

TP N°11 BLOC N°2 BIS



ORCHESTRATION DE CONTENEURS MULTI-SERVICES AVEC DOCKER COMPOSE : DÉPLOIEMENT DES APPLICATIONS GLPI ET OCS INVENTORY

SOMMAIRE

Déploiement du service web Nginx dans un conteneur géré à l'aide des composants logiciels de DOCKER

Etape 1 : Création de la VM "SRVDOCKER" et installation des composants logiciels de Docker.....Page 1

Etape 2 : création de la structure du projet "WordPress" et définition de son fichier "docker-compose.yaml".....Page 1

Etape 3 : Création de la structure du projet "GLPI OCS" et définition de son fichier "docker-compose.yaml".....Page 3

ETAPE 1 : Création de la VM "SRVDOCKER" et installation des composants logiciels de Docker

Pour commencer ce TP 11 BIS il faudra juste mettre en place un serveur docker ou réutiliser le serveur docker que tu nous avons utiliser pour le TP 11

ETAPE 2 : Création de la structure du projet "WordPress" et définition de son fichier "docker-compose.yml"

14.) Dans cette étape nous allons créer la structure du projet "WordPress", pour commencer depuis, le "SRVDOCKER", nous allons nous connectez en tant qu'Administrateur puis créez un répertoire nommé, pour cela saisissez la commande pour créez le répertoire wordpress :

- `mkdir wordpress`

15.) Déplacez vous dans le répertoire wordpress en saisissant la commande :

- `cd workpress`

puis éditer le fichier "docker.compose.yml, en saisissant la commande :

- `nano docker.compose.yml`

avec le contenue suivant :

```
services:
  # Database
  bdd:
    image: mysql:latest
    volumes:
      - './db_data:/var/lib/mysql'
    ports:
      - '3306:3306'
    restart: always
    environment:
      MYSQL_DATABASE: wordpress
      MYSQL_USER: tepson
      MYSQL_PASSWORD: tepson2526
      MYSQL_ROOT_PASSWORD: admin2526

  # PhpMyAdmin
  phpmyadmin:
    depends_on:
      - bdd
    image: phpmyadmin/phpmyadmin:latest
    restart: always
    ports:
      - '8084:80'
    environment:
      PMA_HOST: bdd
      MYSQL_ROOT_PASSWORD: admin2526

  # Wordpress
  wordpress:
    depends_on:
      - bdd
    image: wordpress:latest
    ports:
      - '8080:80'
    environment:
      WORDPRESS_DB_HOST: bdd
      WORDPRESS_DB_USER: tepson
      WORDPRESS_DB_PASSWORD: tepson2526
      WORDPRESS_DB_NAME: wordpress
    restart: always
```

16.) Dans le fichier docker.compose.yml, on peut constater différent service :

- **Service Database :**
 - **Image :** mysql:latest : Définit l'image Docker utilisée (MySQL dernière version)
 - **volumes :** - './db_data:/var/lib/mysql' : Monte le répertoire local db_data dans le conteneur pour persister les données (empêche la perte des bases lors de la suppression du conteneur)
 - **ports :** - '3306:3306' : Redirige le port 3306 de l'hôte vers le port 3306 du conteneur (permet d'accéder à MySQL depuis l'extérieur)
 - **restart :** always : Redémarre automatiquement le conteneur en cas d'arrêt (panne, crash)
 - **environment :** Définit les variables d'environnement pour configurer MySQL au démarrage
 - **MYSQL_DATABASE :** wordpress : crée automatiquement une base de données nommée wordpress
 - **MYSQL_USER :** tepson : crée l'utilisateur tepson (non-root) pour se connecter à MySQL
 - **MYSQL_PASSWORD :** tepson2526 : Mot de passe de l'utilisateur tepson
 - **MYSQL_ROOT_PASSWORD :** admin2526 : Mot de passe du compte root (super-admin MySQL)
- **Service Phpmyadmi :**
 - **depends_on :** - bdd : PhpMyAdmin attend que le service bdd (MySQL) soit démarré avant de se lancer
 - **image :** phpmyadmin/phpmyadmin:latest : Utilise la dernière version officielle de l'image PhpMyAdmin
 - **restart :** always : Redémarre automatiquement le conteneur en cas d'arrêt (panne, crash)
 - **ports :** - '8084:80' : Redirige le port 8084 de l'hôte vers le port 80 du conteneur // accès via http://Adresse_IP_Du_Conteneur:8084
 - **PMA_HOST :** bdd : indique à PhpMyAdmin le nom du service où se trouve MySQL (bdd)
 - **MYSQL_ROOT_PASSWORD :** admin2526 : Mot de passe root pour se connecter automatiquement
- **Service Wordpress :**
 - **depends_on :** - bdd : WordPress attend que MySQL soit prêt avant de démarrer
 - **image :** wordpress:latest : Utilise la dernière version officielle de l'image Wordpress
 - **ports :** - '8080:80' : Redirige le port 8080 de l'hôte vers le port 80 du conteneur // accès via http://Adresse_IP_Du_Conteneur:8080
 - **WORDPRESS_DB_HOST :** bdd : Indique à WordPress le nom du service MySQL (les conteneurs communiquent via le réseau Docker)
 - **WORDPRESS_DB_USER :** tepson : L'utilisateur MySQL utilisé par WordPress pour se connecter
 - **WORDPRESS_DB_PASSWORD :** tepson2526 : Mot de passe de l'utilisateur MySQL
 - **WORDPRESS_DB_NAME :** wordpress : Nom de la base de données utilisée par WordPress

- restart: always : Redémarre automatiquement le WordPresse s'arrête (panne, crash)

17.) Nous allons orchestrer l'ensemble du processus de création de l'architecture Docker précédente, Pour cela saisissez la commande :

- **docker compose up -d**

Ici l'option permet de lancer les conteneurs en arrière-plan, ainsi le terminal reste libre

18.) Voici ce qu'on constate après avoir saisi la commande précédente

```
root@tepson:/home/tepson/wordpress# docker compose up -d
[+] up 3/3
✓ Container wordpress-bdd-1      Running
✓ Container wordpress-wordpress-1 Running
✓ Container wordpress-phpmyadmin-1 Running
root@tepson:/home/tepson/wordpress#
```

ont constate que la commande docker compose up -d a crée un réseau virtuel wordpress_default, les volumes nécessaires, puis a démarré les trois conteneurs (bdd, phpmyadmin, wordpress). Tous les conteneurs ont le statut Up (en cours d'exécution).

vérifions le nom du réseau virtuel qui permet la communication entre les conteneurs, pour cela saisissez la commande :

- **docker network ls**

```
root@tepson:/home/tepson/wordpress# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
a2a827048b4f        bridge              bridge              local
55c5e797ddcd        host                host                local
80b072e8aa25        none                null                local
e8883015b4b5        tepson_default      bridge              local
9757b764a8a5        wordpress_default   bridge              local
root@tepson:/home/tepson/wordpress#
```

Le nom du réseau virtuel est wordpress_default

19.) saisissez la commande docker compose ps pour vérifiez que les conteneurs tournent.

```
root@tepson:/home/tepson/wordpress# docker compose ps
NAME                IMAGE                COMMAND              SERVICE    CREATED      STATUS      PORTS
wordpress-bdd-1     mysql:latest        "docker-entrypoint.s" bdd        20 minutes ago Up 20 minutes 0.0.0.0:3306->3306/tcp, [::]:3306->3306/tcp, 33060/tcp
wordpress-phpmyadmin-1 phpmyadmin/phpmyadmin:latest "/docker-entrypoint.s" phpmyadmin 20 minutes ago Up 20 minutes 0.0.0.0:8084->80/tcp, [::]:8084->80/tcp
wordpress-wordpress-1 wordpress:latest     "docker-entrypoint.s" wordpress 20 minutes ago Up 20 minutes 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
root@tepson:/home/tepson/wordpress#
```

La commande docker compose ps montre que les trois conteneurs (bdd, phpmyadmin, wordpress) sont en cours d'exécution (statut Up), avec leurs ports respectifs redirigés vers l'hôte.

