Le 17/04/2024

TP Linux

G2R –Formation DevOps

Promotion 11/03 – 20/09/2024

Table des matières

[Contexte 3](#_Toc165687971)

[Installation et configuration de SSH 4](#_Toc165687972)

[Installer la base des données MariaDB 6](#_Toc165687973)

[Installer PHP 8](#_Toc165687974)

[Installation et configuration d'Apache 9](#_Toc165687975)

[Configuration de GLPI avec apache2 11](#_Toc165687976)

[Utilisation de PHP8.3-FPM avec Apache2 12](#_Toc165687977)

[Installation et configuration de vsftpd 13](#_Toc165687978)

[Installation et configuration de GLPI 15](#_Toc165687979)

[Installation et configuration de Zabbix 22](#_Toc165687980)

[Installer Zabbix repository 22](#_Toc165687981)

[Installer Zabbix server, frontend, agent 22](#_Toc165687982)

[Créer une database zabbix 23](#_Toc165687983)

[Profils des utilistauers zabbix 27](#_Toc165687984)

# Contexte

Exercice détaillé pour installer et configurer les services suivants sur une machine Debian :

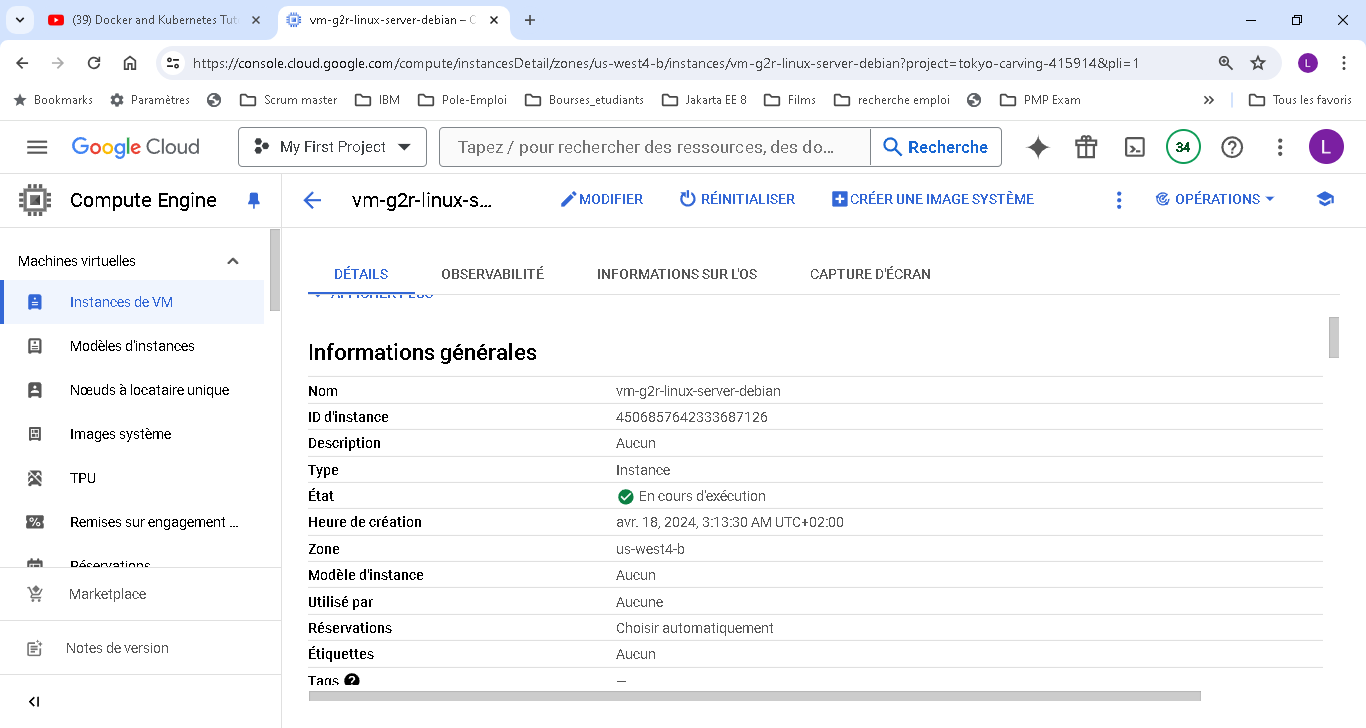
SSH, Apache, vsftpd, GLPI, Zabbix, DHCP et DNS.

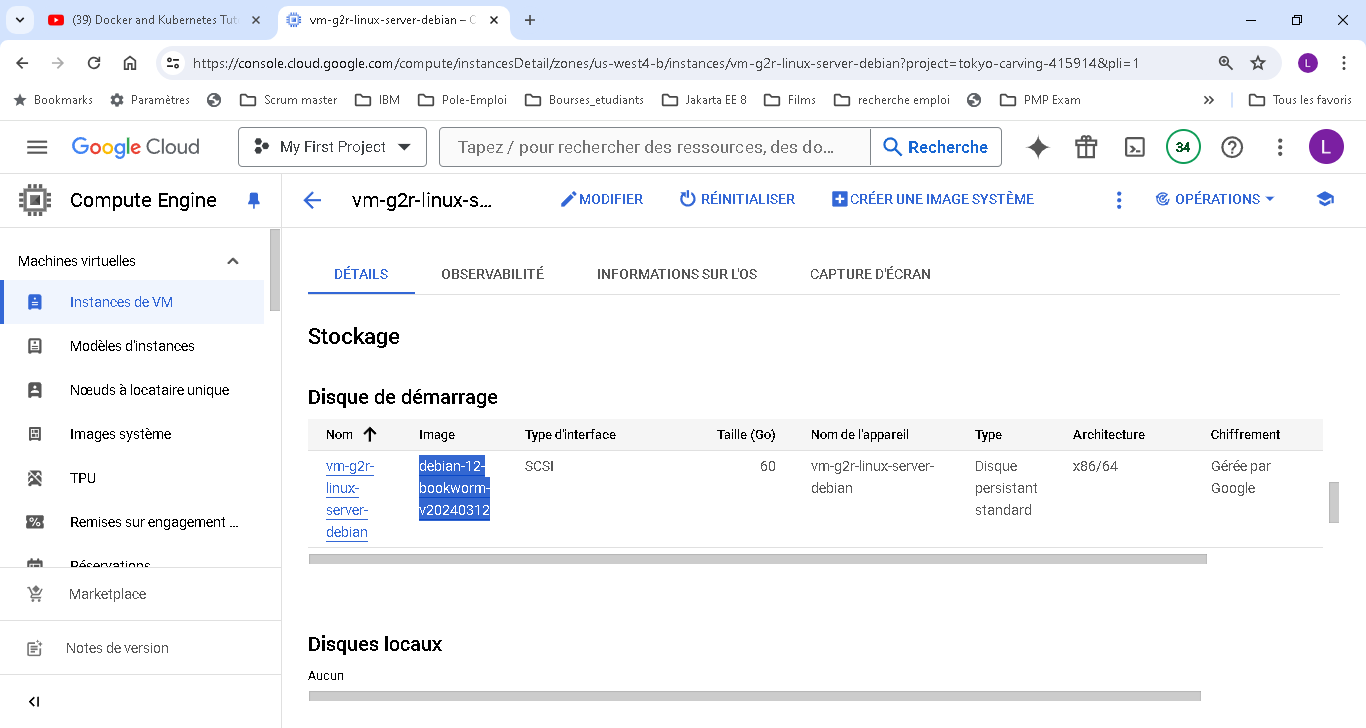
Objectif de l'exercice

Apprendre et se familiariser avec les commandes essentielles et l’exécuter sur une machine Debian.

