



ASSURMER AP n°5

INSTALLATION ET CONFIGURATION DE GLPI DANS UN ENVIRONNEMENT DOCKER

Réalisation AP n°5 - 22/05/2024 v1.1 - AUTEURS :

RODRIGUES Antoine, BOUSSAHA Elijah, Aymeric PERRET DU CRAY

Valdateur : Nassim MATOUK (maitre d'apprentissage Antoine)

TABLES DES MATIERES

Prérequis pour l'installation de GLPI sur Docker	3
Installation de Docker	4
Installation de Portainer.....	5

Prérequis pour l'installation de GLPI sur Docker

Pour installer GLPI, nous avons demandé au service infrastructure de nous fournir une machine physique comprenant certaines caractéristiques.

Cette machine sera l'hôte de Docker. Étant donné que Docker ne nécessite pas de grandes performances, nous avons opté pour une machine avec des spécifications modestes.

Voici les caractéristiques de la machine, nommée ASSURDOCKER :

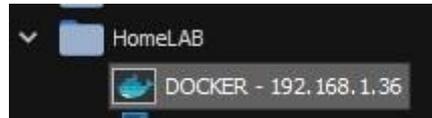
- 2 vCPU
- 8 Go de RAM
- SSD Nvme de 512 Go de stockage
- Système d'exploitation Debian 12 Bullseye

La connexion au réseau interne est également nécessaire pour effectuer la jonction à l'Active Directory, permettant ainsi aux collaborateurs d'accéder à GLPI avec leurs comptes personnels AD.

Installation de Docker

On accède à la machine via un accès SSH dont l'IP est 192.168.1.36.

1. La machine dispose d'un accès SSH incorporé et déjà configuré. On va donc se connecter au serveur en SSH grâce au logiciel MobaXTerm.



```
• MobaXterm 20.6 •
(SSSH client, X-server and networking tools)

► SSH session to root@192.168.1.36
  • SSH compression : ✓
  • SSH-browser      : ✓
  • X11-forwarding  : ✓ (remote display is forwarded through SSH)
  • DISPLAY         : ✓ (automatically set on remote server)

► For more info, ctrl+click on help or visit our website

Linux Debian 6.1.0-18-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-02-01) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 23 11:24:57 2024 from 192.168.1.16
root@Debian:~#
```

3. Ensuite il faut taper ces commandes qui sont les commandes pour l'installation de Docker fournis sur le site : <https://docs.docker.com/engine/install/debian/>
sudo apt-get update
sudo apt-get install ca-certificates curl sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \
"deb [arch=\$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
\$(. /etc/os-release && echo "\$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null apt-get
update