20.) Pour tester l'accès au SGBDR MySQL depuis l'appliation web PHPMyAdmin voulez tout d'abord vous connectez vous à votre Phpmyadmin via votre navigateur :

- **http://Adresse_IP_SRVDOCKER:8084**

8084 port d'écoute pour le service php

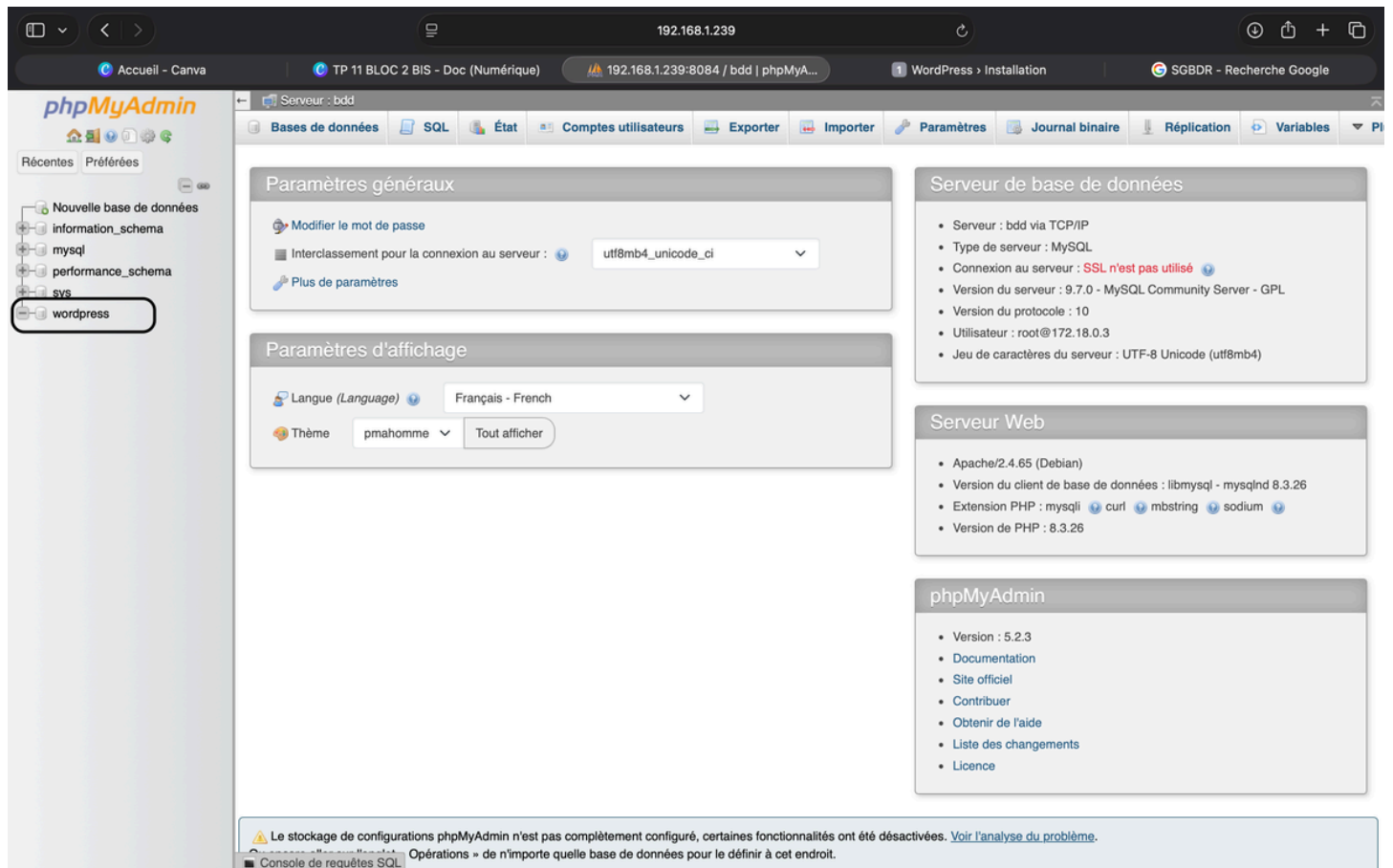
pour ma par c'est : http://192.168.1.239:8084



Connectez vous avec votre compte Administrateur :

- **Utilisateur : root**
- **Mot de passe : admin2526**

Vous tomberez sur l'interface de phpmyadmin



À gauche ce trouve la liste des base de données, ainsi que la base de donnée (wordpress).

21.) Maintenant nous allons tester depuis notre poste de travail l'accès à l'application Wordpress et ainsi finaliser sont installation pour héberger le site du CHU.

Tout d'abord via votre navigateur saisissez ceci à fin de vous connectez à wordpress :

- http://Adresse_IP_SRVDOCKER:8080

pour ma par c'est : <http://192.168.1.239:8080>

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password [Hide](#)

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search engine visibility ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

[Install WordPress](#)

Il vous faudra remplir certaine information :

- **Site Title : CHU de Pointe-à-Pitre**
- **Username : admin**
- **Password : admin2526**
- **Your Email : admin@chu-guadeloupe@gmail.com**

Message avec un message indiquant que votre WordPress à été installer



Success!

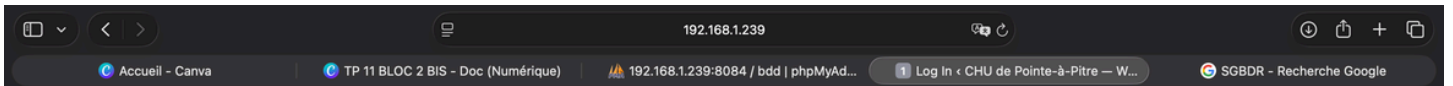
WordPress has been installed. Thank you, and enjoy!

Username admin

Password *Your chosen password.*

[Log In](#)

Connecter vous avec votre compte crée préalablement.



Username or Email Address

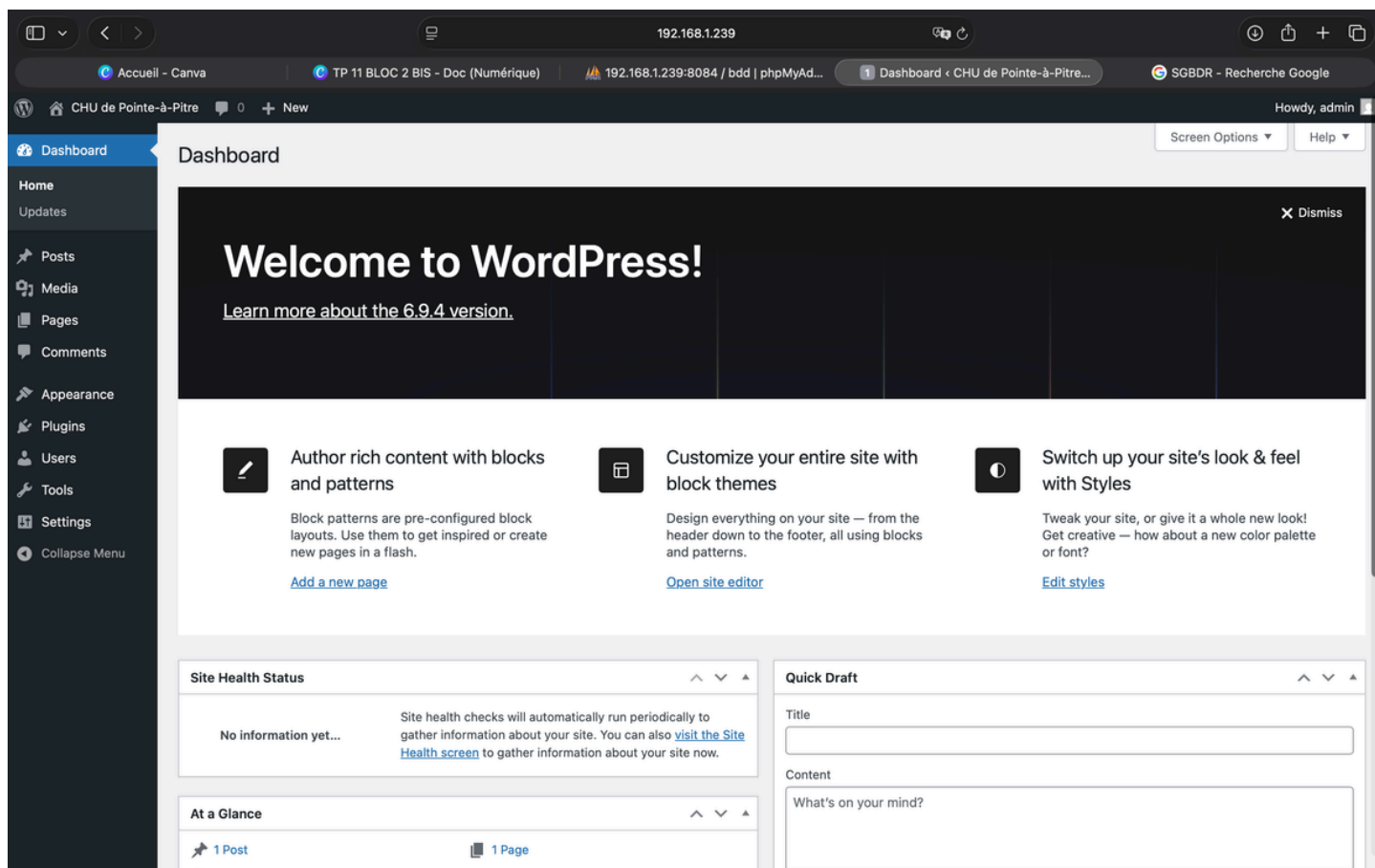
Password

☐ Remember Me

Log In

[Lost your password?](#)

