

RUNBOOK

Migration GLPI en Production

10.0.15 → 11.0.6 | Debian 10 → Debian 13

	Migration GLPI Production — Amarris SRV-GLPI-PROD (Debian 10 / GLPI 10.0.15)
	SRV-GLPI-NEW (Debian 13) GLPI 11.0.6 / PHP 8.4 / MariaDB 11.8 Neven — BTS SIO SISR — Amarris
	À compléter 21h00 → 06h00 (estimation : 5h à 6h30)

0. Points critiques à ne pas oublier

⚠ Vider la table `glpi_queuednotifications` sur la source AVANT le dump — elle fait 4 Go et représente 3 à 4h d'import inutile.

⚠ Récupérer le fichier `glpicrypt.key` depuis la source avant la migration — sans lui, les mots de passe LDAP et collecteurs mail seront illisibles.

⚠ Configurer le buffer MariaDB à 4 Go sur la cible AVANT l'import — sans ça, l'import sera 30% plus lent.

⚠ Ne jamais mettre le PC/terminal en veille pendant l'import — utiliser screen pour que le processus survive à une déconnexion.

✓ Configurer les keepalives PuTTY à 60 secondes (Connection > Seconds between keepalives) pour éviter les déconnexions.

✓ Garder l'ancien serveur allumé et accessible jusqu'à validation complète — plan de rollback possible en 5 minutes.

✓ GLPI 10.0.18 est incompatible avec PHP 8.4 — passer directement à GLPI 11.0.6 qui supporte PHP 8.4.

1. Préparation avant la migration (J-1)

1.1 Vérifications préliminaires

À effectuer la veille de la migration pour éviter les mauvaises surprises :

Sur le serveur source

```
# Vérifier l'espace disque disponible
df -h

# Vérifier la taille de la base GLPI
mysql -u root -p -e "SELECT ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024, 2) AS 'Taille (GB)' FROM information_schema.tables WHERE table_schema='glpi';"

# Vérifier la taille de glpi_queuednotifications
mysql -u root -p -e "SELECT ROUND((data_length + index_length) / 1024 / 1024 / 1024, 2) AS 'Taille (GB)' FROM information_schema.tables WHERE table_schema='glpi' AND table_name='glpi_queuednotifications';"

# Inventaire des plugins
ls /var/www/glpi/plugins/
ls /var/www/glpi/marketplace/

# Vérifier que glpicrypt.key existe
ls -la /var/www/glpi/config/glpicrypt.key
```

Sur le serveur cible

```
# Vérifier l'espace disque (minimum 30 Go libres recommandés)
df -h

# Vérifier la connectivité SSH vers la source
ssh root@IP_SERVEUR_SOURCE 'echo OK'
```

1.2 Vider la table de notifications (CRITIQUE)

⚠ Cette étape est obligatoire — elle évite 3 à 4h d'import inutile sur une table de 4 Go qui ne contient que des emails en attente non critiques.

```
# Sur le serveur SOURCE
mysql -u root -p -e "TRUNCATE TABLE glpi.glpi_queuednotifications;"

# Vérifier que la table est bien vide
mysql -u root -p -e "SELECT COUNT(*) FROM glpi.glpi_queuednotifications;"
```

2. Installation de la stack sur le serveur cible

À effectuer sur le serveur cible (Debian 13) avant le début de la fenêtre de maintenance.

2.1 Mise à jour du système

```
| apt update && apt upgrade -y
```

2.2 Installation Apache + MariaDB + PHP 8.4

```
| apt install -y apache2 mariadb-server mariadb-client curl wget \  
  php8.4 php8.4-mysql php8.4-xml php8.4-curl php8.4-gd \  
  php8.4-mbstring php8.4-intl php8.4-zip php8.4-bz2 \  
  php8.4-opcache php8.4-ldap php8.4-apcu php8.4-bcmath \  
  libapache2-mod-php8.4
```

2.3 Sécurisation MariaDB

```
| mariadb-secure-installation  
# Répondre : n / y / y / y / y / y
```

2.4 Configuration du buffer MariaDB (CRITIQUE)

⚠ À faire AVANT l'import — le buffer passe de 128 Mo à 4 Go, ce qui accélère l'import de 30%.

```
# Ajouter la configuration  
cat >> /etc/mysql/mariadb.conf.d/50-server.cnf << 'EOF'  
  
innodb_buffer_pool_size = 4G  
EOF  
  
# Redémarrer MariaDB  
systemctl restart mariadb  
  
# Vérifier que le buffer est bien à 4 Go  
mysql -u root -p -e "SELECT @@innodb_buffer_pool_size / 1024 / 1024 / 1024 AS 'Buffer  
(GB)';"
```

