



CPAM Ardèche / Greta Ardèche-Drôme

GLPI

TP

Collet Valentin
BTS SIO-SISR / Session 2026

Sommaire

Cahier des charges	2
Descriptifs de l'existant	2
Besoins	2
Contraintes	3
Ressources	3
Analyse	4
Descriptif des solutions.....	4
Comparaison des solutions	5
Choix d'une solution	6
Etude de l'impact sur le SI existant.....	6
Plan d'adressage & schéma réseau :.....	7
Phasage de l'intervention / Prévision de test	7
Déploiement	8
Mise en place.....	8
Rapport de test	16
Rapport de déploiement.....	16
Bilan.....	16
Conclusion.....	16
Autocritique	16

Cahier des charges

Descriptifs de l'existant

Je dispose dans le cadre de ce TP d'un accès aux machines du GRETA avec un hyperviseur type 2 : **VMWARE Workstation**. En ce qui concerne les machines, la mienne est équipée de :

- CPU : Intel I7-10700 @ 2.90GHz
- 32 Go RAM DDR4
- 500Go SSD
- 2 cartes réseaux

De plus, je dispose au préalable déjà de l'ISO nécessaire pour faire une installation de Debian 13, tandis que j'utiliserais un master Windows 10 pour simuler des clients.

Enfin je dispose d'un accès au réseau GRETA avec comme plage :

Réseau pédagogique :	
Réseau :	172.26.0.0/21 (255.255.248.0)
Plage DHCP	172.26.1.0 – 172.26.2.254
IP fixe Vivarais	172.26.4.1 – 172.26.4.254
Passerelle	172.26.7.254
DNS	8.8.8.8

Besoins

Le besoin principal est l'installation d'un gestionnaire de services informatique, au sein d'un serveur LAMP (Linux, Apache, Mariadb/Mysql, PHP). Ce gestionnaire permettra de gérer les ressources matérielles, logicielles et même les demandes de support. De plus la plateforme devra être accessible avec un nom de domaine (local pour le TP) via web avec un échange entre clients et serveur sécurisé avec certificat SSL et donc HTTPS.

Je vais avoir besoin d'une VM Debian 13 :

- 2 GB de ram seront suffisant
- 1 cœur 2 threads pour le CPU
- Espace disque de 10GB suffisant.

Cette VM me permettra de mettre **un serveur LAMP** (Linux, Apache, MariaDB, Php) en place. Ensuite il sera nécessaire de mettre en place glpi sécurisé par un **certificat SSL** et un protocole **https**, avec une **redirection** http vers https. De plus j'aurais besoin d'un

accès **SSH** au serveur afin de pouvoir fluidifier les actions d'installation de GLPI et des paquets nécessaires (copier/coller).

- Ports nécessaires : http (80), https (443) & SSH (22)

Afin d'assurer le bon fonctionnement de GLPI il va me falloir plusieurs paquets sur la Debian 13 :

- PHP & PHP MySQL
- MariaDB
- Apache2

Et plusieurs extensions qui apparaîtront dans la procédure.

Il va falloir créer une **base de données** pour GLPI via MariaDB.

Contraintes

J'ai assez peu de contraintes au cours du TP. La principale qui me semble importante à souligner est la **contrainte temps** : je dispose de 2 TP de 8H pour réaliser l'installation du serveur LAMP GLPI soit **16H** maximum au total.

De plus il faut respecter une certaine convention de nommage :

- Nom de machine = <au choix>, chez moi sera GLPI;
- Nom de domaine = nom.local soit collet.local

Au **niveau matériel**, j'ai accès aux ordinateurs du GRETA mais n'ayant pas d'accès à distance à ceux-ci, je ne peux avancer le TP à domicile mis-à-part en faisant un export des VM ou en reproduisant de zéro l'installation.

Enfin, la dernière contrainte notable est les plages d'IP fixe : il faut que je m'assure que l'IP que j'attribuerai au serveur LAMP GLPI sera libre.

Ressources

Je dispose d'un poste de travail avec un hyperviseur de type 2 sur lequel je ne m'étendrai pas plus (cf existant).

L'intervenant en cours est ici une personne ressource qui peut aiguiller et conseiller en cas de questionnement.