[← Go to CHU de Pointe-à-Pitre](#)

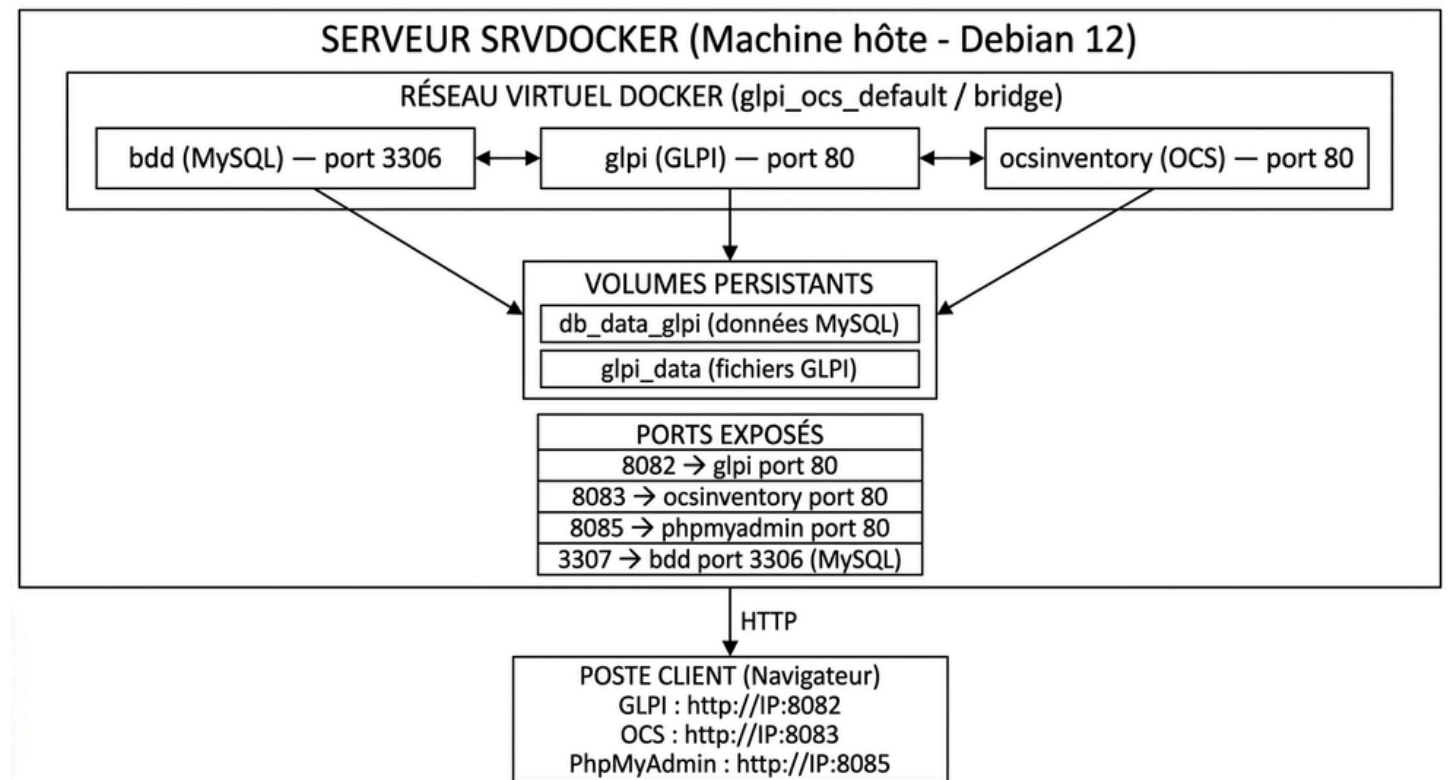


Nous avons dès maintenant accès à l'interface de WordPress

ETAPE 3 : création de la structure du projet "GLPI OCS" et définition de son fichier "docker-compose.yml"

Dans cette dernière étape, nous allons créer une structure du projet "GLPI OCS"

22.) Voici le schéma qui représente l'architecture Docker attendue



23 & 24.) Maintenant nous allons orchestrer ces services dans des conteneurs avec Docker Compose.

Pour commencer déplacer dans le répertoire /home/tepson : pour cela saisissez la commande :

- `cd /home/tepson`

et créer le répertoire /glpi-ocs en saisisant la commande :

- `mkdir glpi-ocs`

Maintenant déplacer vous dans le répertoire glpi-ocs en saisisant la commande :

- `cd glpi-ocs`

Maintenant éditer un nouveau fichier "docker-compose.yml" :

- `nano docker-compose.yml`

Insérer ce contenu :

```

root@tepson:/home/tepson/glpi-ocs# cat docker-compose.yml
services:
  mysql:
    image: mysql:8.0
    container_name: mysql_db
    restart: always
    command: --default-authentication-plugin=mysql_native_password
    environment:
      MYSQL_ROOT_PASSWORD: admin2526
      MYSQL_USER: ocs
      MYSQL_PASSWORD: ocs2526
    volumes:
      - db_data:/var/lib/mysql

  phpmyadmin:
    image: phpmyadmin:5.2
    container_name: phpmyadmin
    depends_on:
      - mysql
    restart: always
    ports:
      - "8085:80"
    environment:
      PMA_HOST: mysql
      PMA_PORT: 3306
      MYSQL_ROOT_PASSWORD: admin2526

  glpi:
    image: diouxx/glpi
    container_name: glpi
    depends_on:
      - mysql
    restart: always
    ports:
      - "8082:80"
    volumes:
      - glpi_data:/var/www/html/glpi

  ocsinventory:
    image: blinzlerone/ocsinventory-docker-image:latest
    container_name: ocsinventory
    depends_on:
      - mysql
    restart: always
    ports:
      - "8083:80"
    environment:
      OCS_DB_SERVER: 172.19.0.2
      OCS_DB_HOST: 172.19.0.2
      OCS_DBSERVER_READ: 172.19.0.2
      OCS_DBSERVER_WRITE: 172.19.0.2
      OCS_DB_USER: ocs
      OCS_DB_PASS: ocs2526
      OCS_DBUSER: ocs
      OCS_DBPASS: ocs2526
      OCS_DB_NAME: ocsweb
      OCS_DBNAME: ocsweb
      OCS_DB_PORT: 3306
      OCS_SSL_ENABLED: 0

volumes:
  db_data:
  glpi_data:

```

Maintenant nous allons démarrer l'orchestration en saisissant la commande :

- **docker compose up -d**

```

root@tepson:/home/tepson/glpi-ocs# docker compose up -d
[+] up 4/4
✓ Container glpi-ocs-bdd-1      Running
✓ Container glpi-ocs-ocsinventory-1 Running
✓ Container glpi-ocs-phpmyadmin-1 Running
✓ Container glpi-ocs-glpi-1     Running
root@tepson:/home/tepson/glpi-ocs#

```

Maintenant vérifions que les conteneurs tournent, pour cela saisissez la commande :