Suivre les étapes suivantes (TP) pour configurer la machine Debian.

Dans le cadre de ce TP, j’ai installé une VM Debian 12-bookworm-v20240312 sur google cloud





**Les questions du TP**

Installer et configurer les services suivants sur une machine Debian :

1. Serveur SSH pour l'accès distant sécurisé.
2. Serveur Web Apache pour l'hébergement de sites web. Héberger un site web de votre choix ou vous
3. Utiliserez le site web fournit.
4. Serveur FTP (vsftpd) pour le transfert de fichiers.
5. GLPI pour la gestion des actifs informatiques et du service d'assistance.
6. Zabbix pour la surveillance du réseau et des serveurs.
7. Serveur DHCP pour attribuer automatiquement des adresses IP aux clients.
8. Serveur DNS pour la résolution des noms de domaine.

# Installation et configuration de SSH

Installer le serveur SSH en utilisant la commande

sudo apt-get install openssh-server

1. Revert to snapshot

Enregistrer un snapshot pour faciliter le retour arrière et retrouver un environnement propre.

Pour l’exécution des commandes, si vous n’avez pas le droit (permission denied) utiliser sudo ou ajouter votre user dans le groupe sudoers.

1. Se connecter à la machine VM Debian et ouvrir un terminal. Vérifier si SSH est installé, sinon installer le.

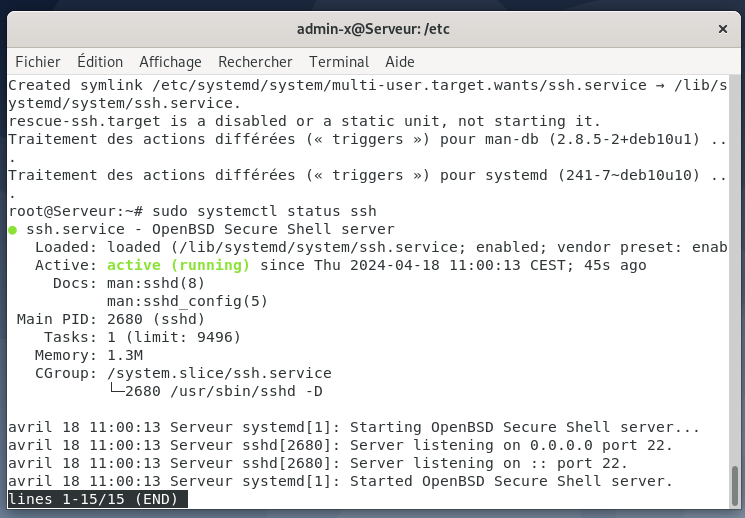
Exécuter la commande suivante :

**apt-get install openssh-server**

Une fois l’installation terminée, vérifier le statut du service SSH en utilisant la commande

**systemctl status ssh**

L’état du service SSH est : **active (running)**



1. Redémarrez un service SSH pour redémarrer et appliquer des modifications

**systemctl restart ssh**



1. Configurer SSH pour une sécurité renforcée en éditant le fichier de configuration

**/etc/ssh/sshd\_config**

Dans ce fichier nous trouverons la configuration de valeurs des paramètres à prendre en compte par le service SSH.

Voici une liste non exhaustive de paramètres qu’on peut trouver et qui peuvent être changer selon le besoin.

Les commentaires expliquent la valeur fixée pour le paramètre.

**Port**: 22 #port par défaut

**PermitRootLogin**: no #interdiction à root de se connecter via ssh

**PasswordAuthentication**: no #Pas besoin d’un password pour se connecter

**AllowUsers**: #prend comme valeur les utilisateurs autorisés

**AllowGroups**: # prend comme valeur les groupes autorisés

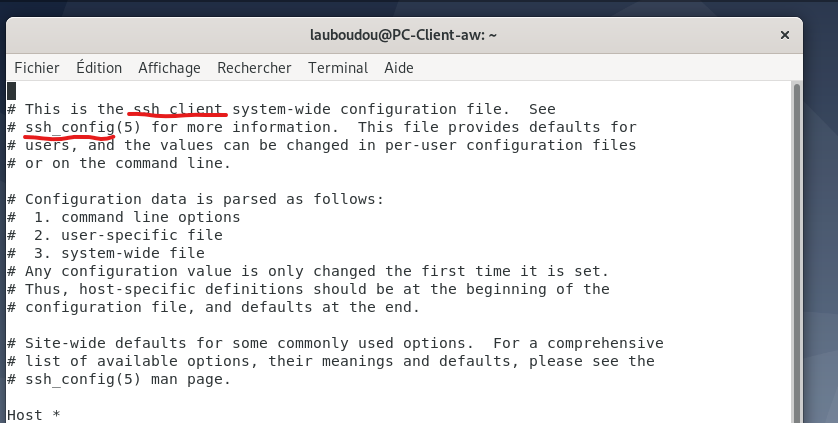
**PubkeyAuthentication**: yes #besoin d’une clé publique pendant l’authentification

**Protocol**: SSH protocol 2 #version du ssh

**PermitEmptyPasswords**: no #n’est pas accepté le password vide

Sur chaque machine VM Debian nous retrouvons une version ssh qui permet de faire le nécessaire en termes d’échange avec le protocole ssh.

Sur un serveur Debian, on installe le service SSH pour gérer l’échange avec une machine cliente.



**Exemple d’utilisation SSH entre serveur et client Debian**

Sur le serveur configurer le fichier **/etc/ssh/sshd\_config** avec les valeurs suivantes :

**Port**: 1904

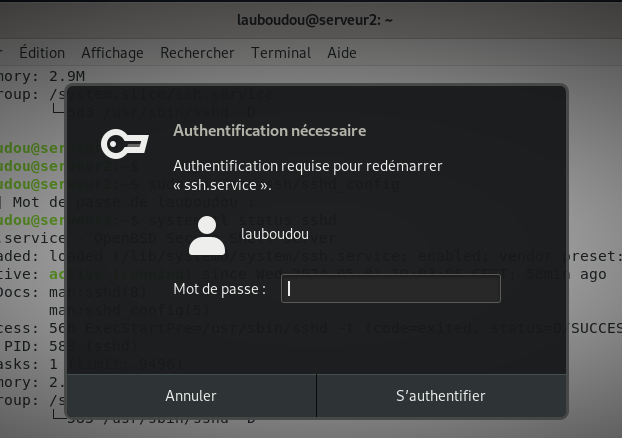
**PermitRootLogin**: yes

**PasswordAuthentication**: no

**PubkeyAuthentication**: yes

**PermitEmptyPasswords**: no

**systemctl restart sshd**



# Installer la base des données MariaDB

MariaDB est une base des données relationnelles et une version modifiée de MySQL. MariaDB a été créé par l'équipe de développement initiale de MySQL en raison de problèmes de licence et de distribution après l'acquisition de MySQL par Oracle Corporation.

Suivre la procédure d’installation de MariaDB sur le site suivant :

<https://www.it-connect.fr/installation-pas-a-pas-de-glpi-10-sur-debian-12/>

**sudo apt-get install apache2 php mariadb-server**

Configurer la base MariaDB, en lançant la commande suivante.