Enfin je dispose de plusieurs cours qui me servent de documentation sur les architectures client-serveur.

Analyse

Descriptif des solutions

GLPI: Gestionnaire Libre de Parc Informatique, est une application web open-source et donc gratuite, qui permet la gestion exhaustive d'un parc informatique : allant de la gestion du matériel (ordinateurs, serveurs, imprimantes, périphériques, etc...) complète avec des informations exhaustives sur les **matérielles** (date d'achat, numéro de série, affectation, etc...), aux **logicielles** (suivi de licence), jusqu'à la **demande d'assistance** avec un système de gestion des tickets avec helpdesk pour les utilisateurs qui leur permettent de créer des demandes d'assistance avec descriptif avancé (dont spécification de l'urgence de la demande), accompagné d'une gestion du support et des interventions côté technicien avec planification, interventions techniques et gestion des incidents selon le modèle **ITIL** (*Information Technology Infrastructure Library*), qui plus est avec *un système de base de donnée de solution*. Cette application open-source est un outil **ITSM** (Information Technology Service Management : pratiques permettant de gérer et améliorer les services informatiques) et **ITAM** (IT Asset Management : plus sur la gestion des actifs). En revanche, l'outil est aussi complexe que complet et nécessite des configurations plus importantes, mais dispose d'une communauté conséquente avec donc des possibilités de support. Enfin il est possible tout de même d'avoir un support professionnel avec plusieurs souscriptions allant de 100€ par mois, 300€ /mois, jusqu'à 1000€ selon les nécessités.

Snipe-IT : est un gestionnaire de ressources informatiques, à la différence de GLPI il est plus orienté pour la gestion du matériel et des logiciels, simple d'accès et d'utilisation avec une interface web intuitive, l'application dispose tout de même d'un support de 4999\$ / an pour les entreprises avec possibilité d'hébergement de l'application, sinon il y a toujours possibilité d'accéder gratuitement à un support open-source. L'applicatif dispose d'une sécurité importante. Enfin celui-ci s'oriente donc plus ITAM.

Comparaison des solutions

	GLPI	SNIPE-IT
Inventaire matériel	Permet le suivi du matériel : poste et périphériques de manière exhaustive avec association entre les objets (postes + souris + écrans) au sein de multiples “entités” (service, site).	Est aussi performant pour la gestion d'inventaire similaire à GLPI avec historique, assignation, personnalisation possible.
Gestionnaire de ticket, support utilisateurs	Propose un système de tickets complexes avec des bases de connaissances permettant d'aider les techniciens à résoudre les tickets récurrents.	Ne propose pas de solution de support.
Software	Permet aussi la gestion des logicielles (versions, dates, etc...), avec les licences et les coûts.	Permet un suivie logiciel mais n'est pas aussi complet sur le plan des licences et surtout du suivi du financement.
Automatisations et personnalisations	Complet à ce niveau avec des possibilités d'inventaire automatique grâce à l'agent GLPI et de nombreuses possibilités de personnalisation avec les plugins, de plus il est compatible avec des services d'annuaire type LDAP ou AD.	Compatible LDAP / AD, mais pas d'agent qui permet d'inventorier les périphériques via le réseau.
Accessibilité	Considéré plus difficile d'accès, avec une configuration lourde en complexité, mais avec des personnalisations possibles très complètes.	Facile d'accès et de mise en place.
Taille	Recommandé pour des parcs d'entreprises complexes.	Parc simple.

Choix d'une solution

Dans le contexte du TP la solution choisie est GLPI, sinon les deux applicatifs ont des points forts très spécifiques et distincts.

Là où Snipe-IT brille est son accessibilité et sa mise en œuvre facile, et peut largement suffire dans le suivi matériel et logiciel au sein d'une petite entreprise. Mais il ne propose pas de solution de ticketing et helpdesk, il ne propose pas d'agent pour la découverte "automatiser" des objets sur le réseau.

Tandis que GLPI est la solution complète par excellence, l'application est adaptée aux grandes entreprises et se conforment bien à de multiples sites (avec sa gestion des entités), l'application est en revanche plus complexe à mettre en place.

