

Haute Disponibilité (HA) pour GLPI

Alerte Incident :

Hier matin, le disque dur du serveur GLPI principal a saturé, entraînant un crash de la base de données. Pendant 4 heures, aucun technicien n'a pu accéder aux tickets, et la DSI était aveugle.

La mission (si vous l'acceptez... Mais bon, pas le choix) :

La direction ne veut plus jamais entendre parler de "Single Point of Failure" (Point de défaillance unique). Votre mission est de transformer notre architecture simple en une architecture Haute Disponibilité. Si un serveur web tombe, l'autre doit prendre le relais sans que l'utilisateur ne s'en aperçoive.

1. Le scénario "Catastrophe"

Mise en situation :

Hier à 14h00, le serveur GLPI principal a cessé de répondre. Disque dur HS.

Conséquence : 200 techniciens au chômage technique, aucun ticket traitable pendant 4 heures.

La demande de la DSI :

"Plus jamais ça. Je veux une architecture résiliente. Si un serveur tombe, un autre doit prendre le relais immédiatement sans intervention humaine."

Votre mission : Transformer votre GLPI "monolithique" (tout sur un serveur) en une **architecture 3-Tier distribuée**.

2. L'architecture (3 VMs)

Nous allons séparer les rôles pour éviter le SPOF (*Single Point of Failure*).

- **VM 1 : SRV-INFRA (Le cerveau)**
 - **Rôle :** Héberge les données critiques et dirige le trafic.
 - **Services :** MariaDB (Base de données), NFS (Partage de fichiers), HAProxy (Répartiteur de charge).
- **VM 2 & 3 : SRV-WEB-01 et SRV-WEB-02 (Les ouvriers)**
 - **Rôle :** Exécutent le code PHP de GLPI. Ils sont "jetables" et identiques.
 - **Services :** Apache2, PHP 8.3, Client NFS.

Étape 1 : Préparation de SRV-INFRA (Le cœur du système)

Cette machine est la plus importante. Elle va centraliser les données et la sécurité.

1.1 Externalisation de la base de données

Dans le TP précédent, la base était locale (localhost). Ici, elle doit accepter les connexions des serveurs Web.

1. **Installation :** `sudo apt install mariadb-server -y`
2. **Configuration réseau :**
 - o Éditez `/etc/mysql/mariadb.conf.d/50-server.cnf`.
 - o Changez `bind-address = 127.0.0.1` par `bind-address = 0.0.0.0` (Pour écouter tout le monde).
3. **Création de l'utilisateur distant :**

```
CREATE DATABASE glpidb;  
CREATE USER 'glpi_remote'@'%' IDENTIFIED BY 'password_solide';  
GRANT ALL PRIVILEGES ON glpidb.* TO 'glpi_remote'@'%';  
FLUSH PRIVILEGES;
```

(Rappel : N'utilisez jamais root pour l'application)

4. **Import des données :** Réutilisez votre sauvegarde .sql du TP "Crash Test" pour restaurer les données ici.

Le fichier se trouve normalement dans `/var/backups/glpi/` si tu as suivi le TP "Sauvegarde Automatique" **sur l'ancienne VM GLPI**. Identifie le nom exact du fichier .sql le plus récent. Il faut envoyer ce fichier vers **le nouveau serveur (SRV-INFRA)**. La meilleure méthode est d'utiliser la commande `scp` (Secure Copy) depuis l'ancienne machine.

Depuis l'ancienne machine, tape :

```
# Syntaxe : scp CHEMIN_FICHIER_SOURCE  
UTILISATEUR@IP_DESTINATION:/DOSSIER_CIBLE  
scp /var/backups/glpi/glpi_db_2024-11-27.sql utilisateur@IP_SRV_INFRA:/tmp/
```

(Remplace utilisateur par le nom de ton utilisateur sur la nouvelle VM et IP_SRV_INFRA par son IP réelle).

Une fois que le fichier est arrivé dans le dossier `/tmp` de ton nouveau serveur, connecte-toi dessus et lance l'importation.

Rappel du TP Crash Test: on utilise le chevron < pour injecter le contenu.

Haute disponibilité pour GLPI

Sur **SRV-INFRA** :

```
# On injecte le fichier SQL dans la base "glpidb" qu'on vient de créer
sudo mariadb glpidb < /tmp/glpi_db_2024-11-27.sql
```

Si aucune erreur ne s'affiche, c'est gagné. Tu peux vérifier que les tables sont bien là en faisant :
sudo mariadb -e "USE glpidb; SHOW TABLES;" Tu devrais voir une longue liste (glpi_computers, glpi_users, etc.).

1.2 Centralisation des Fichiers (NFS)

Les pièces jointes des tickets (images, PDF) ne doivent pas rester sur les serveurs Web, sinon elles ne seraient visibles que sur un seul serveur.