Veuillez à faire les actions suivantes pendant la configuration de la base MariaDB

* changer le mot de passe root,
* supprimer les utilisateurs anonymes,
* désactiver l’accès distant à l’utilisateur root,
* supprimer la base de test et ses tables et privilèges,
* recharger les tables de privilèges

**sudo mysql\_secure\_installation**

Une fois la base installée, nous allons créer une base de données dédiée pour GLPI et celle-ci sera accessible par un utilisateur dédié. Hors de question d'utiliser le compte root de MariaDB : une base de données = un utilisateur.

Connectez-vous à votre instance MariaDB :

**sudo mysql -u root -p**

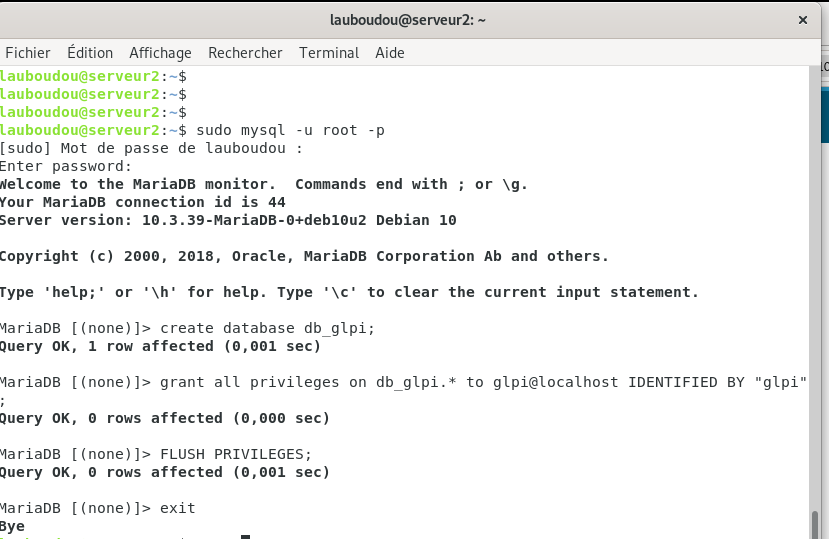
Saisissez le mot de passe root de MariaDB, que vous venez de définir à l'étape précédente.

Puis, nous allons exécuter les requêtes SQL ci-dessous pour créer la base de données "db\_glpi" ainsi que l'utilisateur "glpi" avec le mot de passe "glpi" (que vous changez, bien sûr). Cet utilisateur aura tous les droits sur cette base de données (et uniquement sur celle-ci).

**CREATE DATABASE db\_glpi;**

**GRANT ALL PRIVILEGES ON db\_glpi.\* TO glpi\_adm@localhost IDENTIFIED BY "glpi";**

**FLUSH PRIVILEGES;**



# Installer PHP

Aller à ce site et suivre l’installation de PHP en changeant la version à 8.3 dans les commandes

<https://sys-admin.fr/comment-installer-php-8-1-sur-debian-11-debian-10-debian-9/>

Installe PHP en exécutant cette commande,

**sudo apt update**

**sudo apt upgrade**

Ajout du dépôt :

**sudo apt install ca-certificates apt-transport-https lsb-release**

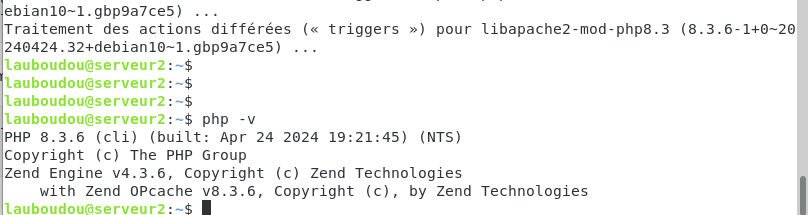
**wget -q https://packages.sury.org/php/apt.gpg -O- | sudo apt-key add -**

**echo "deb https://packages.sury.org/php/ $(lsb\_release -sc) main" | sudo tee /etc/apt/sources.list.d/php.list**

Installation de PHP 8.3

**sudo apt update**

**sudo apt install php8.3**



Installation des modules pour PHP 8.3

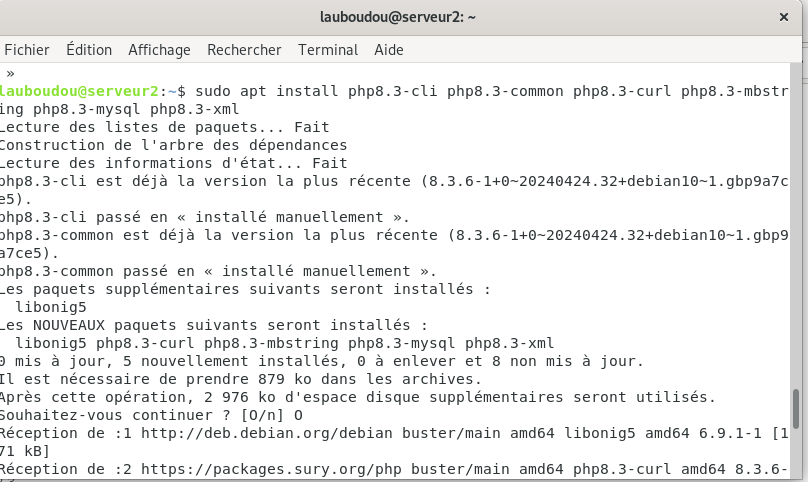
**sudo apt install php8.3-cli php8.3-common php8.3-curl php8.3-mbstring php8.3-mysql php8.3-xml**

Installer les extensions au bon fonctionnement de GLPI

**sudo apt-get install php8.3-json php8.3-gd php8.3-intl php8.3-zip php8.3-bz2 php8.3-imap php8.3-apcu php8.3-gd php8.3-bz2 php8.3-zip**

Si nous souhaitons associer GLPI avec un annuaire LDAP

**sudo apt-get install php8.3-ldap**



Vérifier que PHP est bien installé



PHP peut être installé comme un serveur ou un module à intégrer à Appache.

Le serveur PHP se lance avec la commande suivante

**php -S localhost :5000**

Pour un module php intégré à apache, le service php sera lancé et arrêté avec le service apache2

# Installation et configuration d'Apache

Vérifier qu’apache2 n’est pas installé par la commande suivante,

Si le service apache pas installé, exécuter la commande suivante,

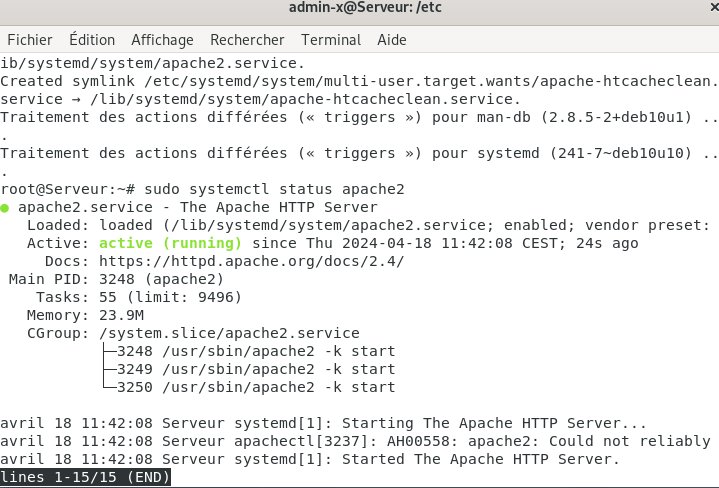
**systemctl status apache2**

Installer apache2 avec la commande suivante,

**apt-get install apache2**

Maintenant vérifier le service apache2

**systemctl status apache2**



1. **Personnaliser la configuration d'Apache**

Selon les besoins, le service apache peut être configuré. Par exemple, configurer des hôtes virtuels, activer des modules, déployer des applications, …etc.