Etude de l'impact sur le SI existant

Le SI disposera donc d'un gestionnaire de parc informatique et d'un gestionnaire de ticket accessible, de plus toutes les prochaines machines sur le site pourront être déployées avec l'agent GLPI pour faciliter le suivi du matériel connecté sur le réseau. Les impacts seront :

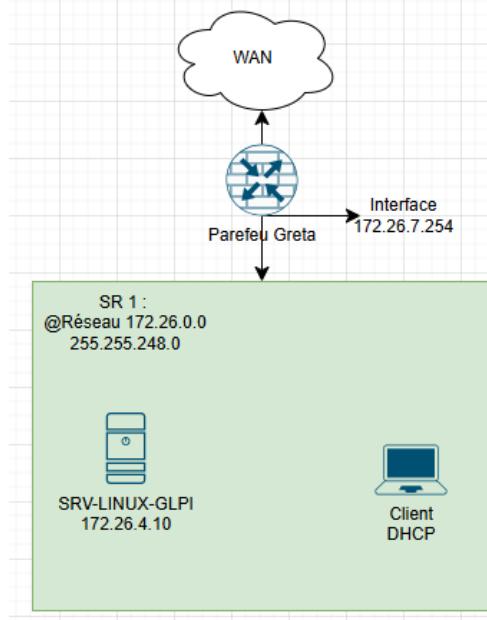
Organisationnels : en simplifiant le suivie des ressources matérielles et logicielles, en fluidifiant les échanges autour d'un seul outil et en fluidifiant la communication interne

Economique : le suivie des licences peut ainsi éviter des problèmes de paiement (type retard) et ainsi réduire les coûts d'exploitation

Humains : grâce à la gestion des tickets et la possibilité de proposer une plateforme de support aux utilisateurs où les techniciens pourront dépanner. Enfin l'agent permet de sauvegarder des ressources humaines en faisant remonter automatiquement les informations des postes ce qui évite des erreurs humaines.

Plan d'adressage & schéma réseau :

VM	Adresse	Gateway
GLPI	172.26.4.10/21	172.26.7.254
Client Windows 1	DHCP	DHCP



Phasage de l'intervention / Prévision de test

- 1) Installation VM Debian 13 avec partitionnement adapté au serveur LAMP
- 2) Installation composants LAMP et fonctionnement
 - o Test des services Apache2 et mariadb
 - o Test des ports nécessaires (80 http, 443 https, 22 ssh)
 - Net-tools : ss -tln
- 3) Vérification côté client du fonctionnement d'Apache:
 - o Test 1 : Accès web via <http://<ip>>
 - o Test 2 : Mise en place d'un index.php avec du langage **php** pour vérifier sa traduction : <http://<ip>/index.php>
- 4) Création des fichiers de configuration de GLPI :
 - o Hôte virtuel
- 5) Activation du site par défaut pour le service apache.
- 6) Préparation de la base de données pour GLPI
 - o Test : vérification de la **création de la base de données** GLPI sur mariadb
 - Via connect glpi / show databases ;
- 7) Téléchargement et gestion des permissions Debian
 - a. Sélection de la version "stable GLPI"

- b. Gestion des permissions fichiers debian
- c. Test : vérifier les droits et propriétaire du dossier GLPI avec : **ls -l**
- 8) Installation GLPI via la console web.
 - o Ajouts des extensions
 - o Normes de sécurités
 - o Suppression du dossier installation
- 9) Installation d'un agent GLPI sur un poste
 - o Test : Vérifier que le client remonte bien sur l'interface web GLPI dans **l'inventaire**
- 10) Mise en place HTTPS et certificats SSL
 - o Activation du module SSL
 - o Générations clés publiques / privées
 - o Création fichier de configuration HTTPS (virtual host https)
 - o Test : pour vérifier le bon fonctionnement il suffit d'accéder à la console web GLPI en <https://<nom>.de domaine>/glpi>
- 11) Mise en place de la redirection automatique HTTP vers HTTPS
 - o Test : Tenter d'accéder à la console web via navigateur en HTTP, le transfert en https devrait être automatique.