- **docker compose ps**

```

root@tepson:/home/tepson/glpi-ocs# docker compose ps
NAME                IMAGE                COMMAND                SERVICE    CREATED        STATUS        PORTS
glpi-ocs-bdd-1      mysql:latest         "docker-entrypoint.s..." bdd        3 hours ago    Up 3 hours    33060/tcp, 0.0.0.0:3307->3306/tcp, [::]:3307->3306/tcp
glpi-ocs-glpi-1     diouxx/glpi:latest   "/opt/glpi-start.sh"    glpi       3 hours ago    Up 3 hours    443/tcp, 0.0.0.0:8082->80/tcp, [::]:8082->80/tcp
glpi-ocs-phpmyadmin-1 phpmyadmin/phpmyadmin:latest "/docker-entrypoint..." phpmyadmin  3 hours ago    Up 3 hours    0.0.0.0:8085->80/tcp, [::]:8085->80/tcp
ocsinventory        blinzlerone/ocsinventory-docker-image:latest "/bin/bash /root/run..." ocsinventory 4 minutes ago    Up 4 minutes    443/tcp, 0.0.0.0:8083->80/tcp, [::]:8083->80/tcp

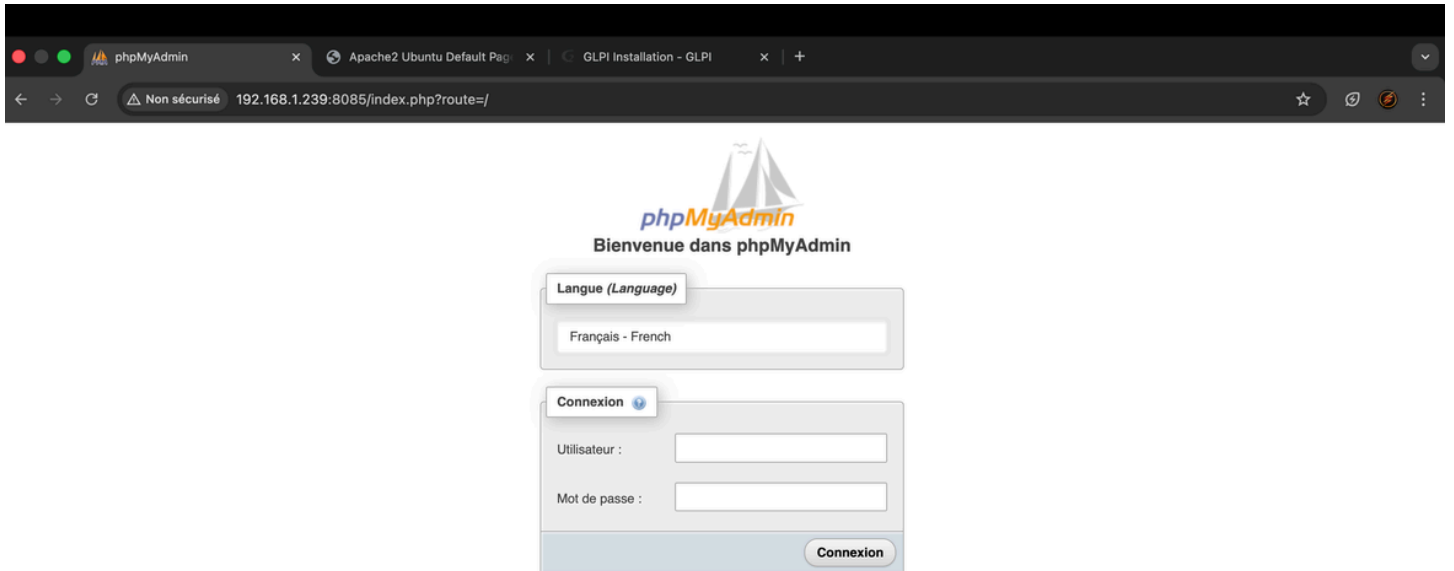
```

Accédons à Votre base de donnée en saisissant dans votre navigateur :