* 1. **Configuration des hôtes virtuels**

Apache utilise des fichiers de configuration séparés pour chaque hôte virtuel. Ces fichiers seront créés dans le répertoire **/etc/apache2/sites-available/**

* 1. **Vérifier les fichier config existant**

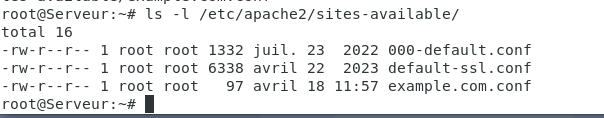
Par exemple, créer un fichier example.com.config pour l’hôte virtuel example.com avec cette configuration

<VirtualHost \*:80>

ServerName example.com

DocumentRoot /var/www/example.com

</VirtualHost>



* 1. **Activation des modules**

A l’aide de ces commandes, on peut activer ou désactiver un module selon le besoin

La commande **a2enmod** pour apache enable mode : active un module

* Exemple : **a2enmod cgi**

La commande **a2dismod** pour apache disable mode : désactive un module

* Exemple : **a2dismod cgi**

1. **Redémarrez Apache**

Pour prendre en compte et appliquer toute modification ou tout simplement démarrer le service, exécuter cette commande,

**systemctl restart apache2**

# Configuration de GLPI avec apache2

**Cette partie est à reprendre une fois le serveur GLPI installé.**

Nous allons créer un nouveau fichier de configuration qui va permettre de configurer le VirtualHost dédié à GLPI. Le fichier s'appelle "vh-glpi.conf".

**sudo nano /etc/apache2/sites-available/vh-glpi.conf**

Ce qui donne la configuration suivante (selon le modèle officiel de la documentation) :

<VirtualHost \*:80>

ServerName vh-gpli

DocumentRoot /var/www/glpi/public

# If you want to place GLPI in a subfolder of your site (e.g. your virtual host is serving multiple applications),

# you can use an Alias directive. If you do this, the DocumentRoot directive MUST NOT target the GLPI directory itself.

# Alias "/glpi" "/var/www/glpi/public"

<Directory /var/www/glpi/public>

Require all granted

RewriteEngine On

# Redirect all requests to GLPI router, unless file exists.

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteRule ^(.\*)$ index.php [QSA,L]

</Directory>

</VirtualHost>

Quand la configuration est prête, enregistrez le fichier.

Puis, nous allons activer ce nouveau site dans Apache2 :

**sudo a2ensite support.it-connect.tech.conf**

Nous en profitons également pour désactiver le site par défaut car il est inutile :

**sudo a2dissite 000-default.conf**

Nous allons aussi activer le module "rewrite" (pour les règles de réécriture) car on l'a utilisé dans le fichier de configuration du VirtualHost (RewriteCond / RewriteRule).

**sudo a2enmod rewrite**

Il ne reste plus qu'à redémarrer le service Apache2 :

**sudo systemctl restart apache2**

# Utilisation de PHP8.3-FPM avec Apache2

**Cette partie est à reprendre une fois PHPinstallé.**

Pour utiliser PHP en tant que moteur de scripts avec Apache2, il y a deux possibilités : **utiliser le module PHP pour Apache2 (libapache2-mod-php8.2) ou utiliser PHP-FPM**.

Il est **recommandé d'utiliser PHP-FPM** car il est plus performant et se présente comme un service indépendant. Dans l'autre mode, chaque processus Apache2 exécute son propre moteur de scripts PHP.

Si vous souhaitez utiliser PHP-FPM, suivez les étapes ci-dessous. Sinon, passez à la suite mais veillez à **configurer l'option "session.cookie\_httponly"** évoquée ci-dessous.

Nous allons commencer par**installer PHP8.3-FPM** avec la commande suivante :

**sudo apt-get install php8.2-fpm**

Puis, nous allons activer deux modules dans Apache et la configuration de PHP-FPM, avant de recharger Apache2 :

**sudo a2enmod proxy\_fcgi setenvif**

**sudo a2enconf php8.3-fpm**

**sudo systemctl reload apache2**

Pour **configurer PHP-FPM pour Apache2**, nous n'allons pas éditer le fichier **/etc/php/8.3/apache2/php.ini** mais **plutôt**ce fichier :

**sudo nano /etc/php/8.3/fpm/php.ini**

Dans ce fichier, recherchez l'option **session.cookie\_httponly** et indiquez la valeur on pour l'activer, afin de protéger les cookies de GLPI.

; Whether or not to add the httpOnly flag to the cookie, which makes it  
; inaccessible to browser scripting languages such as JavaScript.  
; https://php.net/session.cookie-httponly  
session.cookie\_httponly = on

Enregistrez le fichier quand c'est fait. Par la suite, vous pourriez être amené à effectuer d'autres modifications, notamment pour augmenter la taille des uploads sur GLPI, etc.

Pour appliquer les modifications, nous devons redémarrer PHP-FPM :

**sudo systemctl restart php8.3-fpm.service**

Pour finir, nous devons **modifier notre VirtualHost** pour préciser à Apache2 que PHP-FPM doit être utilisé pour les fichiers PHP :

**sudo nano /etc/apache2/sites-available/vh-glpi.conf**

<FilesMatch \.php$>

SetHandler "proxy:unix:/run/php/php8.3-fpm.sock|fcgi://localhost/"

</FilesMatch>



Quand c'est fait, relancer Apache2 :

**sudo systemctl restart apache2**

# Installation et configuration de vsftpd

Pour installation du service vsftp (Very Secure FTP Daemon) qui permet le transfert de fichiers entre une machine Debian et un autre hôte, exécuter la commande suivante,

Vérifier si le service vsftpd existe,

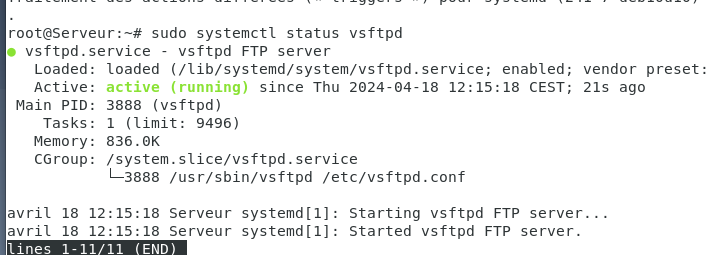
**systemctl status apache2**

Si le service n’existe pas, exécuter la commande,

**apt-get install vsftpd**

Vérifier l’état du service vsftpd

**sudo systemctl status vsftpd**



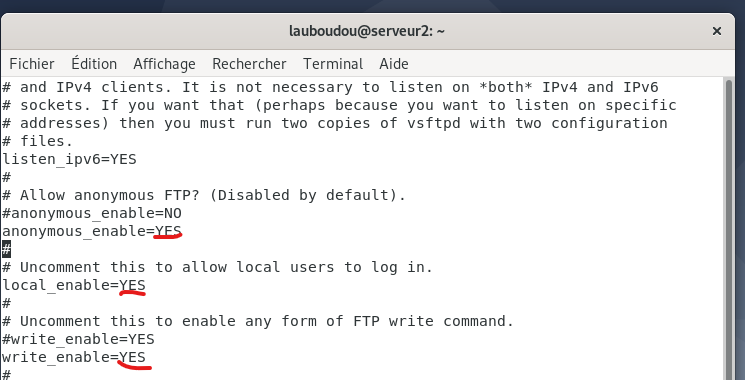
1. **Configurer vsftpd**

Le service vsftpd peut être configuré, par exemple, pour restreindre l'accès aux utilisateurs autorisés, désactiver l'accès anonyme, etc.