2.5 Création de la base de données GLPI

```
mysql -u root -p  
  
CREATE DATABASE glpi CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;  
CREATE USER 'glpi'@'localhost' IDENTIFIED BY 'MOT_DE_PASSE_GLPI';  
GRANT ALL PRIVILEGES ON glpi.* TO 'glpi'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;
```

2.6 Configuration Apache

```
# Activer le module rewrite  
a2enmod rewrite  
  
# Créer le VirtualHost GLPI  
cat > /etc/apache2/sites-available/glpi.conf << 'EOF'  
<VirtualHost *:80>  
  ServerName IP_SERVEUR_CIBLE
```

```
DocumentRoot /var/www/glpi/public
<Directory /var/www/glpi/public>
    AllowOverride All
    Require all granted
</Directory>
LogLevel warn
ErrorLog ${APACHE_LOG_DIR}/glpi_error.log
CustomLog ${APACHE_LOG_DIR}/glpi_access.log combined
</VirtualHost>
EOF

# Activer le site et désactiver le site par défaut
a2dissite 000-default
a2ensite glpi
systemctl restart apache2
```

2.7 Configuration PHP

```
cat > /etc/php/8.4/apache2/conf.d/99-glpi.ini << 'EOF'
memory_limit = 256M
upload_max_filesize = 20M
post_max_size = 20M
max_execution_time = 600
session.cookie_httponly = On
session.cookie_samesite = Lax
EOF

systemctl restart apache2
```

2.8 Téléchargement de GLPI 11.0.6

```
wget https://github.com/glpi-project/glpi/releases/download/11.0.6/glpi-11.0.6.tgz

# Décompresser dans /var/www
cd /var/www
tar xzf /root/glpi-11.0.6.tgz

# Créer le .htaccess
cat > /var/www/glpi/public/.htaccess << 'EOF'
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php [QSA,L]
EOF
```

3. Sauvegarde du serveur source

À effectuer au début de la fenêtre de maintenance, après avoir prévenu les utilisateurs.

3.1 Dump de la base de données

✓ Utiliser `--single-transaction` pour ne pas bloquer GLPI pendant le dump. La compression gzip réduit 8 Go en ~400 Mo.

```
# Sur le serveur SOURCE
mysqldump -u root -p --single-transaction --quick glpi | gzip >
/root/glpi_backup.sql.gz

# Vérifier la taille du dump (doit faire ~300-500 Mo)
ls -lh /root/glpi_backup.sql.gz
```

3.2 Récupération du fichier de chiffrement (CRITIQUE)

⚠ Sans ce fichier, les mots de passe LDAP et les collecteurs mail seront illisibles sur le nouveau serveur.

```
# Vérifier que le fichier existe
ls -la /var/www/glpi/config/glpicrypt.key

# Le transférer vers la cible
scp /var/www/glpi/config/glpicrypt.key root@IP_SERVEUR_CIBLE:/root/glpicrypt.key
```

3.3 Relever les identifiants de connexion BDD

```
cat /var/www/glpi/config/config_db.php
# Noter : dbuser, dbpassword, dbdefault
```

4. Transfert des données vers la cible

4.1 Transfert du dump SQL

```
# Sur le serveur SOURCE
scp /root/glpi_backup.sql.gz root@IP_SERVEUR_CIBLE:/root/

# Vérifier que le fichier est bien arrivé sur la cible
# Sur le serveur CIBLE :
ls -lh /root/glpi_backup.sql.gz
```

4.2 Transfert des fichiers GLPI

✓ Les logs ne sont pas transférés (_log) — ils représentent plusieurs Go inutiles. Les fichiers _cache et _sessions se régénèrent automatiquement.

```
# Sur le serveur SOURCE
rsync -av \
  --exclude='_cache' \
  --exclude='_sessions' \
  --exclude='_tmp' \
  --exclude='_log' \
  /var/www/glpi/files/ root@IP_SERVEUR_CIBLE:/var/www/glpi/files/
```

4.3 Transfert des plugins

```
# Sur le serveur SOURCE
rsync -av /var/www/glpi/plugins/ root@IP_SERVEUR_CIBLE:/var/www/glpi/plugins/
rsync -av /var/www/glpi/marketplace/ root@IP_SERVEUR_CIBLE:/var/www/glpi/marketplace/
```