Déploiement

Le premier TP de 8h aura pour but l'installation et la mise en service http de GLPI. Tandis que dans un deuxième temps, les dernières 8h concerteront la mise en place de l'agent natif GLPI sur client et le switch du HTTP vers le HTTPS.

Mise en place

1) Installation Debian 13 :

Ici la seule particularité va être le partitionnement du disque et le nom de domaine :

- Nom de machine : GLPI
- Nom de domaine : collet.local
- Partitionner les disques > **manuel** :
 - o / : 6 go (primaire) > début (Amorçage présent)
 - o var : 3 go primaire > début >
 - o swap : 1go > primaire > fin

▽ SCSI28 (0,0,0) (sda) - 21.5 GB VMware, VMware Virtual S

>	n° 1	primaire	10.0 GB	B f	ext4	/
>	n° 2	primaire	3.0 GB	f	ext4	/var
>		pri/log	7.5 GB		Espace libre	
>	n° 3	primaire	999.3 MB	f	swap	swap

2) Installation des composants LAMP

Il faut installer les paquets nécessaires : php et php-mysql, apache2 et mariadb-server

- **apt install apache2**
- **apt install mariadb-server**
- **apt install php && php-mysql**

Vérification :

- systemctl status apache2

```
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
  Active: active (running) since Tue 2025-10-14 09:00:06 CEST; 13s ago
```

- systemctl status mariadb

```
● mariadb.service - MariaDB 11.8.3 database server
  Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
  Active: active (running) since Tue 2025-10-14 08:59:43 CEST; 1min 2s ago
    Invocation: 4d96402118454369a093285046ef5302
```

Vu que Php n'est pas un service il faut vérifier via :

- **apt show php / php-mysql**

Vérification des ports :

Il va falloir le port 80 (http), le port 22 (ssh) pour l'accès en ssh et ainsi faciliter l'installation via putty ou juste cmd, et enfin plus tard le port 443 (https).

Pour ceci j'utilise l'outils **net-tools** :

- **apt install net-tools**
- **ss -tl** (pour voir les port d'écoutes)

```
root@TelloCDeb:~# ss -tl
State      Recv-Q      Send-Q      Local Address:Port          Peer Address:Port
LISTEN      0            80          127.0.0.1:mysql          0.0.0.0:*
LISTEN      0           128          0.0.0.0:ssh             0.0.0.0:*
LISTEN      0           128          [::]:ssh              [::]:*
LISTEN      0            511          *:http               *:*
```

OU

- **ss -tln**

```
root@TellocDeb:~# ss -tln
State      Recv-Q      Send-Q      Local Address:Port
LISTEN      0          80          127.0.0.1:3306
LISTEN      0          128         0.0.0.0:22
LISTEN      0          128         [::]:22
LISTEN      0          511         *:80
```

3) Vérification côté client Apache et PHP

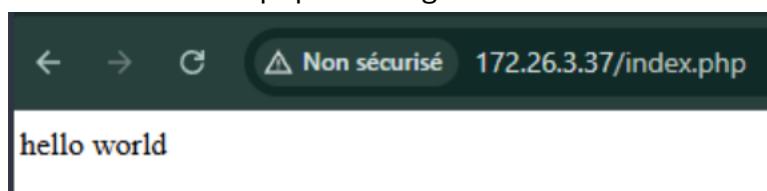
Via le navigateur web il suffit d'aller sur <http://<ip>> du serveur>



Puis, pour vérifier le bon fonctionnement de PHP :

J'ai créé un fichier index.php dans /var/www/html/index.php comme ceci :

- <?php
echo "hello world" ;
?>
- Je vérifie la traduction du php via navigateur web :



4) Crédit des fichiers de configuration de GLPI :

Dans /etc/apache2/sites-available :