Aller dans le fichier **/etc/vsftpd.conf**

Restreindre l'accès qu’aux utilisateurs autorisés 🡺 vérifier que la valeur de directive local\_enable=YES

Désactiver l'accès anonyme 🡺 vérifier que la valeur de directive anonymous\_enable = no



On peut spécifier le répertoire racine pour les utilisateurs locaux en utilisant la directive chroot\_local\_user. Cela confinera les utilisateurs dans leur répertoire personnel.

Affecter à la directive chroot\_local\_user = YES

Redémarrez vsftpd pour appliquer les modifications que nous avons configurées

**systemctl restart vsftpd**



# Installation et configuration de GLPI

GLPI est un logiciel libre de gestion des services informatiques (ITSM) et de gestion des services d'assistance (issue tracking system et ServiceDesk). Il est édité en PHP et distribué sous licence GPL.

Télécharger les fichiers d'installation de GLPI à partir du site officiel

<https://glpi-project.org/fr/telecharger-glpi/>

1. **Installation GLPI**

Dans le cadre de cette installation et pour le bon fonctionnement du serveur GLPI nous avons besoin de s’assurer que les outils nécessaires au bon fonctionnement de GLPI sont installés et sont dans des versions compatibles avec GLPI.

Consulter la page pour connaitre la compatibilité : <http://readthedocs.org/projects/glpi-install-fr/downloads/pdf/latest/>

**Les outils nécessaires et les versions à installer sont :**

**GLPI version 10.0.10**

**PHP 8.3.6 🡺 installé**

**Apache2 🡺 installé**

**Mariadb 🡺 installé**

Se rendre à la page suivante et suivre la procédure d’installation

Installation : <https://www.it-connect.fr/installation-pas-a-pas-de-glpi-10-sur-debian-12/>

**Télécharger GLPI 10.0.10 en exécutant la commande suivante,**

**wget** [**https://github.com/glpi-project/glpi/releases/download/10.0.10/glpi-10.0.10.tgz**](https://github.com/glpi-project/glpi/releases/download/10.0.10/glpi-10.0.10.tgz)

Supprimer l’archive téléchargée

D’abord déplacer le dossier glpi dans le répertoire /var/www/html

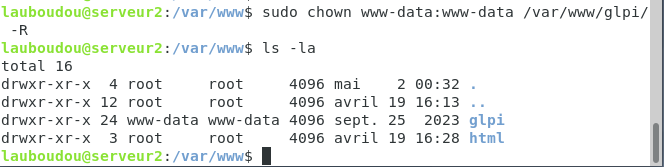
**mv /home/lauboudou/Téléchargements/glpi/var/www/html**

Puis, nous allons exécuter la commande ci-dessous pour décompresser l'archive .tgz dans le répertoire /var/www/, ce qui donnera le chemin d'accès /var/www/glpi pour GLPI.

**sudo tar -xzvf glpi-10.0.10.tgz -C /var/www/**

Nous allons définir l'utilisateur www-data correspondant à Apache2, en tant que propriétaire sur les fichiers GLPI.

**chown -R** [**www.data:www.data**](http://www.data:www.data) **/var/www/glpi**



Ensuite, nous allons devoir créer plusieurs dossiers et sortir des données de la racine Web du répertoire /var/www/glpi de manière à les stocker dans les nouveaux dossiers que nous allons créer.

Ceci va permettre de faire une installation sécurisée de GLPI, qui suit les recommandations de l'éditeur.

* Le répertoire /etc/glpi

Commencez par créer le répertoire /etc/glpi qui va recevoir les fichiers de configuration de GLPI. Nous donnons des autorisations à www-data sur ce répertoire car il a besoin de pouvoir y accéder.

**sudo mkdir /etc/glpi**

**sudo chown www-data /etc/glpi/**

Puis, nous allons déplacer le répertoire "config" de GLPI vers ce nouveau dossier :

**sudo mv /var/www/glpi/config /etc/glpi**

* Le répertoire /var/lib/glpi

Répétons la même opération avec la création du répertoire /var/lib/glpi :

**sudo mkdir /var/lib/glpi**

**sudo chown www-data /var/lib/glpi/**

Dans lequel nous déplaçons également le dossier "files" qui contient la majorité des fichiers de GLPI : CSS, plugins, etc.

**sudo mv /var/www/glpi/files /var/lib/glpi**

* Le répertoire /var/log/glpi

Terminons par la création du répertoire /var/log/glpi destiné à stocker les journaux de GLPI. Toujours sur le même principe :

**sudo mkdir /var/log/glpi**

**sudo chown www-data /var/log/glpi**

Nous n'avons rien à déplacer dans ce répertoire.

* Créer les fichiers de configuration

Nous devons configurer GLPI pour qu'il sache où aller chercher les données. Autrement dit, nous allons déclarer les nouveaux répertoires fraichement créés.

Nous allons créer ce premier fichier :

**sudo nano /var/www/glpi/inc/downstream.php**

Afin d'ajouter le contenu ci-dessous qui indique le chemin vers le répertoire de configuration :

<?php

define('GLPI\_CONFIG\_DIR', '/etc/glpi/');

if (file\_exists(GLPI\_CONFIG\_DIR . '/local\_define.php')) {

require\_once GLPI\_CONFIG\_DIR . '/local\_define.php';

}

Ensuite, nous allons créer ce second fichier :

**sudo nano /etc/glpi/local\_define.php**

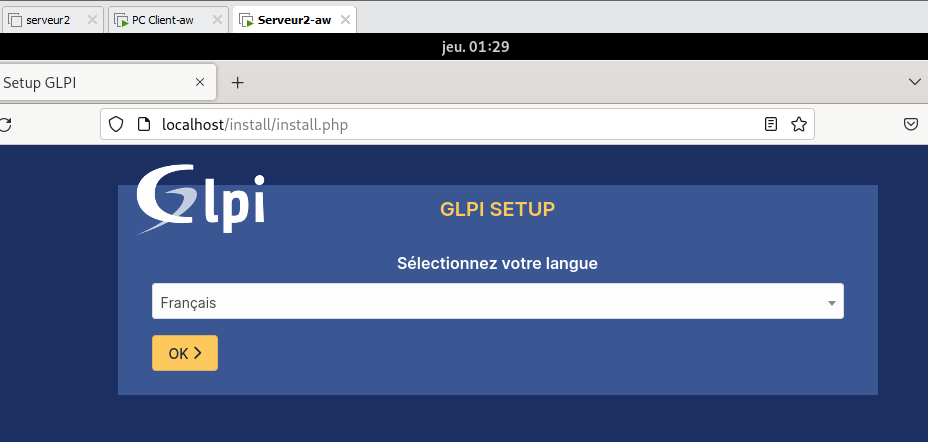
Afin d'ajouter le contenu ci-dessous permettant de déclarer deux variables permettant de préciser les chemins vers les répertoires "files" et "log" que l'on a préparé précédemment.