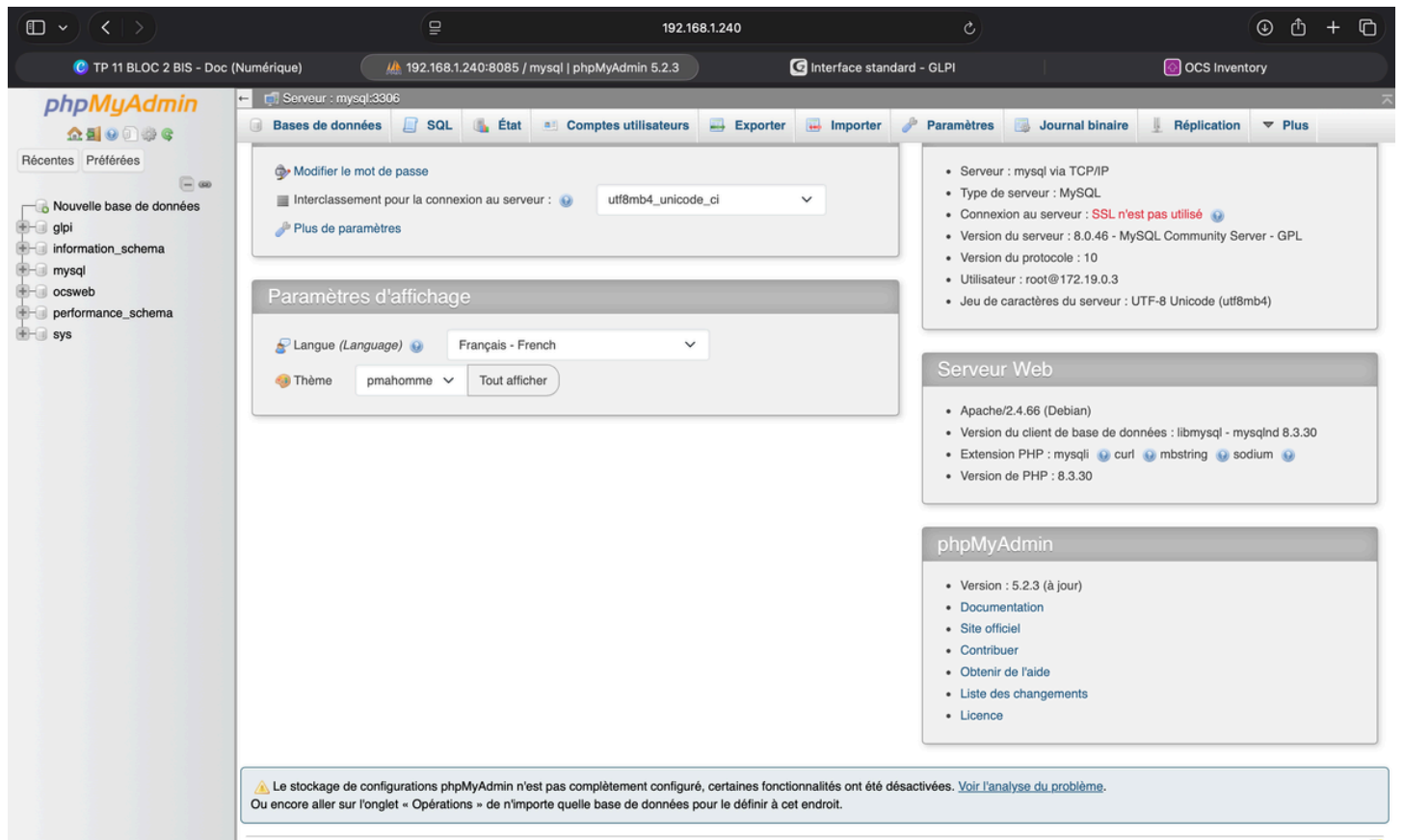
- http://Adresse_IP_SRVDOCKER:8085

8085 port d'écoute pour le service php

pour ma par c'est : <http://192.168.1.239:8085>



Puis connectez vous a votre base de donnée



On peut constater que la base de données glpi est bien présente à droite

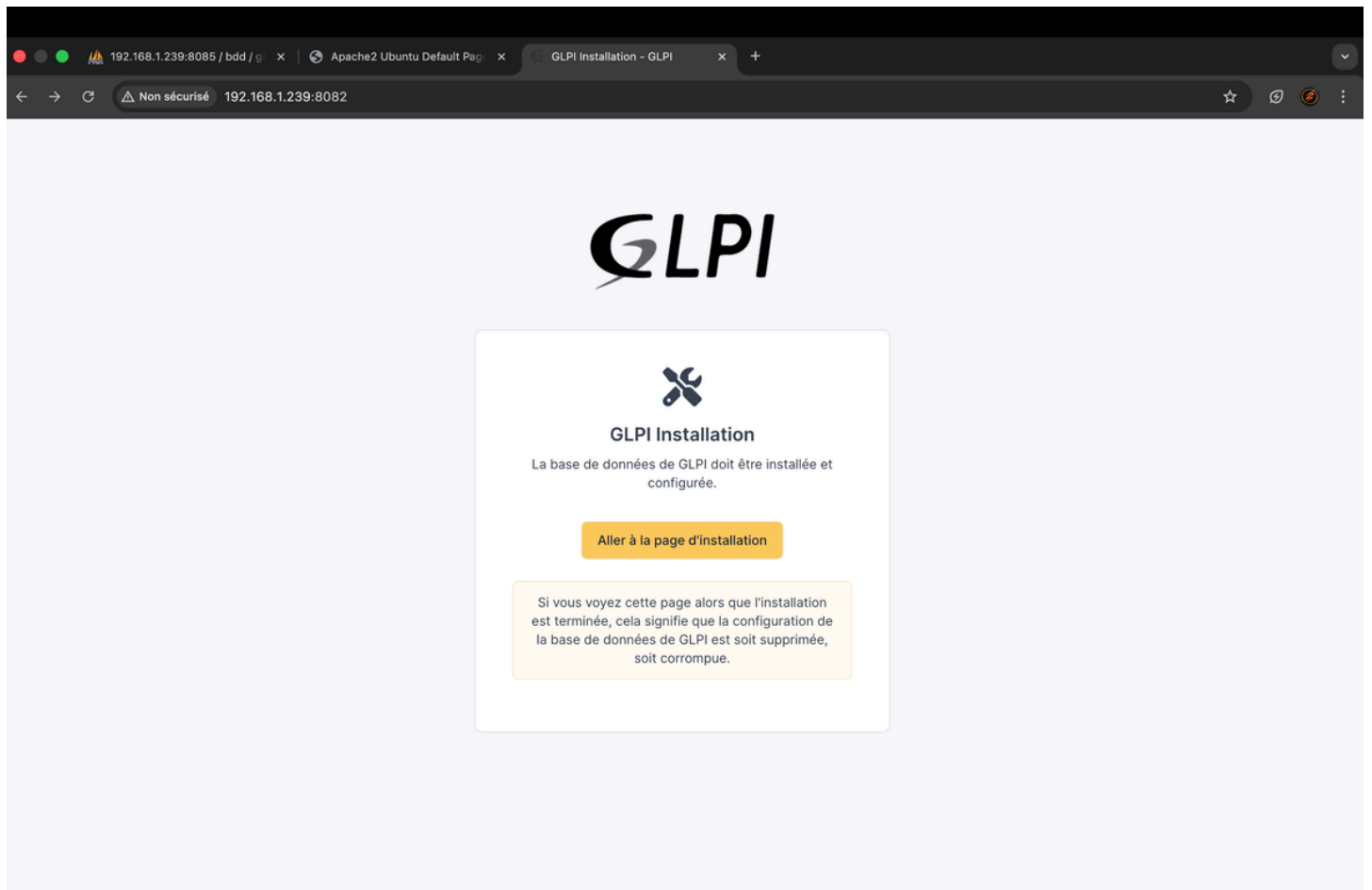
Connexion à glpi. Pour vous connecter à glpi saisissez dans votre navigateur :