4. Nous vérifions si Docker est maintenant bien installé sur notre machine avec cette commande : `systemctl status Docker`

```

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 Wed 2024-05-22 14:59:06 CEST; 26h ago
     TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 552 (dockerd)
     Tasks: 106
    Memory: 225.0M
         CGroup: /system.slice/docker.service
                └─┬─ /usr/sbin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
                   ├─ /usr/sbin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 9443 -container-ip 172.17.0.2 -container-port 9443
                   ├─ /usr/sbin/docker-proxy -proto tcp -host-ip :: -host-port 9443 -container-ip 172.17.0.2 -container-port 9443
                   ├─ /usr/sbin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 9900 -container-ip 172.17.0.2 -container-port 9900
                   ├─ /usr/sbin/docker-proxy -proto tcp -host-ip :: -host-port 9900 -container-ip 172.17.0.2 -container-port 9900
                   ├─ /usr/sbin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 3306 -container-ip 172.19.0.7 -container-port 3306
                   ├─ /usr/sbin/docker-proxy -proto tcp -host-ip :: -host-port 3306 -container-ip 172.19.0.7 -container-port 3306
                   ├─ /usr/sbin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 443 -container-ip 172.19.0.9 -container-port 443
                   ├─ /usr/sbin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 80 -container-ip 172.19.0.9 -container-port 80
                   └─ /usr/sbin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 3000 -container-ip 172.19.0.13 -container-port 3000
                   └─ /usr/sbin/docker-proxy -proto tcp -host-ip :: -host-port 3000 -container-ip 172.19.0.13 -container-port 3000

mai 22 14:58:56 Docker-Host:AB dockerd[582]: time="2024-05-22T14:58:56.025005175+02:00" level=info msg="[core] Channel connectivity change to READY" module=grpc
mai 22 14:58:56 Docker-Host:AB dockerd[582]: time="2024-05-22T14:58:56.090545498+02:00" level=info msg="[graphdriver] using prior storage driver: overlay2"
mai 22 14:58:56 Docker-Host:AB dockerd[582]: time="2024-05-22T14:58:56.348901623+02:00" level=info msg="loading containers: start."
mai 22 14:58:57 Docker-Host:AB dockerd[582]: time="2024-05-22T14:58:57.496342102+02:00" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon option --bri
mai 22 14:59:04 Docker-Host:AB dockerd[582]: time="2024-05-22T14:59:04.090045050+02:00" level=info msg="loading containers: done."
mai 22 14:59:04 Docker-Host:AB dockerd[582]: time="2024-05-22T14:59:04.227052119+02:00" level=info msg="Docker daemon" commit=5d5db84 graphdriver(s)=overlay2 version=20.10.24+dfsg1
mai 22 14:59:04 Docker-Host:AB dockerd[582]: time="2024-05-22T14:59:04.258150003+02:00" level=info msg="Daemon has completed initialization"
mai 22 14:59:06 Docker-Host:AB systemctl[1]: Started docker.service - Docker Application Container Engine.
mai 22 14:59:06 Docker-Host:AB dockerd[582]: time="2024-05-22T14:59:06.369607389+02:00" level=info msg="API listen on /run/docker.sock"
mai 22 15:00:07 Docker-Host:AB dockerd[582]: time="2024-05-22T15:00:07.093524705+02:00" level=info msg="Ignoring event" container=418be179952fa59fd415ea4be8246457faa3934164ba71ff5f58ed2807f5c
lines 1-32/32 [END]

```

Docker est bien installé. On passe désormais à l'installation de Portainer.

Installation de Portainer

Avant de commencer, Que-est-ce que Portainer ?

Portainer est lui-même un container, permettant de gérer graphiquement des containers que l'on a installé sur notre machine Docker. Cela permet une gestion plus simple, à l'aide d'une interface utilisateur simple d'utilisation et intuitive sur un navigateur web.

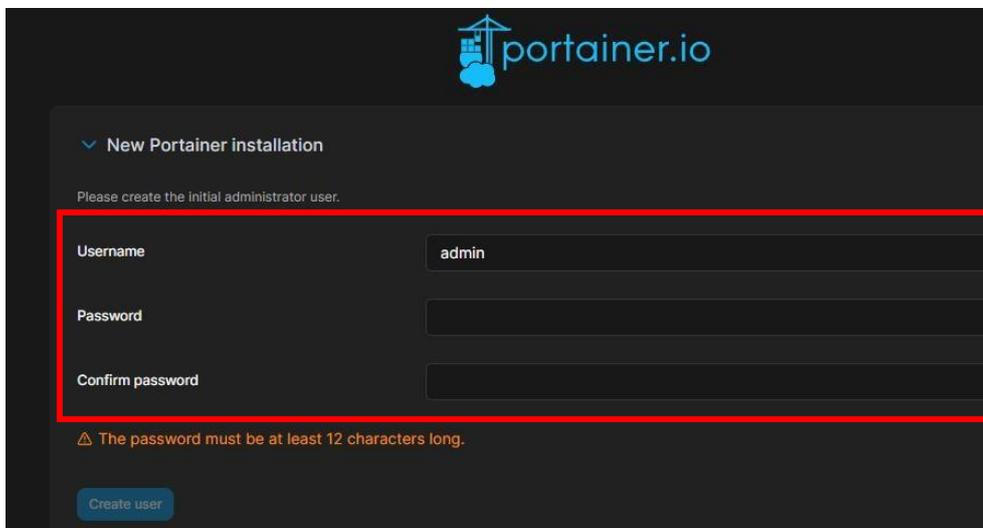
1. Toujours via SSH, on lance cette commande : `docker run -d -p 9000:9000 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest` Cela va nous permettre d'installer le container "portainer" sur le port 9000.

```

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
Unable to find image 'portainer/portainer-ce:latest' locally
latest: Pulling from portainer/portainer-ce
57654d40e0a5: Pull complete
1f476acfabd6: Pull complete
5171176db7f2: Pull complete
52e9438966a5: Pull complete
43d4775415ac: Pull complete
c1cad9f5288f: Pull complete
a5e2b359b78b: Pull complete
eb172612bcb: Pull complete
0be7b2ac1fb5: Pull complete
391dff0fb880: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:4a1ceaddf7898d9190ee0a6d22234c4323aefd80e796e84f5e57127f4370f1
Status: Downloaded newer image for portainer/portainer-ce:latest
d1288ff26f549ff4b898780754a8751d8ecdbb9e01bc11af8c1ebfee0603ede
root@debian:~#