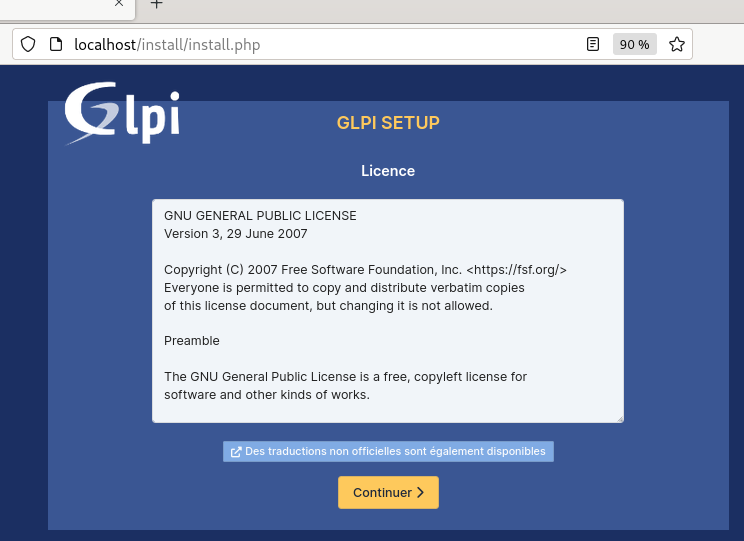
<?php

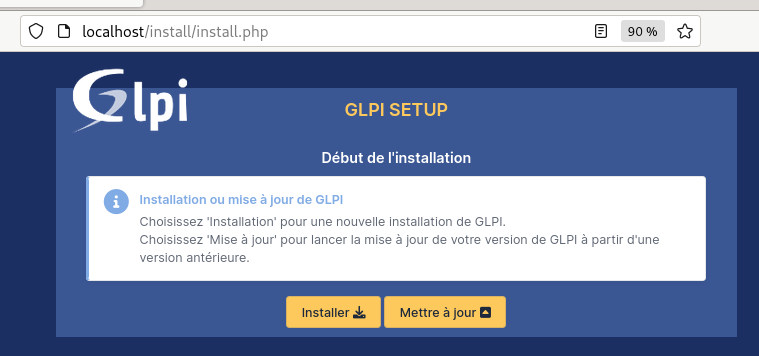
define('GLPI\_VAR\_DIR', '/var/lib/glpi/files');

define('GLPI\_LOG\_DIR', '/var/log/glpi');

Maintenant que la configuration de GLPI, Apapche2, PHP est terminée. Lancer GLPI pour finir l’installation.

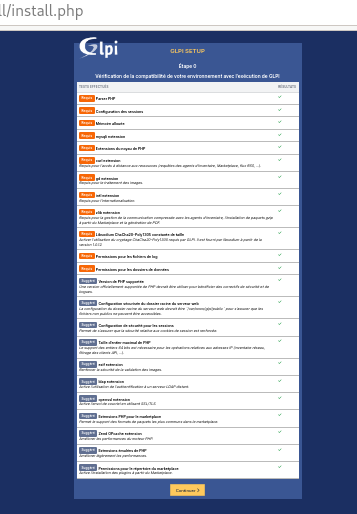






Cliquer sur installer

Si des extensions manquent, veuillez à fixer le problème. Une fois fixer, cliquer sur ressayer



Configurer la base des données créée pour glpi

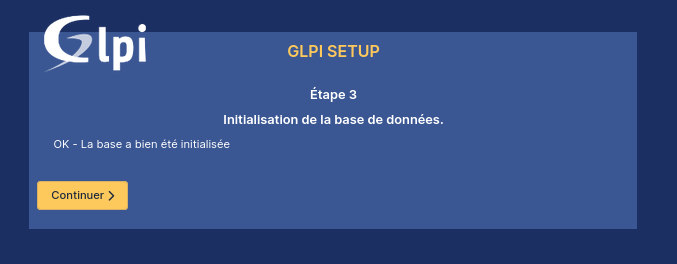
Serveur : localhost compte glpi / glpi



La connexion est établie. Sélectionner la base db\_glpi



La base est bien initialisée







L’installation est terminée. Cliquer sur Utiliser GLPI

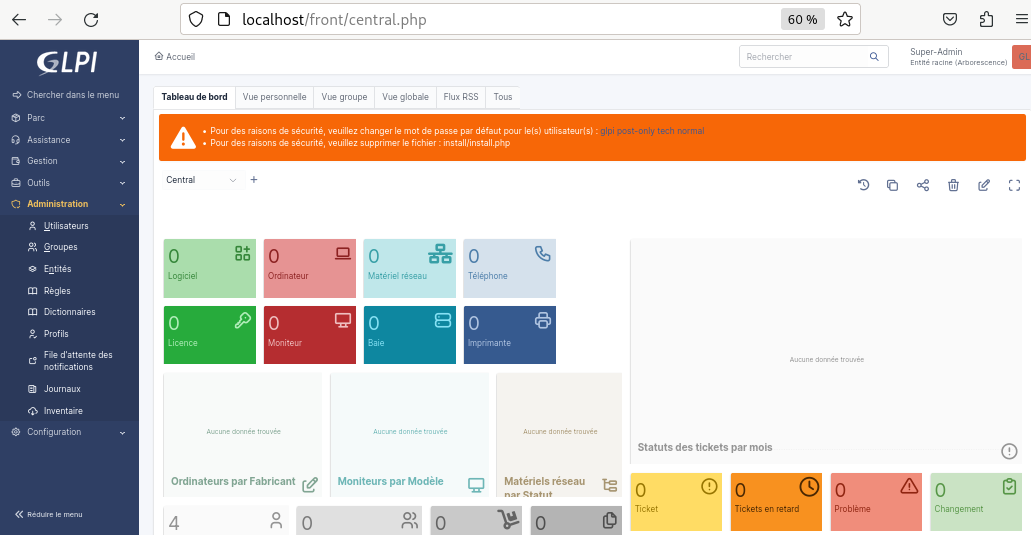


Se connecter avec le compte glpi / glpi



Maintenant nous accédons au site GLPI.

Pour des raisons de sécurité, les warning affiché nous demande de supprimer le fichier install/install.php pour éviter de réinstaller le serveur glpi et de changer le mot de passe du compte glpi à la première connexion au serveur.



# Installation et configuration de Zabbix

Zabbix est un logiciel qui supervise de nombreux paramètres réseaux ainsi que la santé et l'intégrité des serveurs. Zabbix utilise un mécanisme de notification flexible qui permet aux utilisateurs de configurer une base d'alerte e-mail pour pratiquement tous les événements. Cela permet une réponse rapide aux problèmes serveurs. Zabbix offre un excellent reporting et des fonctionnalités de visualisation de données basées sur les données stockées.

