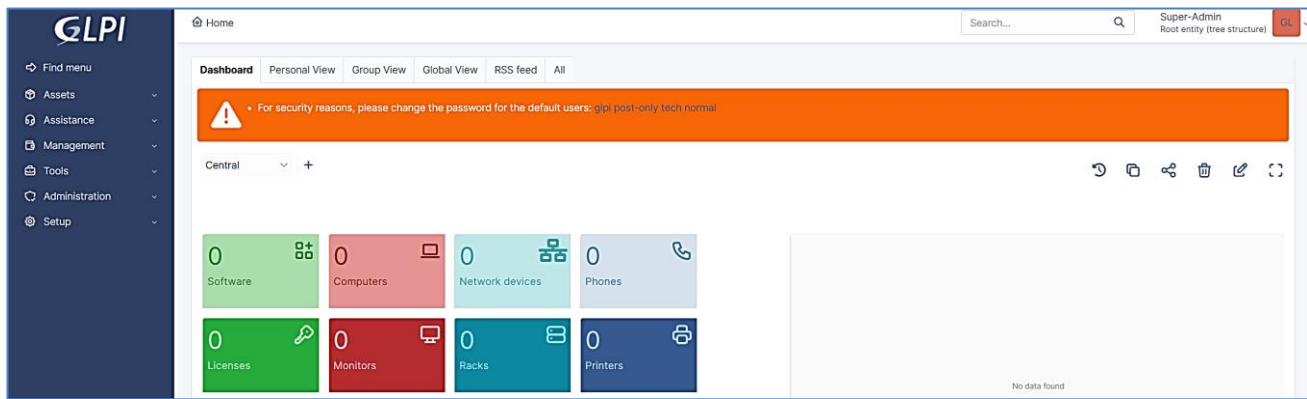
☒ Se souvenir de moi

Se connecter

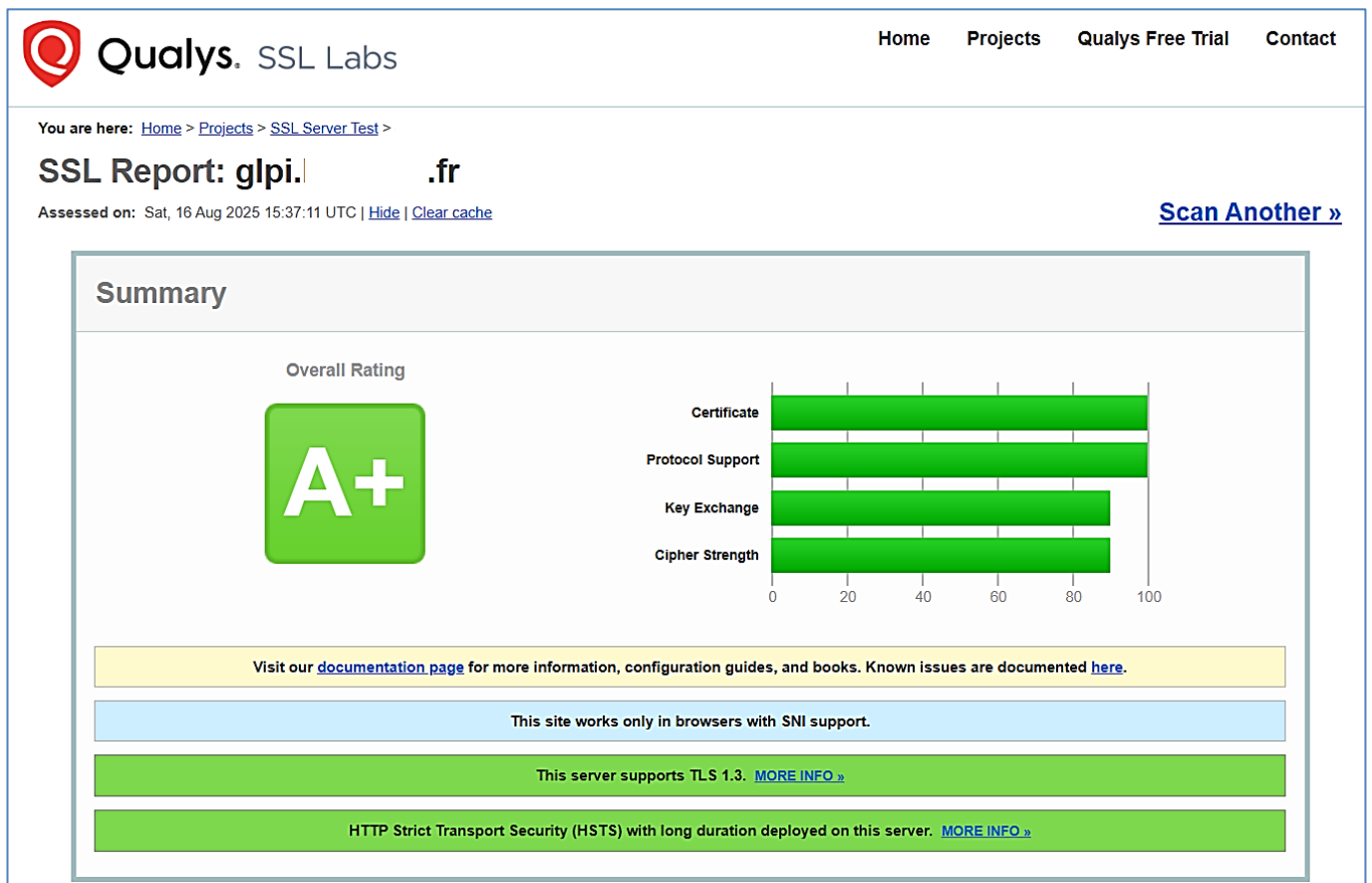
GLPI Copyright (C) 2015-2025 Teclib' and contributors