5. Import de la base de données

5.1 Lancer l'import dans un screen

⚠ Utiliser screen obligatoirement — si la connexion SSH se coupe, l'import continue en arrière-plan.

```
# Sur le serveur CIBLE
apt install -y screen

# Ouvrir un screen
screen -S import

# Lancer l'import avec optimisations
zcat /root/glpi_backup.sql.gz | mysql -u root -p \
  --init-command="SET foreign_key_checks=0; SET unique_checks=0; SET sql_log_bin=0;" \
  glpi

# Se détacher du screen sans stopper l'import : Ctrl+A puis D
```

5.2 Surveiller la progression

```
# Taille importée (relancer toutes les 15-20 min)
mysql -u root -p -e "SELECT ROUND(SUM(data_length + index_length) / 1024 / 1024 /
1024, 2) AS 'Taille (GB)' FROM information_schema.tables WHERE table_schema='glpi';"

# Voir quelle opération est en cours
mysql -u root -p -e "SHOW PROCESSLIST;"

# Revenir dans le screen si besoin
screen -r import
```

Taille base source	Durée estimée de l'import
< 3 Go	~1h
3 à 6 Go	~1h30 à 2h30
6 à 10 Go	~2h30 à 4h
> 10 Go	> 4h — envisager mysdumper/myloader

6. Configuration et migration GLPI

6.1 Créer le fichier config_db.php

```
cat > /var/www/glpi/config/config_db.php << 'EOF'
<?php
class DB extends DBmysql {
    public $dbhost      = 'localhost';
    public $dbuser      = 'glpi';
    public $dbpassword  = 'MOT_DE_PASSE_GLPI';
    public $dbdefault   = 'glpi';
    public $use_utf8mb4 = true;
    public $allow_signed_keys = false;
    public $allow_datetime = false;
    public $use_timezones = true;
}
EOF

chown www-data:www-data /var/www/glpi/config/config_db.php
chmod 640 /var/www/glpi/config/config_db.php
```

6.2 Restaurer le fichier de chiffrement

```
cp /root/glpicrypt.key /var/www/glpi/config/glpicrypt.key
chown www-data:www-data /var/www/glpi/config/glpicrypt.key
chmod 640 /var/www/glpi/config/glpicrypt.key
```

6.3 Appliquer les permissions

```
chown -R www-data:www-data /var/www/glpi
chmod -R 755 /var/www/glpi
chmod -R 775 /var/www/glpi/files
chmod -R 775 /var/www/glpi/config
```

6.4 Lancer la migration GLPI

⚠ Cette étape peut prendre 30 min à 2h selon la taille de la base. Ne pas interrompre.

```
php /var/www/glpi/bin/console db:update --allow-superuser
# Répondre 'Yes' aux deux questions
```

6.5 Migrations complémentaires

```
# Migrer les tables en utf8mb4
php /var/www/glpi/bin/console migration:utf8mb4 --allow-superuser

# Migrer les clés en unsigned
php /var/www/glpi/bin/console migration:unsigned_keys --allow-superuser
```

7. Tests et validation

7.1 Vérifications techniques

```
# Vérifier les prérequis GLPI
php /var/www/glpi/bin/console system:check_requirements

# Vérifier qu'Apache fonctionne
systemctl status apache2 --no-pager

# Vérifier les logs Apache
tail -20 /var/log/apache2/glpi_error.log
```

7.2 Grille de validation fonctionnelle

Test	Résultat attendu	Statut
Accès à http://IP_CIBLE via navigateur	Page de connexion GLPI	<input type="checkbox"/> À valider
Connexion compte administrateur	Tableau de bord affiché	<input type="checkbox"/> À valider
Vérification entités (GROUPE > IT > sous-entités)	Hiérarchie complète visible	<input type="checkbox"/> À valider
Inventaire ordinateurs (Parc > Ordinateurs)	Nombre identique à la source	<input type="checkbox"/> À valider
Tickets (Assistance > Tickets)	Historique complet présent	<input type="checkbox"/> À valider
Utilisateurs (Administration > Utilisateurs)	Même nombre que la source	<input type="checkbox"/> À valider
Pièces jointes téléchargeables	Fichiers accessibles	<input type="checkbox"/> À valider
Authentification LDAP	Connexion avec compte AD	<input type="checkbox"/> À valider
Plugins fonctionnels	Pas d'erreur plugin	<input type="checkbox"/> À valider