```

- Maintenant il faut accéder à l'interface Web de Portainer, à l'adresse 192.168.1.36:9000 et ensuite créé les [identifiants administrateurs](#).



portainer.io

▼ New Portainer installation

Please create the initial administrator user.

Username

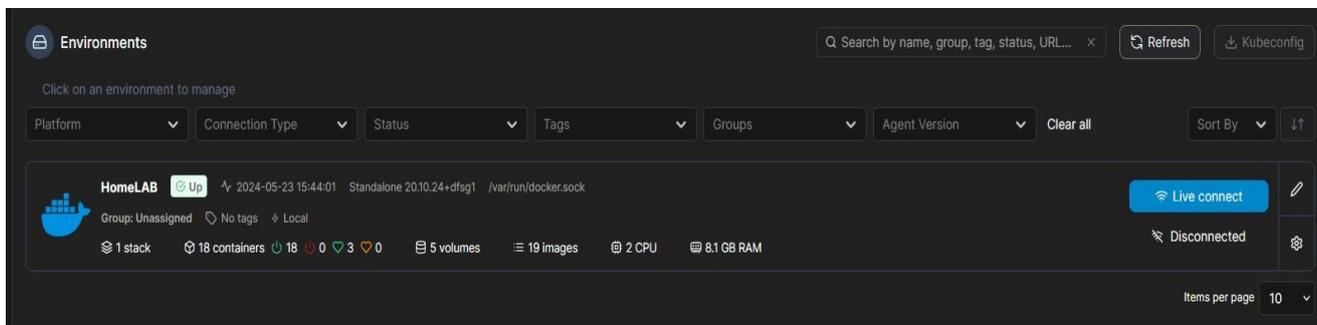
Password

Confirm password

⚠ The password must be at least 12 characters long.

Create user

- Portainer reconnais automatiquement l'environnement Docker que nous avons installé précédemment.



Environments

Q Search by name, group, tag, status, URL... × Refresh Kubeconfig

Click on an environment to manage

Platform Connection Type Status Tags Groups Agent Version Clear all Sort By ↓↑

HomeLAB Up 2024-05-23 15:44:01 Standalone 20.10.24+dfsg1 /var/run/docker.sock

Group: Unassigned No tags Local

1 stack 18 containers 18 up 0 down 3 pending 0 stopped 5 volumes 19 images 2 CPU 8.1 GB RAM

Live connect Disconnected

Items per page 10



Installation et configuration de la base de données MySQL + PhpMyAdmin



Nous allons utiliser l'image Docker (version dérivée de l'initial) de GLPI créé par diouxx (<https://github.com/DiouX/docker-glpi>). Cette image est très renommé et régulièrement mis à jour, elle est elle même open-source (pour être en accord avec la licence de GLPI), et il est donc possible de vérifier très rapidement son contenu et sa fiabilité.

Nous allons nous baser sur l'installation avec une base de données (MySQL) et données persistantes, vouée à l'utilisation en environnement de production. Ce type de base de données est aussi faite pour gérer une grande quantité de données le tout en restant très optimiser et peu gourmand en ressources.

Nous allons également rajouter un container PhPMyAdmin qui nous permettra de gérer toutes les bases de données dont celle de GLPI plus efficacement via une interface graphique. Cet ajout n'est pas nécessaire mais dans un but de facilité et d'un environnement de production c'est ce qu'il nous faut.