- **Touch** glpi-telloc.conf

```
<VirtualHost :80>
    ServerName www.glpi.collet.local
    ServerAlias glpi.collet.local

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/glpi
    <Directory "/var/www/html/glpi">
        require all granted
        Options Indexes FollowSymLinks
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

5) Activation du site par défaut pour le service apache.

Sur le serveur ou en SSH :

- **a2ensite glpi-telloc.conf**
- Puis, **a2dissite 000-default.conf**
- **Systemctl reload apache2**

6) Préparation de la base de données pour GLPI

Pour ceci sur le serveur GLPI ou en SSH :

- **Mariadb -u root -p**
- **CREATE DATABASE glpi ;**
 - o Tester avec : **connect glpi / show databases ;**
- **CREATE USER glpi@localhost identified by « <mot de passe> » ;**
- **GRANT ALL PRIVILEGES ON glpi.* to glpi@localhost identified by “<mot de passe>” ;**

7) Téléchargement et permission GLPI

Il faut ensuite télécharger la dernière version de glpi, pour ceci j'y accède via github, il faut idéalement sélectionner la dernière version "stable" :

- **wget <https://github.com/glpi-project/glpi/releases/download/11.0.1/glpi-11.0.1.tgz> / essayer de : 10.0xx.19**
- **tar -xvf glpi.tgz /var/www/html**
 - o L'archive sera donc extraite dans : /var/www/html/glpi

- Puis au niveau de la gestion des **permissions** :
 - o **chown -R www-data:www-data glpi**
 - o **chmod -R 755 glpi**

Vérification :

- Via la commande : **ls -l**
 - o Droits RWX du fichier = 755 soit : RWX/R_X/R_X
 - o Propriétaire / groupe du fichier = www-data

8) Installation GLPI via navigateur web

Ici je me connecte via navigateur web à www.glpi.collet.local , pour cela il faut penser à modifier le fichier host de windows (windows/system32/drivers/etc/host) pour y ajouter le domaine avec IP :

```
# localhost name resolution is handled within DNS
#      127.0.0.1      localhost
#      ::1            localhost
172.26.3.10      glpi.collet.local
```



Il faut ensuite créer les fichiers demandés comme ceci (chemin direct = **/var/www/html/glpi/config**) :

- **touch config_db.php**
- **touch glpicrypt.key**

The screenshot shows the GLPI Setup page with the title "GLPI SETUP" and "Étape 0". It displays a table of compatibility tests:

TESTS EFFECTUÉS	RÉSULTATS
Requis Parser PHP	✓
Requis Configuration des sessions	✓
Requis Mémoire allouée	✓
Requis mysql extension <i>l'extension mysql est absente.</i>	✗
Requis Extensions du noyau de PHP <i>Les extensions suivantes sont installées : fileinfo, filter, libxml, json.</i> <i>Les extensions suivantes sont manquantes : dom, simplexml, xmlreader, xmlwriter.</i>	✗
Requis curl extension <i>Requis pour l'accès à distance aux ressources (requêtes des agents d'inventaire, Marketplace, flux RSS, ...).</i> <i>l'extension curl est absente.</i>	✗
Requis gd extension <i>Requis pour le traitement des images.</i> <i>l'extension gd est absente.</i>	✗
Requis intl extension <i>Requis pour l'internationalisation.</i> <i>l'extension intl est absente.</i>	✗

Il faut installer toutes les extension demandées :

- **apt install php-xml php-common php-json php-mysql php-mbstring php-curl php-gd php-intl php-zip php-bz php-apcu**

Puis, dans le fichier **/etc/php/8.4/apache2/php.ini**

- **ctrl + f (chercher) : session.cookie_httponly = ON**

Enfin se connecter à la base de données

- Ici typiquement => id : glpi + mdp = glpi

Du côté du serveur maintenant que l'installation est terminé : penser à supprimer le fichier install de : **/var/www/html/glpi/install**

9) Installation de l'agent natif :

Pour cela il suffit de télécharger l'agent GLPI compatible avec la version GLPI installé sur le debian, puis lors de l'installation il faut choisir un dossier et indiquer l'url du serveur GLPI. Bien penser à rentrer le **nom de domaine** (<http://glpi.collet.local/glpi>).

Sur l'interface d'admin GLPI :

- Administration -> inventaire -> **activer inventaire**

/ ! \ En cas de problème :

Sur le client :

- **net stop agent-glpi**
- **net stat agent-glpi**
- possible de 127.0.0.1:62xxxx force agent
- via cmd : lancer le : agent.bat –force (dans le répertoire de l'agent)

10) Mise en place HTTPS et SSL

Il faut activer le module SSL

- **A2enmode ssl**
- **Systemctl restart apache2**

Puis il faut générer une clé publique et privée, pour cela je crée un dossier spécifique dans **/etc/apache2/ssl/** :

- **mkdir /etc/apache2/ssl/glpi**

Puis depuis ce dossier créer :

- **openssl req -x509 -nodes -days 365 -newkey rsa:2048 -sha256 -out /etc/apache2/ssl/glpi/apache.crt -keyout /etc/apache2/ssl/glpi/apache.key**

Régler les permissions des clés, la clé publique doit respecter :