Note : Les comptes par défaut sont « glpi/glpi » (admin), « tech/tech », « normal/normal » et « post-only/post-only » → changez-les immédiatement pour des raisons de sécurité !

L'interface de gestion de GLPI (version 10.0.19 ici) s'affiche :



La note SSL Labs obtenue est « **A+** » :



6 – MANIPULATIONS « POST-INSTALLATION » RECOMMANDÉES

1^{ère} étape : activer les fuseaux horaires pour l'affichage correct des dates

- Donnez les droits à l'utilisateur GLPI sur les timezones à l'aide de la commande suivante :

```
cd /opt/containers/glpi
```

```
sudo docker exec -it $(sudo docker compose ps -q db) \  
sh -lc 'mysql -uroot -p"$MARIADB_ROOT_PASSWORD" \  
-e "GRANT SELECT ON mysql.time_zone_name TO \'glpi\'@"\'%"'; FLUSH PRIVILEGES;"
```

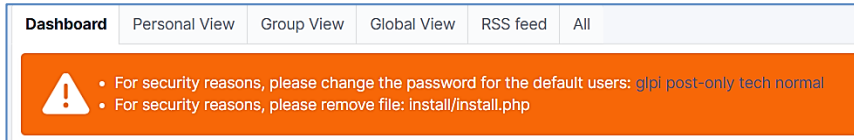
- Activez les timezones côté GLPI avec la commande suivante :

```
sudo docker exec -it $(sudo docker compose ps -q glpi) \
/var/www/glpi/bin/console database:enable_timezones
```

```
debian@CT-DEBIAN-DOCKER:/opt/containers/glpi$ sudo docker exec -it $(sudo docker compose ps -q glpi) \
/var/www/glpi/bin/console database:enable_timezones
Timezone usage has been enabled.
```

2^{ème} étape : supprimer l'installateur GLPI

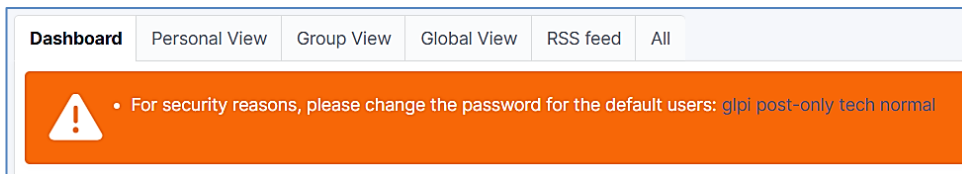
Lors de la 1^{ère} connexion à l'interface de gestion de GLPI, le message « **For security reasons, please remove file: install/install.php** » s'est affiché en haut de l'écran :



- Depuis votre machine GLPI, saisissez la commande suivante :

```
sudo docker compose exec -T glpi rm -f /var/www/glpi/install/install.php
```

- Actualisez la page GLPI ; le message a disparu. Il ne reste plus qu'à changer les mots de passe par défaut des utilisateurs de base :



3^{ème} étape : modifier le mot de passe de l'utilisateur administrateur « glpi » et changer la langue de l'interface

Depuis l'interface de gestion de GLPI, effectuez les manipulations suivantes :

- Cliquez, en haut à droite de l'écran, sur « **Super-Admin** » :



- Cliquez sur « **My settings** »
- Modifiez la langue dans la rubrique « **Language** » et saisissez votre nouveau mot de passe
- Cliquez le bouton jaune « **Save** » en bas de l'écran :

Language	Français ▼
Password	
Password confirmation	

Votre interface est maintenant en français et le message n'affiche plus qu'une alerte de mot de passe pour les autres utilisateurs de base ; changez les mots de passe pour effacer définitivement l'alerte.