1. C'est le début ! Et comme au début rien n'est créé alors nous allons devoir créer pour la première et dernière fois une solution de reverse proxy avec nginx qui gère aussi les certificats dans un container docker afin de principalement rediriger les différents trafics (ex: glpi.assurmer-asurance.fr) vers les bons containers. Pour commencer nous allons créer un dossier nginx dans le /home et y insérer le contenu ce trouvant sur ce repo [github](https://github.com). Enfin, nous allons exécuter cette commande : `./fresh-start.sh --yes --skip-docker-image-check --use-nginx-conf-files -e contact@antoinerodrigues.com` pour crée le container. Maintenant qu'il est crée il n'y à plus qu'a rajouter les variables d'environnement :

```
-e VIRTUAL_HOST=glpi.antoinerodrigues.com
```

```
-e LETSENCRYPT_HOST=glpi.antoinerodrigues.com
```

Pour que le container fasse le tout automatiquement de la gestion des certificats SSL aux redirections :

```
root@Debian:~# ./fresh-start.sh --yes --skip-docker-image-check --use-nginx-conf-files -e contact@antoinerodrigues.com
```

Exemple avec mysql :

```
Creating/renewal mysql.antoinerodrigues.com certificates... (mysql.antoinerodrigues.com)
[Thu May 23 16:03:49 UTC 2024] Domains not changed.
[Thu May 23 16:03:49 UTC 2024] Skip, Next renewal time is: Sat Jul 6 21:26:24 UTC 2024
[Thu May 23 16:03:49 UTC 2024] Add '--force' to force to renew.
```

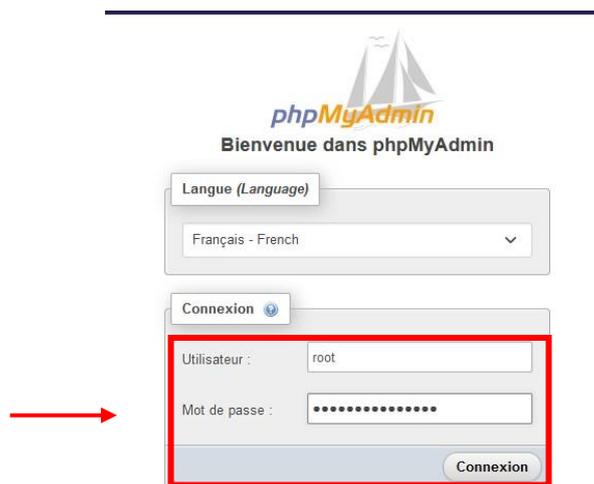
2. On créer le container qui contiendra la base de données MySQL avec la commande : `docker run -p 3306:3306 -d -v /home/mysql/data:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=[MotDePasseSQLRobuste] -e MYSQL_ROOT_HOST='% ' --restart=always --network=proxy --name mysql mysql`

```
root@Debian:~# docker run -p 3306:3306 -d -v /home/mysql/data:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=[MotDePasseSQLRobuste] -e MYSQL_ROOT_HOST='% ' --restart=always --network=proxy --name mysql mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
fcbdc4900331: Pull complete
35db5973e36f: Pull complete
5e5ba9e6412a: Pull complete
ee4654eb29b7: Pull complete
1d22181e6b86: Pull complete
bb9712936412: Pull complete
4a6ef2152809: Pull complete
0f7c3b67bb2b: Pull complete
7ccd17e83c8a: Pull complete
04b72c110556: Pull complete
Digest: sha256:4a4e5e2d199ab7ae7070588052e8f401e17a330466ecfc55c9acf51196da5bd0
Status: Downloaded newer image for mysql:latest
bae6311d0098ba65fa7a321f4dbed01a249d655d0dbaf5e1abd1301752ba2c10
root@Debian:~#
```