```
-rw-r----- 1 root www-data 396 2 sept. 16:19 apache.crt
-rw----- 1 root root     1679 2 sept. 16:19 apache.key
```

- **chown root:www-data apache.crt**
- **chmod 440 apache.crt**

Puis il faut créer un nouveau fichier “virtual hosts” pour https :

Dans **/etc/apache2/sites-available** :

- **cp default-ssl.conf glpi-telloc-ssl.conf OU touch glpi-telloc-ssl.conf**

```
<ifmodule mod_ssl.c>
    <VirtualHost *:443>
        ServerName www.glpi.collet.local
        ServerAlias glpi.collet.local
        DocumentRoot /var/www/example.com/public_html

        SSLEngine on
        SSLCertificateFile /etc/apache2/ssl/glpi/apache.crt
        SSLCertificateKeyFile /etc/apache2/ssl/glpi/apache.key
    </VirtualHost>
</ifmodule>
```

- A2ensite glpi-telloc-ssl.conf
- Systemctl reload apache2

Puis il suffit pour tester la fonctionnalité d'aller sur le navigateur web est d'accéder en <https://<nom>/glpi>.

11) Redirection HTTP vers HTTPS :

Dans le fichier : /etc/apache2/sites-available/glpi-collet.conf :

- Rajouter la ligne : **Redirect permanent / <https://glpi.collet.local>**

```
<VirtualHost *:80>
    ServerName glpi.baudiment.local

    DocumentRoot "/var/www/html/glpi"
    Redirect permanent / https://glpi.baudiment.local
    <Directory "/var/www/html/glpi">
        require all granted
        Options Indexes FollowSymLinks
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Vérification :

Tester le fonctionnement sur navigateur (le http devrait transférer vers https).

Rapport de test

Les tests sont réalisés et documentés dans le détail au cours de la procédure, afin de permettre une inclusion des tests ordonnée.

Les tests vont premièrement concernés les services, en s'assurant leur bon fonctionnement sur le serveur GLPI, puis vérifier l'ouverture des ports nécessaires (80 http, 443 https, 22 SSH). Vis-à-vis d'Apache et PHP il est possible de les vérifier via un poste client sur navigateur web.

Ensuite il faut s'assurer de la création de la base de données MariaDB.

Puis, les droits et propriétaire doivent être correctement positionné sur le serveur GLPI.

Enfin il faut vérifier le bon fonctionnement du site en http puis de la redirection en HTTPS dans un second temps.

Côté client il faut s'assurer que les postes avec l'agent remontent dans l'inventaire GLPI.

Rapport de déploiement

Le déploiement de GLPI s'est correctement déroulé, avec des postes qui sont inventoriés automatiquement grâce à l'agent GLPI, le site est sécurisé en HTTPS, avec l'url http qui redirige automatiquement vers HTTPS.

Bilan

Conclusion

Ce TP s'est plutôt bien déroulé et m'a permis d'appréhender certains protocoles avec lesquels je n'étais pas à l'aise (principalement https avec les clés publiques et privés de sécurité SSL).

Enfin il m'a permis de me renseigner de manière plus exhaustive sur les alternatives de gestionnaires de parc, et m'a permis de commencer à appréhender GLPI qui est un gestionnaire aussi complet que complexe.

Autocritique

Le test PHP via un « echo » m'a demandé une remise au goût du jour, de plus j'ai pu réaliser sans difficultés la mise en place de la sécurité HTTPS qui me paraissait difficile (ce qui au passage m'a permis d'appréhender le système de certification par clé publique et clé privée).