- http://Adresse_IP_SRVDOCKER:8082

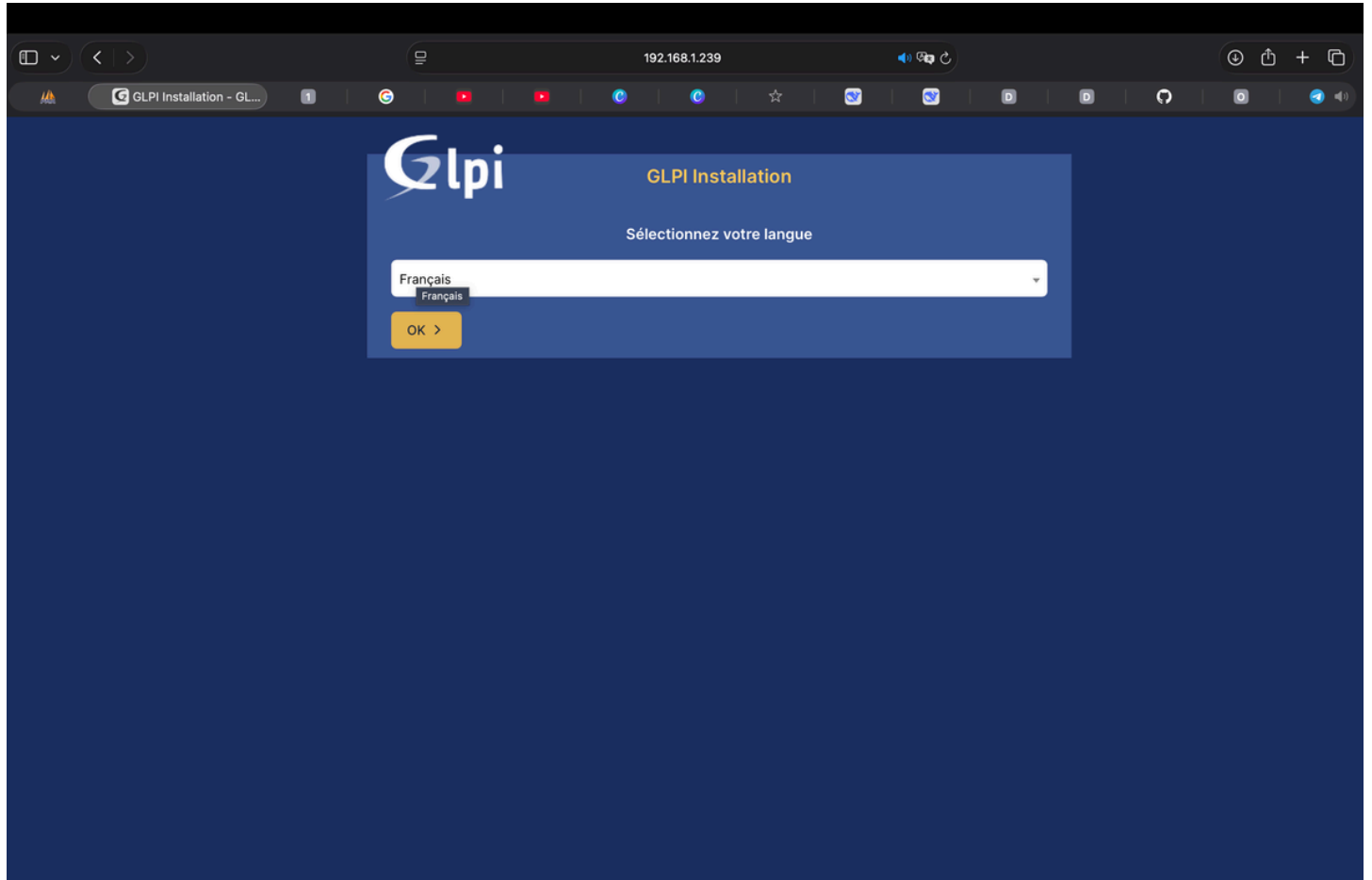
8085 port d'écoute pour glpi

pour moi par c'est : <http://192.168.1.240:8082>

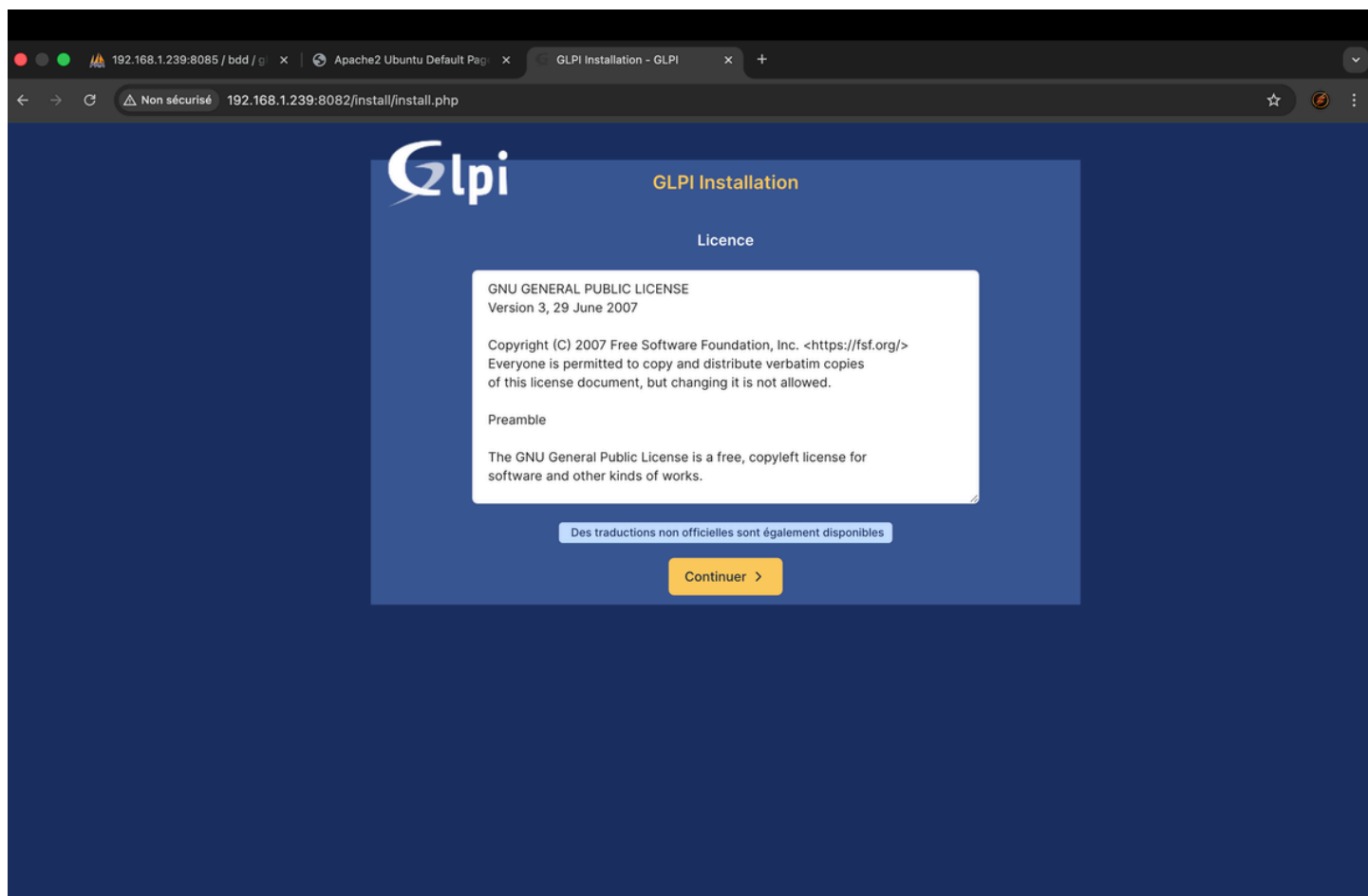
- sur les captures c'est le dernier octet est 239 mais c'est 240 (je suis passé d'une connexion sans fil à filaire)



Cliquez sur aller à la page d'installation



Sélectionner votre langue puis faite ok



Sélectionner continuer



Selectionner l'option installer car ceci est une nouvelle installtion de GLPI

← → ↻ Non sécurisé 192.168.1.239:8082/install/install.php ☆ ⚙



GLPI Installation

Étape 0

Vérification de la compatibilité de votre environnement avec l'exécution de GLPI

TESTS EFFECTUÉS	RÉSULTATS
Requis Parser PHP	✓
Requis Taille d'entier maximal de PHP <i>Le support des entiers 64 bits est nécessaire pour les opérations relatives aux adresses IP (inventaire réseau, filtrage des clients API, ...).</i>	✓
Requis Configuration des sessions	✓
Requis Mémoire allouée	✓
Requis Extensions du noyau de PHP	✓
Requis mysql extension <i>Requis pour l'accès à la base de données.</i>	✓
Requis curl extension <i>Requis pour l'accès à distance aux ressources (requêtes des agents d'inventaire, Marketplace, flux RSS, ...).</i>	✓
Requis gd extension <i>Requis pour le traitement des images.</i>	✓
Requis intl extension <i>Requis pour l'internationalisation.</i>	✓
Requis mbstring extension <i>Requis pour la prise en charge des caractères multioctets et la conversion de jeu de caractères.</i>	✓
Requis zlib extension <i>Requis pour la gestion de la communication compressée avec les agents d'inventaire, l'installation de paquets gzip à partir du Marketplace et la génération de PDF.</i>	✓

Ici Glpi vérifie la compatibilité de notre œuvre, environnement avec celui-ci. Faite continuer

← → ↻ Non sécurisé 192.168.1.239:8082/install/install.php ☆ ⚙



GLPI Installation

Étape 1

Configuration de la connexion à la base de données

Serveur SQL (MariaDB ou MySQL)

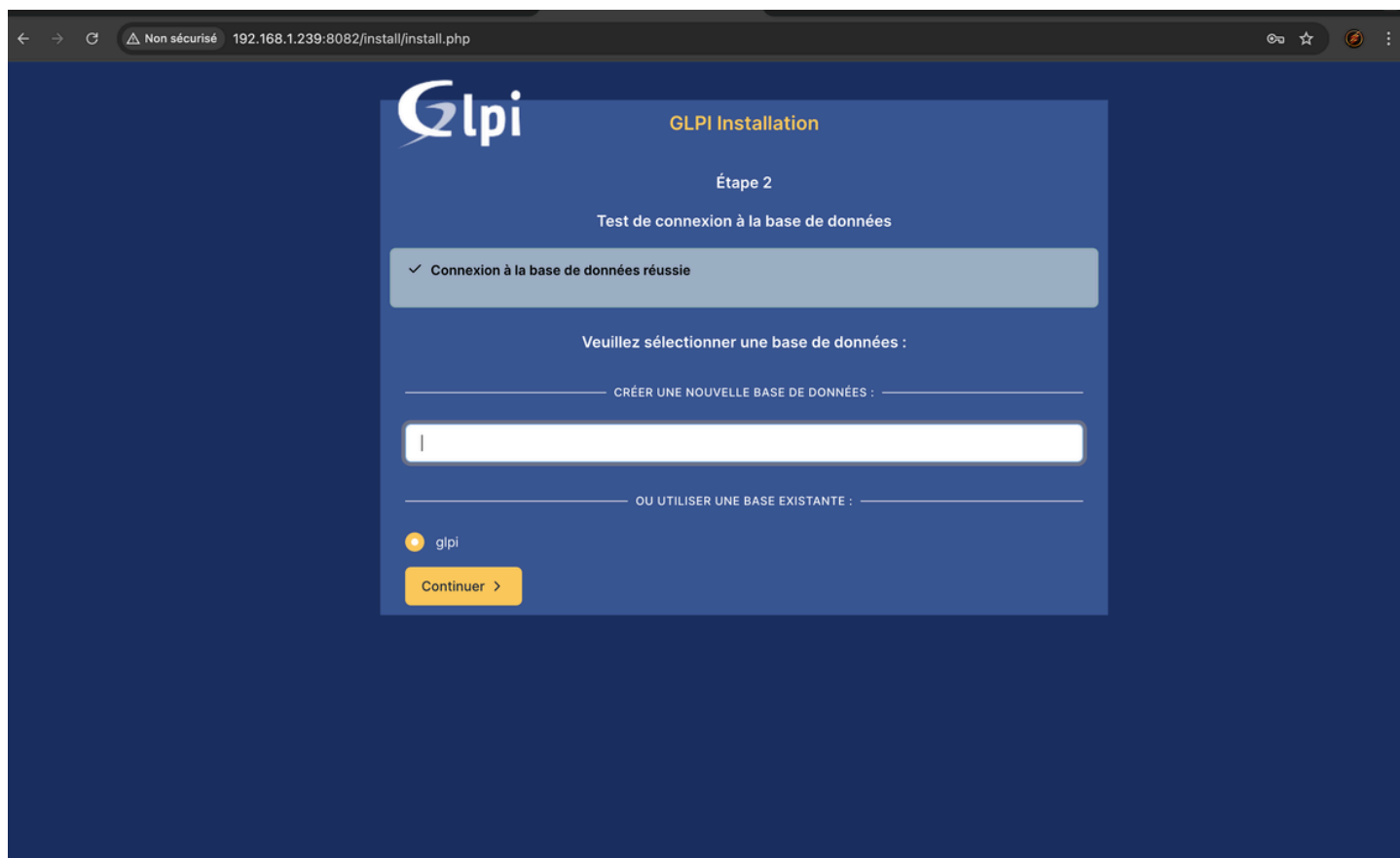
Utilisateur SQL

Mot de passe SQL

[Continuer >](#)

Saisissez les mots de passe et le nom d'utilisateur que nous avons mis pour la base de données :

- Serveur SQL : bdd
- Utilisateur SQL : glpi_user
- Mot de passe SQL : glpi2526



GLPI Installation

Étape 2

Test de connexion à la base de données

✓ Connexion à la base de données réussie

Veillez sélectionner une base de données :

CRÉER UNE NOUVELLE BASE DE DONNÉES :

OU UTILISER UNE BASE EXISTANTE :

glpi

Continuer >

Ont constate que la connexion à la base de données à fonctionné (n'oublier pas de cocher votre base donner au dessus de continuer). maintenant cliquez sur continuer.