- Et on fait la même chose pour le container PhpMyAdmin : `docker run -d -e PMA_PMADB=phpmyadmin -e PMA_QUERYHISTORYDB=true -e PMA_HOST=mysql -e PMA_PORT=3306 -e MYSQL_ROOT_PASSWORD= [MotDePasseSQLRobuste] -restart=always --network=proxy --name phpmyadmin phpmyadmin/phpmyadmin`

```
root@Debian:/home/mysql# docker run -d -e PMA_PMADB=phpmyadmin -e PMA_QUERYHISTORYDB=true -e PMA_HOST=mysql -e PMA_PORT=3306 -e MYSQL_ROOT_PASSWORD= [MotDePasseSQLRobuste] --restart=always --network=proxy --name phpmyadmin phpmyadmin/phpmyadmin
```

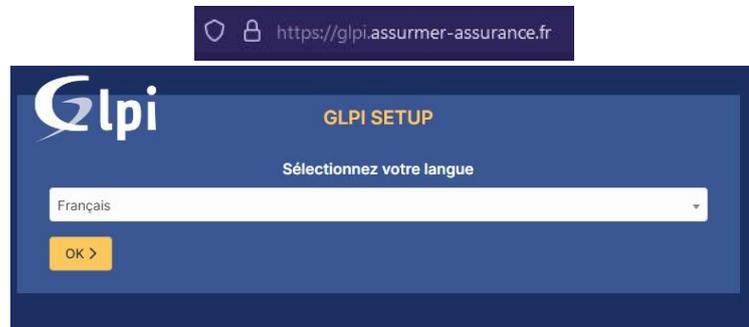
Puis on vérifie que phpmyadmin fonctionne bien :



- Et l'on va faire de même pour GLPI: `docker run -d --user "${id -u}" --volume "$PWD/glpi/var/www/html/glpi:/var/www/html/glpi" --env-file /home/glpi/mariadb.env -name=glpi --restart=always --network=proxy diouxx/glpi`
Comme le but étant de mettre GLPI en environnement de production nous avons fait en sorte de rediriger le nom de domaine glpi.assurmer-assurance.fr vers le container GLPI que nous venons de créer sur docker grâce au système de reverse proxy que nous avons précédemment créé / configuré.

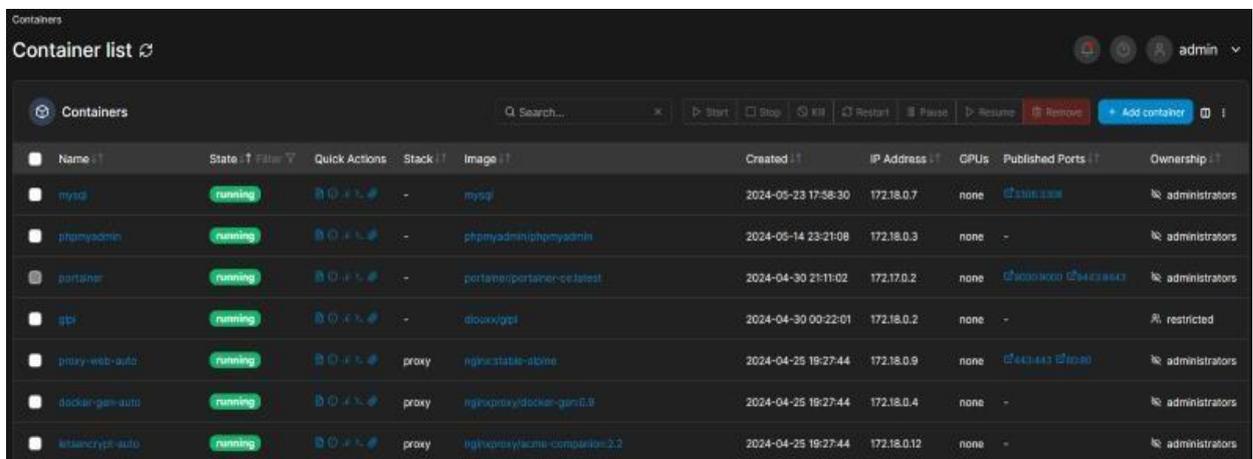
```
root@Debian:~# docker run -d --user "${id -u}" --volume "$PWD/glpi/var/www/html/glpi:/var/www/html/glpi" --env-file /home/glpi/mariadb.env --name=glpi --restart=always --network=proxy diouxx/glpi
```

Et on vérifie la encore si GLPI est bien accessible depuis l'url glpi.assurmer-assurance.fr :