8. Configuration post-migration

8.1 Configurer le cron GLPI

```
echo "*/*2 * * * * www-data /usr/bin/php /var/www/glpi/bin/console system:run_actions > /dev/null 2>&1" > /etc/cron.d/glpi
chmod 644 /etc/cron.d/glpi
```

8.2 Vérifier les ACL après migration

GLPI 11 a modifié certains droits lors de la migration. Vérifier dans Administration → Profils :

- Profil Super-Admin — vérifier tous les droits
- Profil Technicien — vérifier droits sur tickets et inventaire
- Profil Observateur — vérifier accès en lecture seule
- Droits sur projecttask, tasktemplate, form — modifiés lors de la migration vers 11.0.6

8.3 Recommandations

- Mettre en place HTTPS avec un certificat SSL
- Configurer les sauvegardes automatiques de la base de données
- Vérifier et mettre à jour les plugins vers leurs versions compatibles GLPI 11
- Tester l'envoi de notifications email
- Désactiver l'ancien serveur après 48h de validation

9. Plan de rollback

En cas d'échec ou de problème bloquant, le retour à l'ancien serveur prend moins de 5 minutes.

⚠ Ne pas éteindre l'ancien serveur avant validation complète du nouveau — il sert de filet de sécurité.

9.1 Procédure de rollback

- Remettre l'ancien serveur en production (modifier le DNS ou l'IP si un basculement a été fait)
- Informer les utilisateurs que le service est restauré sur l'ancien serveur
- Analyser les erreurs sur le nouveau serveur avant de relancer la migration

9.2 Causes fréquentes d'échec et solutions

Problème	Solution
Import BDD interrompu	Vider la base, relancer dans screen avec les optimisations
GLPI affiche 'Not Found'	Créer /var/www/glpi/public/.htaccess avec les règles RewriteEngine
Erreur PHP version	GLPI 10.x incompatible PHP 8.4 — utiliser GLPI 11.0.6
Access denied MariaDB	Recréer l'utilisateur glpi avec GRANT ALL PRIVILEGES
Migration CLI bloquée sur ALTER TABLE	KILL le processus, TRUNCATE la table concernée, relancer
Données manquantes après migration	Vérifier l'entité sélectionnée — passer en 'toutes les entités'

10. Checklist de migration

À cocher au fur et à mesure de la migration :

#	Action	Heure
<input type="checkbox"/>	Prévenir les utilisateurs de l'indisponibilité	
<input type="checkbox"/>	Vider gpi_queuednotifications sur la source	
<input type="checkbox"/>	Stack Apache + MariaDB + PHP installée sur la cible	
<input type="checkbox"/>	Buffer MariaDB configuré à 4 Go sur la cible	
<input type="checkbox"/>	Base gpi créée sur la cible	
<input type="checkbox"/>	Dump SQL effectué sur la source	
<input type="checkbox"/>	gpicrypt.key récupéré depuis la source	
<input type="checkbox"/>	Dump SQL transféré vers la cible	
<input type="checkbox"/>	Fichiers GLPI (files/) transférés sans les logs	
<input type="checkbox"/>	Plugins transférés	
<input type="checkbox"/>	Import BDD lancé dans screen avec optimisations	
<input type="checkbox"/>	Import BDD terminé — taille vérifiée	
<input type="checkbox"/>	GLPI 11.0.6 décompressé dans /var/www/gpi	
<input type="checkbox"/>	config_db.php créé avec les bons identifiants	
<input type="checkbox"/>	gpicrypt.key copié dans /var/www/gpi/config/	
<input type="checkbox"/>	Permissions appliquées (www-data)	
<input type="checkbox"/>	Migration CLI GLPI lancée (db:update)	
<input type="checkbox"/>	utf8mb4 et unsigned_keys migrés	
<input type="checkbox"/>	VirtualHost Apache configuré + .htaccess créé	
<input type="checkbox"/>	GLPI accessible via navigateur	
<input type="checkbox"/>	Données validées (PC, tickets, utilisateurs)	
<input type="checkbox"/>	Cron GLPI configuré	
<input type="checkbox"/>	Utilisateurs prévenus de la remise en service	

11. Nouvelles fonctionnalités GLPI 11.0.6 vs 10.0.15

Cette migration vous fait passer d'une version de maintenance (10.0.15) à une version majeure (11.0.6). Voici les principales nouveautés dont votre équipe pourra bénéficier.