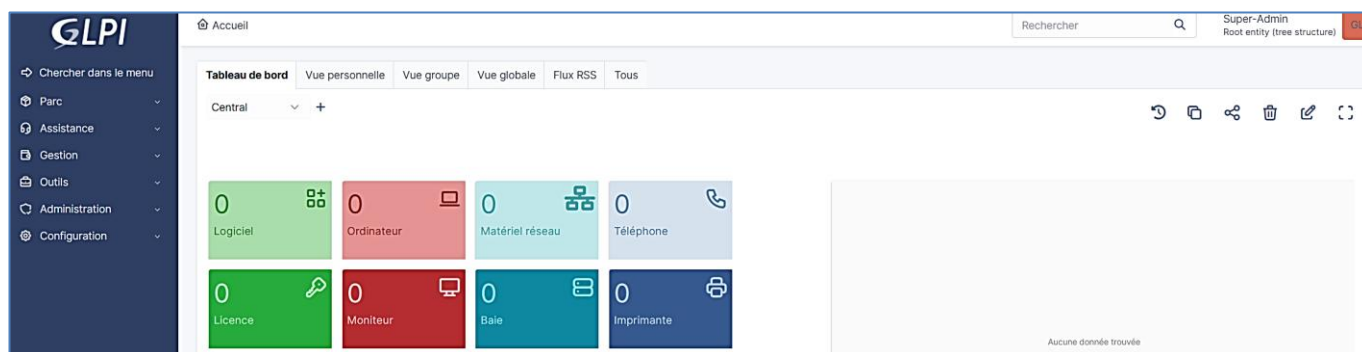
- Pour des raisons de sécurité, veuillez changer le mot de passe par défaut pour le(s) utilisateur(s) : post-only tech normal

Vous pouvez aussi supprimer un utilisateur de base si vous n'en n'avez pas l'utilité. Dans ce cas : cliquez sur le nom de l'utilisateur dans le message sur fond orange, par exemple « tech » et cliquez « Mettre à la corbeille » :



Mettre à la corbeille

L'interface de gestion est maintenant débarrassée des messages d'alerte :



FICHER « `docket-compose.yml` » - MACHINE GLPI (/opt/containers/glpi)

```

---

services:
  db:
    image: mariadb:10.11
    command: ["--character-set-server=utf8mb4","--collation-server=utf8mb4_unicode_ci"]
    environment:
      MARIADB_ROOT_PASSWORD: "${DB_ROOT_PASSWORD}"
      MARIADB_DATABASE: "${GLPI_DB_NAME}"
      MARIADB_USER: "${GLPI_DB_USER}"
      MARIADB_PASSWORD: "${GLPI_DB_PASSWORD}"
      TZ: "${TZ}"
    volumes:
      - ./data/mysql:/var/lib/mysql
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "127.0.0.1", "-u", "${GLPI_DB_USER}", "--password=${GLPI_DB_PASSWORD}"]
      interval: 5s
      timeout: 3s
      retries: 30
      start_period: 20s

  glpi:
    image: glpi/glpi:10.0.19
    restart: unless-stopped
    env_file: .env
    depends_on:
      db:
        condition: service_healthy
    environment:
      TZ: "${TZ}"
    volumes:
      - ./data/glpi:/var/glpi
      - ./data/marketplace:/var/www/glpi/marketplace
    ports:
      - "8800:80" # Ici on expose, par exemple, le port 8800 pour que Traefik y accède

  glpi-cron:
    image: glpi/glpi:10.0.19
    restart: unless-stopped
    depends_on:
      glpi:
        condition: service_started
    environment:
      TZ: "${TZ}"
    volumes:
      - ./data/glpi:/var/glpi
      - ./data/marketplace:/var/www/glpi/marketplace
    entrypoint: ["/bin/sh", "-c", "while true; do php /var/www/glpi/front/cron.php; sleep 60; done"]

```

```
http:
  routers:
    glpi-https:
      rule: Host(`glpi.mon_domaine.fr`)
      entryPoints: [websecure]
      service: glpi-svc
      tls:
        certResolver: letsencrypt
      middlewares: [glpi-sec]

    glpi-http:
      rule: Host(`glpi.mon_domaine.fr`)
      entryPoints: [web]
      middlewares: [redirect-to-https]
      service: glpi-svc

  middlewares:
    redirect-to-https:
      redirectScheme:
        scheme: https
    glpi-sec:
      headers:
        stsSeconds: 63072000
        stsIncludeSubdomains: true
        stsPreload: true
        frameDeny: true
        contentTypeNosniff: true
        referrerPolicy: no-referrer-when-downgrade

  services:
    glpi-svc:
      loadBalancer:
        servers:
          - url: "http://ip_machine_glpi:8800"
        passHostHeader: true # (par défaut true)
```

```
# DB GLPI
GLPI_DB_HOST=db
GLPI_DB_PORT=3306
GLPI_DB_NAME=glpi
GLPI_DB_USER=glpi
GLPI_DB_PASSWORD=ChangeMe-GlpiDB! # à personnaliser

# DB root (pour opérations ponctuelles)
DB_ROOT_PASSWORD=ChangeMe-RootDB!

# Traefik
TRAEFIK_CERTRESOLVER=letsencrypt # nom de notre resolver

# Fuseau
TZ=Europe/Paris
```