<https://www.zabbix.com/download?zabbix=6.4&os_distribution=debian&os_version=12&components=server_frontend_agent&db=mysql&ws=apache>

## Installer Zabbix repository

Exécuter les commandes suivantes pour récupérer le package zabbix

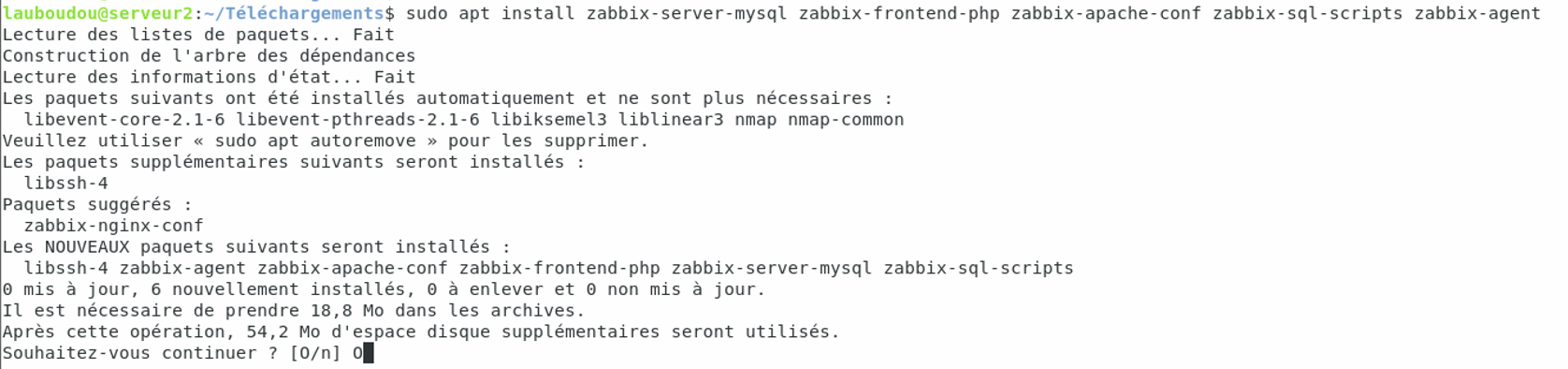
**Wget https://repo.zabbix.com/zabbix/6.0/debian/pool/main/z/zabbix-release/zabbix-release\_6.0-4+debian10\_all.deb**

**sudo dpkg -i zabbix-release\_6.0-4+debian10\_all.deb**

**sudo apt update**

## Installer Zabbix server, frontend, agent

**sudo apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent**



## Créer une database zabbix

Vérifier que la base MariaDB est installée.

Renseigner le mot de passe de l’utilisateur root de la base mariaDB installée précédemment

# mysql -u root -p

Password

mysql> create database zabbix character set utf8mb4 collate utf8mb4\_bin;

mysql> create user zabbix@localhost identified by 'zabbix';

mysql> grant all privileges on zabbix.\* to zabbix@localhost;

mysql> set global log\_bin\_trust\_function\_creators = 1;

mysql> quit;

Importer le schéma initial et les données. Entrer le mot de passe de l’utilisateur zabbix.

Exécuter cette commande, le mot de passe de l’utilisateur zabbix sera demandé

**cat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -uzabbix**

Désactiver log\_bin\_trust\_function\_creators option après l’impotation du schéma dans la database schéma.

# mysql -uroot -p

password

mysql> set global log\_bin\_trust\_function\_creators = 0;

mysql> quit;

Configurer la database de Zabbix server

Edit file **/etc/zabbix/zabbix\_server.conf**

DBPassword=password

Démarrer le serveur Zabbix et l’agent de processes

**systemctl restart zabbix-server zabbix-agent apache2**

Si vous rencontrez de problèmes genre unit : zabbix-server.service is masked désactiver le mask en exécutant la commande suivante :

**sudo systemctl unmask --now zabbix-server.service**

Refaire la commande

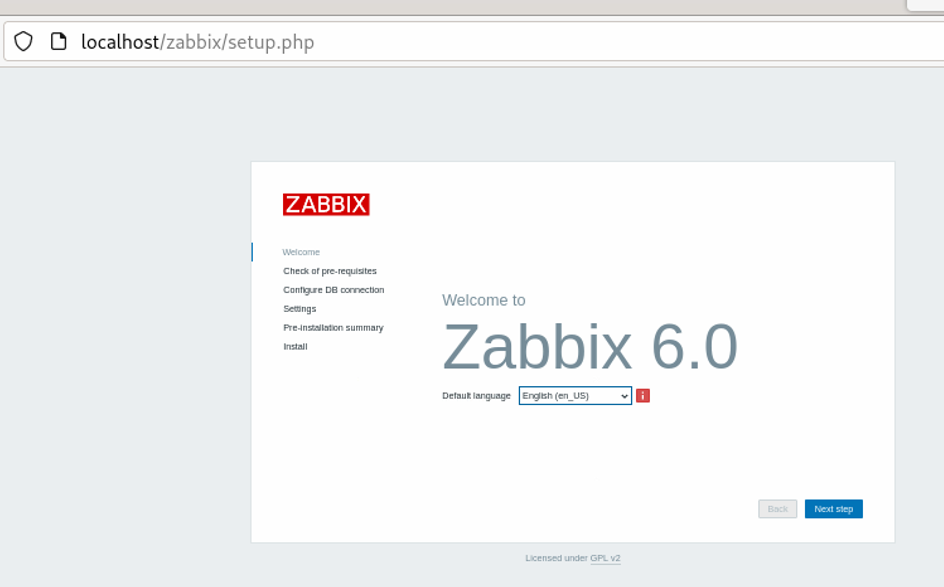
**Sudo systemctl restart zabbix-server zabbix-agent apache2**

Rendre Zabbix disponible

**Sudo systemctl enable zabbix-server zabbix-agent apache2**

Aller sur un navigateur et se connecter à zabbix

<http://localhost/zabbix>

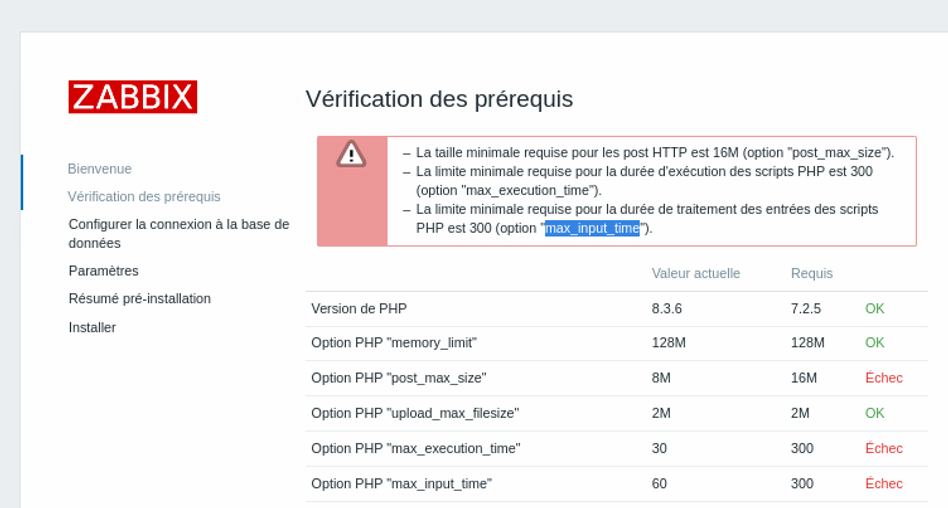


Cliquer sur next step

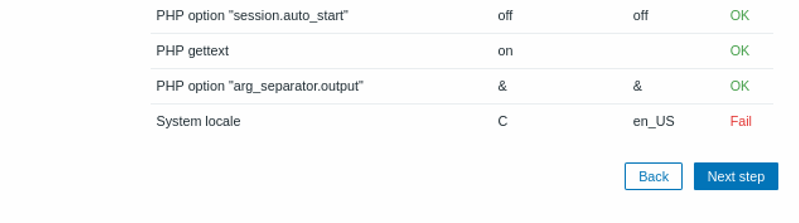
Vérifier que toute la configuration est OK.