Maintenant attendez que l'initialisation à la base donnée ce termine



GLPI Installation

Étape 3

Initialisation de la base de données.

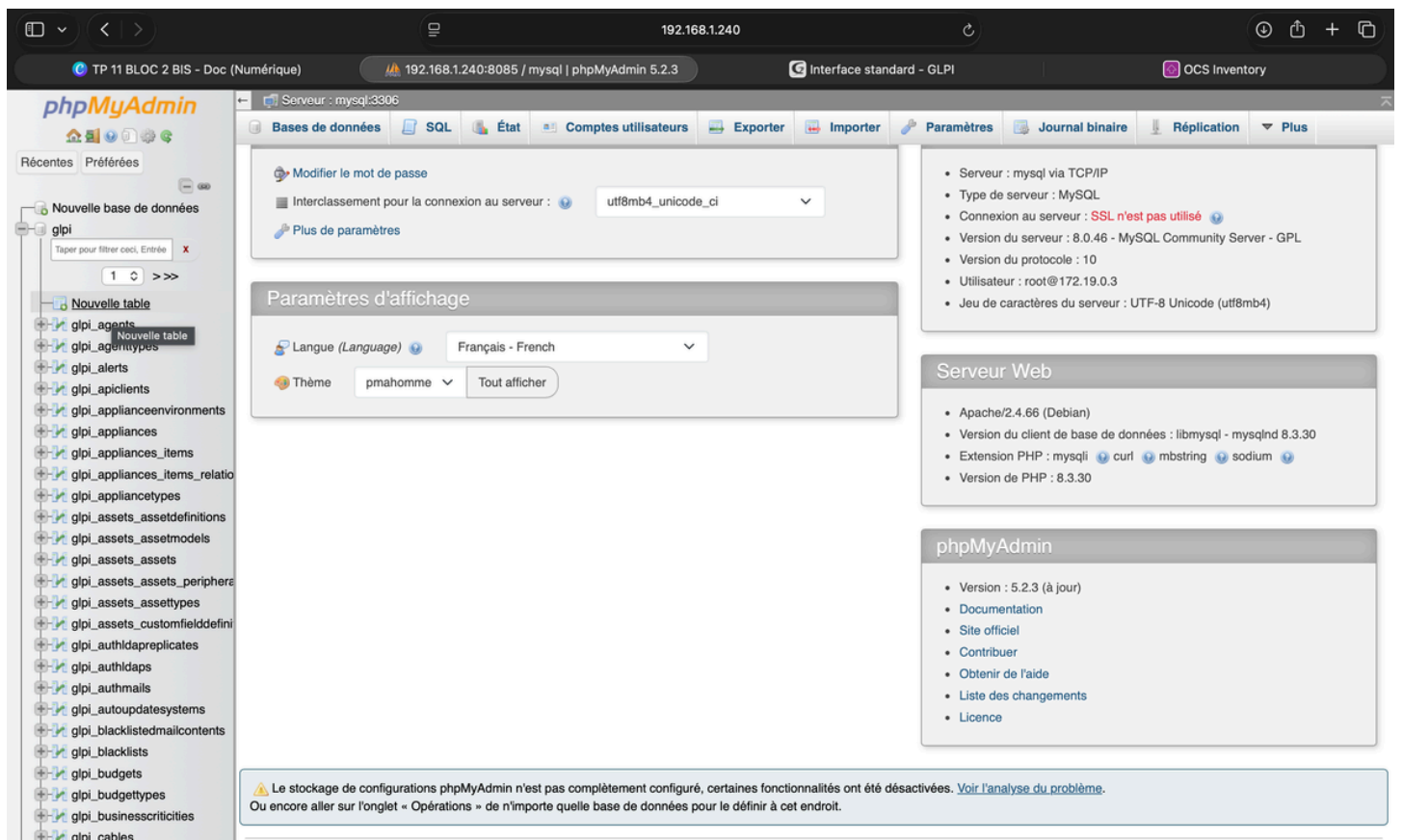
Initialisation des tables de la base de données avec ses données par défaut...

Création de la structure de la base de données...



L'initialisation des tables de la base de données avec ses données à été réalisé avec succès

Avant de cliquer sur continuer, retournons sur PHPMYADMIN



On constate dans la zone de droite que la base de données glpi à initier les tables pour glpi.

Maintenant retournons sur glpi et cliquez sur continuer

TP 11 BLOC 2 BIS - Doc (Numérique) 192.168.1.240:8085 / mysql | phpMyAdmin 5.2.3 Authentification - GLPI OCS Inventory

GLPI

Connexion à votre compte

Identifiant

glpi

Mot de passe

....

Source de connexion

Base interne GLPI

☒ Se souvenir de moi

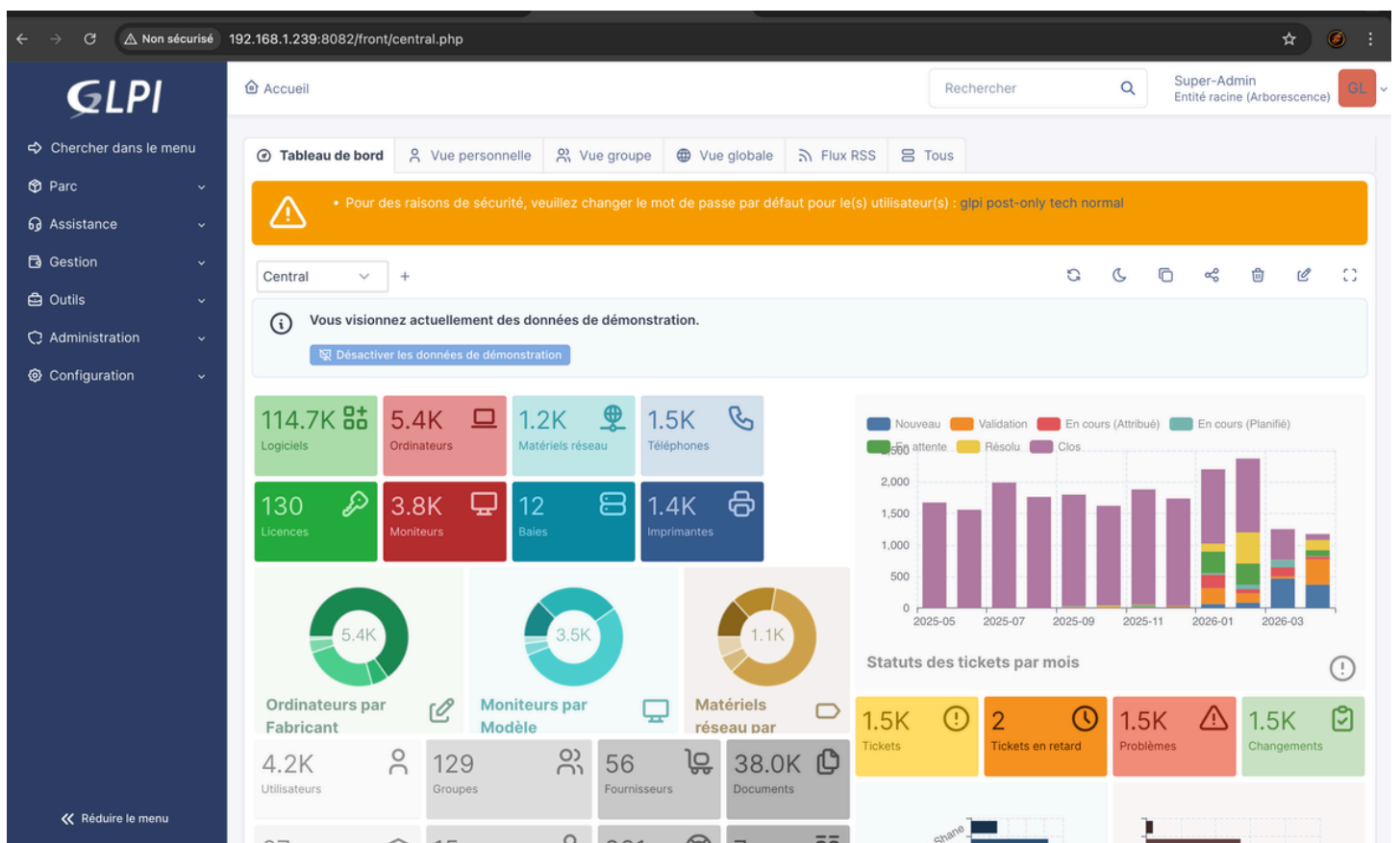
Se connecter

GLPI Copyright (C) 2015-2026 Teclib' and contributors

Connectez-vous avec votre les identifiant par défaut de glpi

Identifiant : glpi

Mot de passe : glpi



Maintenant depuis GLPI nous allons créer les utilisateurs dans GLPI :