A la suite de toutes ces commandes effectuées en SSH nous devrions maintenant avoirs 7 containers créés et visible sur portainer :

- 3 pour le reverse proxy (proxy-web-auto, docker-gen-auto, letsencrypt-auto)
- 1 pour GLPI (glpi)
- 2 pour la gestion de base de données (mysql, phpmyadmin)
- 1 pour la gestion de containers (portainer) :



The screenshot shows the Portainer 'Container list' interface. It features a search bar and several action buttons (Start, Stop, Kill, Restart, Pause, Resume, Remove, Add container). Below the buttons is a table with the following columns: Name, State, Quick Actions, Stack, Image, Created, IP Address, GPUs, Published Ports, and Ownership. The table lists seven containers, all in a 'running' state.

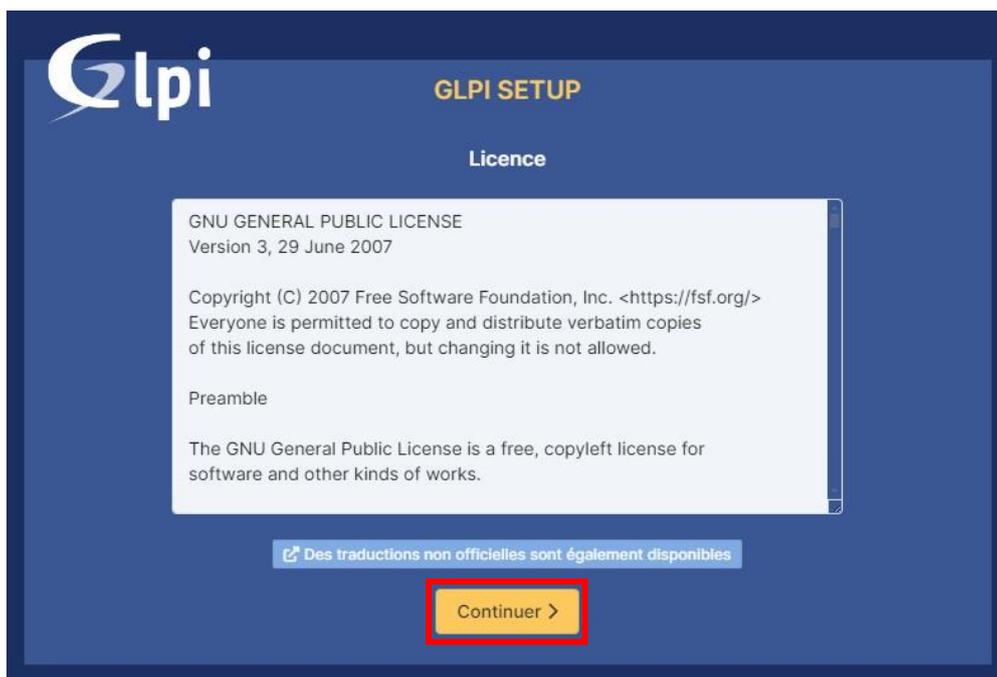
Name	State	Quick Actions	Stack	Image	Created	IP Address	GPUs	Published Ports	Ownership
mysql	running	[Stop] [Start] [Kill] [Restart] [Pause] [Resume] [Remove]	-	mysql	2024-05-23 17:58:30	172.18.0.7	none	3306:3306	administrators
phpmyadmin	running	[Stop] [Start] [Kill] [Restart] [Pause] [Resume] [Remove]	-	phpmyadmin/phpmyadmin	2024-05-14 23:21:08	172.18.0.3	none	-	administrators
portainer	running	[Stop] [Start] [Kill] [Restart] [Pause] [Resume] [Remove]	-	portainer/portainer-ce:latest	2024-04-30 21:11:02	172.17.0.2	none	9000:9000 3443:3443	administrators
glpi	running	[Stop] [Start] [Kill] [Restart] [Pause] [Resume] [Remove]	-	elocoox/glpi	2024-04-30 00:22:01	172.18.0.2	none	-	restricted
proxy-web-auto	running	[Stop] [Start] [Kill] [Restart] [Pause] [Resume] [Remove]	proxy	nginxstable-alpine	2024-04-25 19:27:44	172.18.0.9	none	80:80 443:443	administrators
docker-gen-auto	running	[Stop] [Start] [Kill] [Restart] [Pause] [Resume] [Remove]	proxy	nginxproxy/docker-gen:0.9	2024-04-25 19:27:44	172.18.0.4	none	-	administrators
letsencrypt-auto	running	[Stop] [Start] [Kill] [Restart] [Pause] [Resume] [Remove]	proxy	letsencrypt/letsencrypt:2.2	2024-04-25 19:27:44	172.18.0.12	none	-	administrators