11.1 Actifs personnalisés natifs

Anciennement disponible via le plugin Generic Objects, cette fonctionnalité est désormais intégrée nativement dans GLPI 11.

- Création de tout type d'actif personnalisé directement dans le menu Parc
- Définition de champs sur mesure (texte, date, URL, liste déroulante, oui/non, nombre...)
- Association possible avec l'inventaire automatique
- Permissions granulaires par profil sur chaque champ (lecture seule, masqué, obligatoire)
- Remplacement natif du plugin Generic Objects — migration automatique lors de la mise à jour

✓ **Utile pour Amarris : créer des actifs de type 'Serveurs virtuels', 'Équipements réseau', 'Licences SaaS' avec des champs spécifiques à votre parc.**

11.2 Formulaires natifs

Le système de formulaires, anciennement géré par le plugin FormCreator, est désormais intégré nativement avec un éditeur repensé.

- Éditeur de formulaires interactif avec glisser-déposer
- Création de formulaires pour techniciens et utilisateurs finaux
- Liaison directe avec les tickets et les actifs
- Nouveau portail libre-service repensé pour les utilisateurs finaux
- Migration automatique depuis FormCreator lors de la mise à jour GLPI

11.3 Authentification multifacteurs (2FA / MFA)

GLPI 11 introduit l'authentification à deux facteurs pour renforcer la sécurité des comptes.

- Support du TOTP (Time-based One-Time Password)
- Compatible avec Microsoft Authenticator, Google Authenticator, Proton Authenticator
- Activable sur les comptes locaux GLPI et les comptes liés à une source externe (LDAP/AD)
- Peut être rendu obligatoire pour certains profils (ex : Super-Admin, techniciens)

✓ **Recommandé pour Amarris : activer le MFA obligatoire sur les comptes Super-Admin et techniciens pour renforcer la sécurité.**

11.4 Webhooks

GLPI 11 peut désormais déclencher des appels HTTP vers des services externes en fonction d'événements.

- Déclenchement sur événements liés aux tickets, à l'inventaire et à la gestion
- Intégration possible avec Microsoft Teams, Slack, SMS, outils ITSM tiers
- Exemple : notification Teams automatique quand un ticket urgent est créé
- Exemple : alerte SMS quand un équipement critique est modifié dans l'inventaire

11.5 Nouveau portail libre-service

- Interface utilisateur final entièrement repensée, plus intuitive
- Catalogue de services personnalisable
- Meilleure visibilité sur l'état des demandes en cours
- Compatible mobile

11.6 Améliorations de l'inventaire

- Association d'actifs à plusieurs groupes simultanément
- Filtrage des états des actifs dans les formulaires de périphériques
- Meilleure intégration avec GLPI Inventory Agent
- Nouvelles capacités de liaison entre actifs personnalisés et inventaire automatique

11.7 API haute disponibilité (HLAPI v2.1)

- Nouveaux champs disponibles pour les tickets
- Support étendu des objets GLPI via l'API
- Meilleure compatibilité avec les intégrations tierces

11.8 Compatibilité et infrastructure

Composant	GLPI 10.0.15	GLPI 11.0.6
PHP	7.4 à 8.3	8.2 à 8.4
MariaDB	10.2+	10.11+ (recommandé 11.x)
MySQL	8.0+	8.0+
Debian	10 / 11	12 / 13
Formulaires natifs	Plugin FormCreator requis	Intégré nativement
Actifs personnalisés	Plugin Generic Objects requis	Intégré nativement
MFA / 2FA	Non disponible	Disponible (TOTP)
Webhooks	Non disponible	Disponible nativement

11.9 Plugins intégrés nativement dans GLPI 11

Certains plugins très utilisés en GLPI 10 sont désormais intégrés nativement — ils ne nécessitent plus d'installation séparée.

Plugin GLPI 10	Équivalent GLPI 11	Migration
FormCreator	Formulaires natifs	Automatique via CLI
Generic Objects	Actifs personnalisés	Commande migration:genericobject_plugin_to_core
Fields	Champs personnalisés (partiel)	Plugin toujours nécessaire pour les actifs natifs

⚠ Les plugins FormCreator et Generic Objects installés sur votre ancien serveur doivent être migrés. Vérifier leur état après migration avec : `php bin/console migration:genericobject_plugin_to_core --allow-superuser`