**Note** : après chaque modification pour régler un problème de configuration détecté par zabbix, n’oublier pas de redémarrer zabbix et apache2

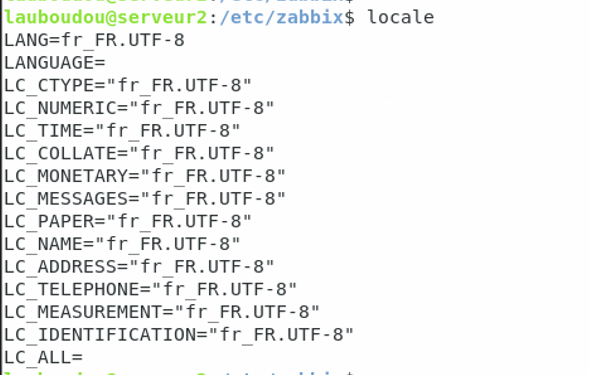
Les erreurs que nous pouvons rencontrer pendant la configuration de zabbix via son interface :



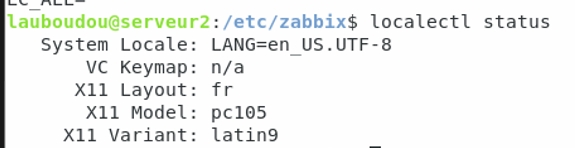
* System local Fail



Sur un terminal taper la commande locale pour voir les paramètres de la langue.



La commande localctl status affiche la langue sur le système,



Pour changer la langue



* Problèmes sur les paramètres PHP, consulter le fichier php.ini

Lancer la commande php –ini

Pour récupérer les fichiers php.ini exstant.

Zabbix utilise le fichier front php /etc/8.3/fpm/php.ini

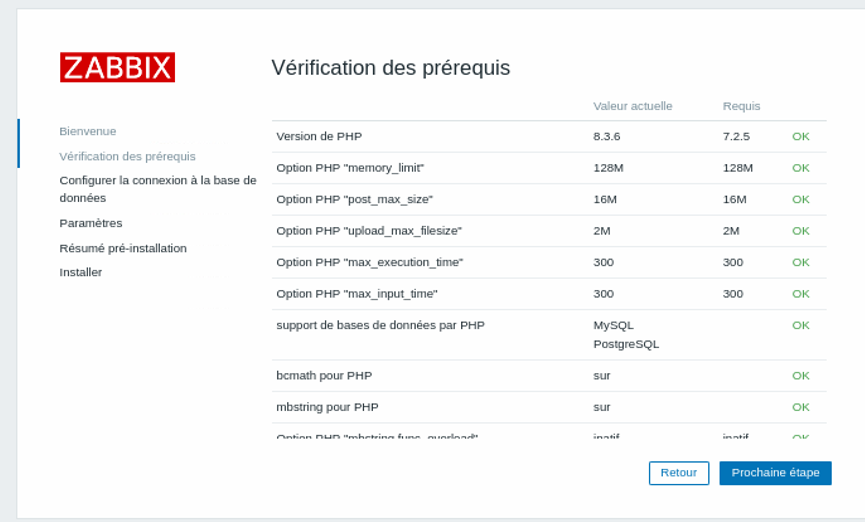
Il contient la configuration des paramètres PHP que zabbix affiche en erreur par exemlpe. Modifier les paramètres en question pour répondre aux attentes de zabbix

Redémarrer

/etc/init.d/php8.3-fpm restart

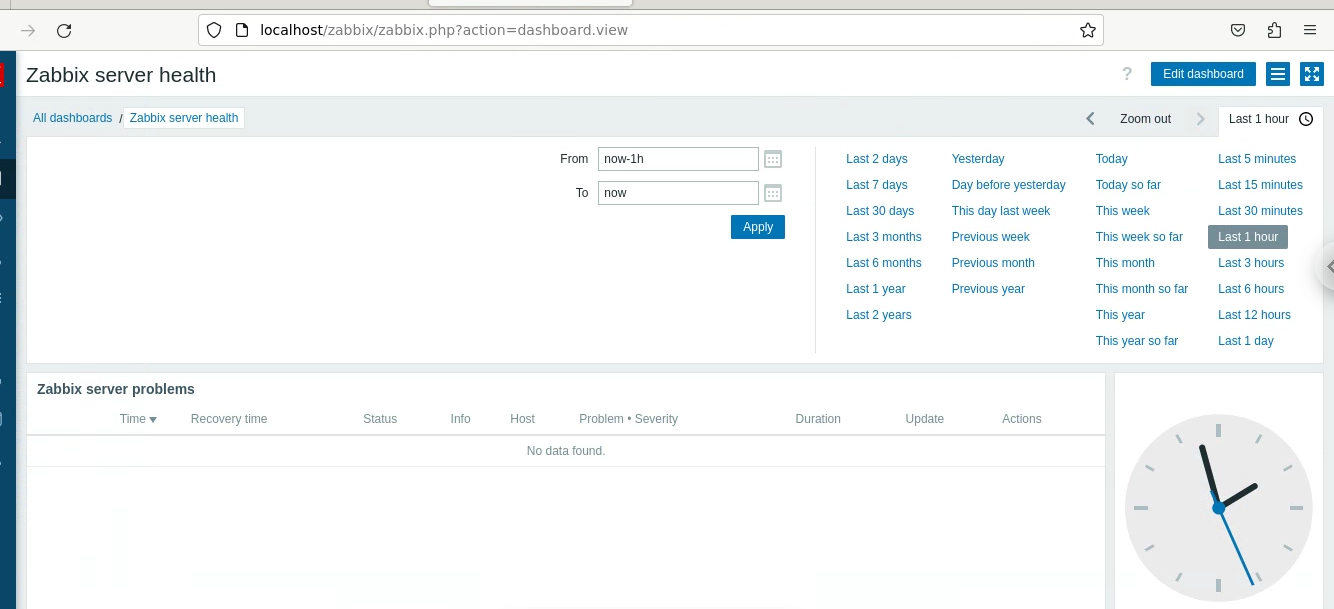
Redémarrer zabbix et apache2

Retester l’interface zabbix





Le serveur zabbix est installé



## Profils des utilistauers zabbix

Documentations : <https://www.zabbix.com/documentation/6.4/en/manual/quickstart/login>

* Default username/password is Admin/zabbix.
* Guest/ mot de passe = vide

Ajoutez le référentiel Zabbix à votre système Debian.

Pour ajouter le référentiel Zabbix au système Debian, suivez ces étapes :

1. Ajouter la clé GPG du référentiel Zabbix :

Cette étape permet à votre système de vérifier l'authenticité des paquets provenant du référentiel Zabbix.

wget https://repo.zabbix.com/zabbix-official-repo.key

sudo apt-key add zabbix-official-repo.key

2. Ajouter le référentiel Zabbix à la liste des sources de paquets APT :

Utilisez le script add-apt-repository pour ajouter le référentiel Zabbix. Si vous n'avez pas ce script installé, vous pouvez l'installer en installant le paquet software-properties-common :

sudo apt-get update

sudo apt-get install software-properties-common

Ensuite, ajoutez le référentiel Zabbix :

sudo add-apt-repository "deb https://repo.zabbix.com/zabbix/6.4.12/debian $(lsb\_release -sc) main"

Assurez-vous de remplacer 5.0 dans l'URL du référentiel par la version de Zabbix que vous souhaitez installer. Vous pouvez utiliser 5.4, 5.2, etc., en fonction de vos besoins.

3. Mettre à jour la liste des paquets disponibles :

Après avoir ajouté le référentiel, mettez à jour la liste des paquets disponibles :

sudo apt-get update