GLPI Configuration partie 1 (de base) de GLPI

1. Pour commencer la configuration de glpi il faut cliquer sur « [OK](#) »



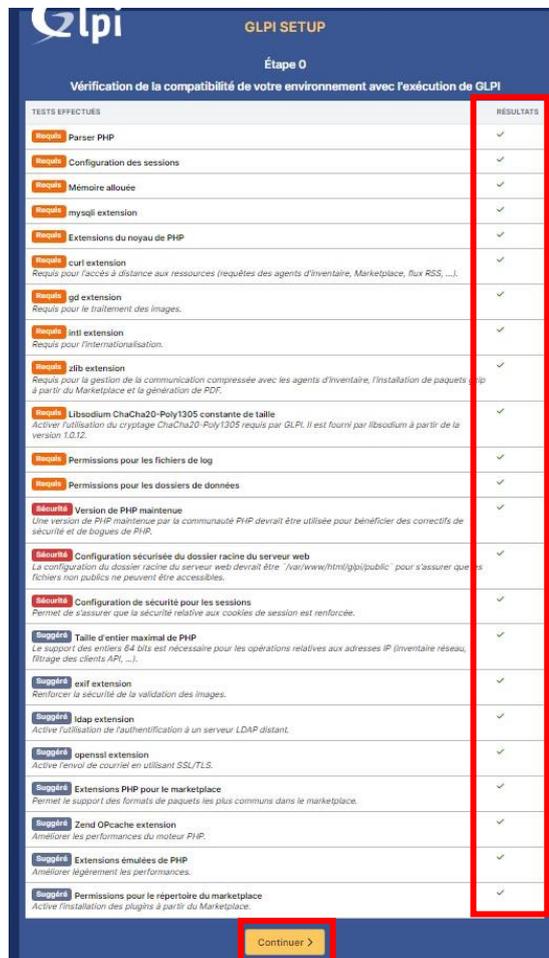
2. Ensuite cliquer sur « [Continuer](#) »



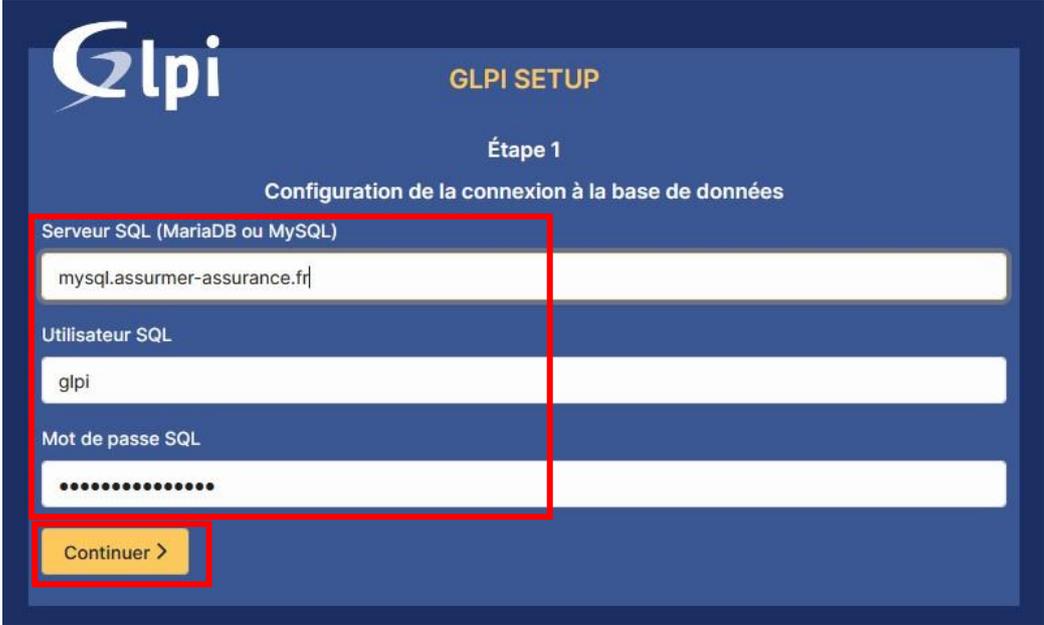
3. Puis sur « [Installer](#) »