NOM – Prénom	Identifiant	Mot de passe	Profil
Mathieu BELLON	mbellon	@mbellon2526@	Admin/technicien
Yohann LEMARC	ylemarc	@ylemarc2526@	Utilisateur
Chantal DUPONT	cdupont	@cdupont2526@	Utilisateur
Christopher LOUIS	clouis	@clouis2526@	Utilisateur

Cliquez sur la partie administration puis utilisateur et cliquez en sur ajouter et créé vos utilisateur. (Le statut d'utilisateur est représenter par Self-Service)

The screenshot displays the GLPI web interface for user management. The main content area shows a table of users with the following data:

ID	IDENTIFIANT	NOM DE FAMILLE	E-MAILS	TÉLÉPHONE	LIEU	ACTIVÉ	DATE DE DÉBUT DE SUBSTITUTION
1	glpi					Oui	
2	post-only					Oui	
3	tech					Oui	
4	normal					Oui	
5	glpi-system	Support				Oui	
6	ylemarc	LEMARC				Oui	
7	mbellon	BELLON				Oui	
8	cdupont	DUPONT				Oui	
9	clouis	LOUIS				Oui	

The interface also includes a sidebar with navigation options and a top navigation bar with the breadcrumb 'Accueil / Administration / Utilisateurs' and an 'Ajouter' button.

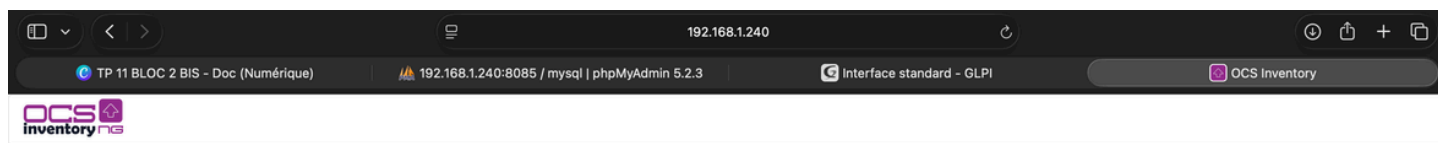
Pour vous connectez à Ocs inventory saisissez dans votre navigateur :

- http://Adresse_IP_SRVDOCKER:8083/ocsreports/

8085 port d'écoute pour glpi

pour ma par c'est : <http://192.168.1.240:8083/ocsreports/>

après connexion a votre base de donné ocs vous tombrez sur cette page



LANGUAGE

 Français

User :

User

Password :

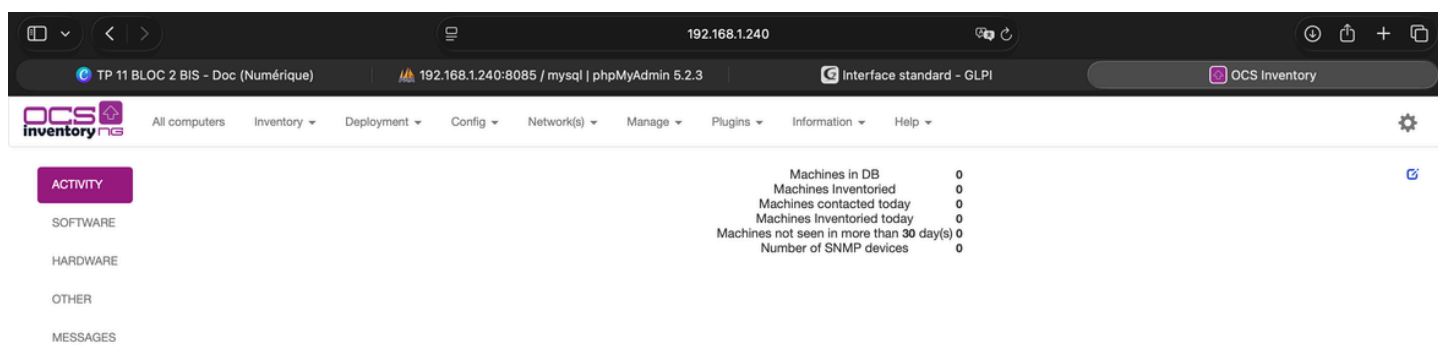
Password

Send

connecter vous avec le profile par défaut de ocs:

User : admin

Password : admin



Et si ont retourne sur phpmyadmin ont constate que la base de donn   ocsweb est rempli

phpMyAdmin

R  centesPr  f  r  es

ocsweb

Taper pour filtrer ceci, Entr  e

1 >>>

Nouvelle table

accesslog

accountinfo

accountinfo_config

batteries

bios

blacklist_macaddresses

blacklist_serials

blacklist_subnets

config

conntrack

controllers

cpus

deleted_equiv

deploy

devices

devicetype

dico_ignored

dico_soft

downloadwk_conf_values

downloadwk_fields

downloadwk_history

downloadwk_pack

downloadwk_statut_request

downloadwk_tab_values

downloadwk_affect_rules

download_available

download_enable

Server : mysql:3306

Bases de donn  esSQL  tatComptes utilisateursExporterImporterParam  tresJournal binaireR  plicationPlus

Modifier le mot de passe

Interclassement pour la connexion au serveur : utf8mb4_unicode_ci

Plus de param  tres

Param  tres d'affichage

Langue (Language)Fran  ais - French

Th  me pmahomme

Tout afficher

Le stockage de configurations phpMyAdmin n'est pas compl  tement configur  , certaines fonctionnalit  s ont   t   d  sactiv  es. Voir l'analyse du probl  me.

Ou encore aller sur l'onglet « Op  rations » de n'importe quelle base de donn  es pour le d  finir    cet endroit.

Console de requ  tes SQL

Server : mysql via TCP/IP

Type de serveur : MySQL

Connexion au serveur : SSL n'est pas utilis  

Version du serveur : 8.0.46 - MySQL Community Server - GPL

Version du protocole : 10

Utilisateur : root@172.19.0.3

Jeu de caract  res du serveur : UTF-8 Unicode (utf8mb4)

Serveur Web

Apache/2.4.66 (Debian)

Version du client de base de donn  es : libmysql - mysqlnd 8.3.30

Extension PHP : mysqli curl mbstring sodium

Version de PHP : 8.3.30

phpMyAdmin

Version : 5.2.3 (   jour)

Documentation

Site officiel

Contribuer

Obtenir de l'aide

Liste des changements

Licence

Nous allons   galement cr  er les utilisateur sur OCS inventory

Utilisateurs - GLPI

Accueil - Canva

TP 11 BLOC 2 BIS - Doc (Num  rique)

OCS Inventory

OCS inventory

All computersInventoryDeploymentConfigNetwork(s)ManagePluginsInformationHelp

Users

Show / Hide : Select columns to show / hide

Search in column : Select All

5 Result(s) (Download)

Show 10 entries

Search :

Select	ID	First name	Last name	Type	Comments	Group	Tags	Actions
☐	admin	admin	admin	sadmin	Default administrator account			
☐	Dupont	Chantal	DUPONT	read-only				
☐	louis	Christopher	LOUIS	read-only				
☐	milbellon	Mathieu	BELLON	admin				
☐	ylemarc	Yohann	LEMARC	read-only				

Showing 1 to 5 of 5 entries

Previous1Next

Pour créer les utilisateur, il vous suffit de cliquez sur l'option config, puis user ensuite create user (Le statut d'utilisateur est représenter par RO "read-only")

CONCLUSION : Ce TP nous a permis de maîtriser Docker Compose pour orchestrer une stack multi-conteneurs (MySQL, phpMyAdmin, WordPress, GLPI, OCS Inventory) dans un contexte professionnel réaliste — celui du CHU de Pointe-à-Pitre. Nous avons vu concrètement l'intérêt du fichier docker-compose.yml : remplacer des dizaines de commandes Docker par une seule, tout en gérant réseaux, volumes et dépendances entre services.

AU REVOIR