4. Veillez à ce que tout soit bien sélectionné et cliquer sur « [Continuer](#) ».

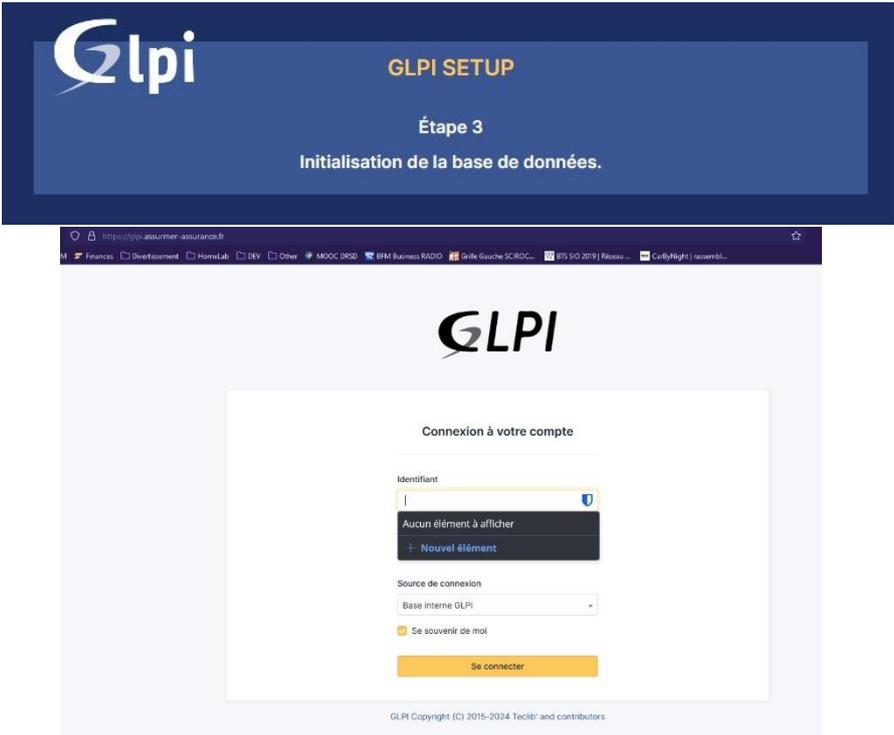


5. Entrer les informations de connexion à la base de données MySQL. Cliquer sur « Continuer ».



The screenshot shows the 'GLPI SETUP' interface at 'Étape 1' titled 'Configuration de la connexion à la base de données'. It features three input fields: 'Serveur SQL (MariaDB ou MySQL)' containing 'mysql.assurmer-assurance.fr', 'Utilisateur SQL' containing 'glpi', and 'Mot de passe SQL' which is masked with dots. A yellow 'Continuer >' button is located at the bottom. Red arrows point to the 'Continuer >' button and the 'Serveur SQL' field.

6. Une fois la base de données initialisé la page de connexion apparait.



The top part of the screenshot shows 'Étape 3' titled 'Initialisation de la base de données.' Below this, a browser window displays the GLPI login page. The page has the GLPI logo at the top and a 'Connexion à votre compte' section. It includes an 'Identifiant' field with a dropdown menu showing 'Aucun élément à afficher' and a 'Nouvel élément' button. Below that is a 'Source de connexion' dropdown menu set to 'Base interne GLPI' and a checked 'Se souvenir de moi' checkbox. A yellow 'Se connecter' button is at the bottom of the form. The footer of the page reads 'GLPI Copyright (C) 2015-2024 Teclib' and contributors'.