1. **Installation** : sudo apt install nfs-kernel-server -y

2. **Création du partage** :

```
sudo mkdir -p /srv/nfs/glpi_files
sudo chown -R www-data:www-data /srv/nfs/glpi_files
```

3. Export : Ajoutez dans /etc/exports :

```
/srv/nfs/glpi_files
*(rw,sync,no_subtree_check,all_squash,anonuid=33,anongid=33)
```

4. **Application** : sudo exportfs -a

1.3 Sécurité Réseau (Rappel TP UFW)

Comme vu précédemment, on ferme tout sauf le nécessaire. Sur **SRV-INFRA** :

```
sudo ufw allow 22/tcp      # SSH
sudo ufw allow 80/tcp      # HTTP (HAProxy)
sudo ufw allow 443/tcp     # HTTPS (HAProxy)
sudo ufw allow 3306/tcp    # MariaDB (Depuis les Web uniquement en prod !)
sudo ufw allow 2049        # NFS
sudo ufw enable
```

Note technique : Parfois, NFS utilise des ports aléatoires pour certains services auxiliaires (mountd), ce qui peut bloquer avec UFW. Si le montage NFS "mouline" dans le vide, essayez de tester en désactivant temporairement UFW. Si ça passe, c'est qu'il faudra fixer les ports NFS (mais généralement le port 2049 suffit pour les versions récentes).

Étape 2 : Les Serveurs Web

Effectuez ces opérations sur **SRV-WEB-01**, puis clonez la VM (ou refaites-le sur **SRV-WEB-02**).

2.1 Installation du moteur

Installez Apache et PHP avec **exactement** les mêmes extensions que lors du TP 1.

```
sudo apt install apache2 php php-xml php-common php-json php-mysql php-mbstring php-curl php-gd php-intl php-zip php-bz2 php-imap php-apcu php-ldap nfs-common -y
```

2.2 Montage du stockage partagé

Les serveurs Web doivent voir le dossier NFS comme s'il était local.

1. Créez le dossier : `sudo mkdir -p /var/www/html/glpi/files`
2. Montez le volume (test) :

```
sudo mount IP_SRV_INFRA:/srv/nfs/glpi_files /var/www/html/glpi/files
```

3. Persistance (fstab) : Ajoutez cette ligne dans /etc/fstab pour que le montage résiste au redémarrage :

```
IP_SRV_INFRA:/srv/nfs/glpi_files /var/www/html/glpi/files nfs defaults 0 0
```

2.3 Configuration GLPI & Base de données

Copiez le code source de GLPI dans /var/www/html/glpi.

Modifiez config/config_db.php pour pointer vers le SRV-INFRA :

```
class DB extends DBmysql {  
    public $dbhost = 'IP_SRV_INFRA'; // Et non plus localhost !  
    public $dbuser = 'glpi_remote';  
    public $dbpassword = 'password_solide';  
    public $dbdefault = 'glpidb';  
}
```

Étape 3 : Le chef d'orchestre (HAProxy sur SRV-INFRA)

C'est ici que la magie de la Haute Disponibilité opère.

1. **Installation** : `sudo apt install haproxy -y`
2. Configuration SSL (Rappel TP HTTPS) :

Haute disponibilité pour GLPI

C'est le Load Balancer qui va porter le certificat HTTPS (SSL Termination).

- Concaténez votre certificat et votre clé privée du TP précédent dans un seul fichier .pem :

```
cat glpi.crt glpi.key > /etc/haproxy/glpi.pem
```

3. Configuration HAProxy (/etc/haproxy/haproxy.cfg) :

```
frontend glpi-front
    bind *:80
    bind *:443 ssl crt /etc/haproxy/glpi.pem # On gère le HTTPS ici
    redirect scheme https if !{ ssl_fc }      # On force le HTTPS comme au
TP Sécurité
    default_backend glpi-backend

backend glpi-backend
    balance roundrobin                      # Un coup toi, un coup moi
    cookie SERVERID insert indirect nocache # Session persistante (Sticky)
    option httpchk GET /                   # Vérifie si le serveur Web
est vivant
    server web01 IP_WEB_01:80 check cookie srv1
    server web02 IP_WEB_02:80 check cookie srv2
```

4. Redémarrez HAProxy.

Étape 4 : Adaptation de la stratégie de sauvegarde

Rappelez-vous du TP "Sauvegarde Automatique". Le script backup_glpi.sh était sur le serveur Web.

Problème : Les serveurs Web ne contiennent plus les données ! La base et les fichiers sont sur SRV-INFRA.

Action : Vous devez déplacer et adapter votre script de sauvegarde sur **SRV-INFRA**.

- La commande mysqldump fonctionnera en local sur SRV-INFRA.
- La commande tar devra sauvegarder /srv/nfs/glpi_files au lieu de /var/www/html/glpi.
- N'oubliez pas de remettre le **CRON** sur cette machine.

5. Validation et crash test

Pour valider le TP, réalisez les tests suivants :

1. **Test de charge** : Ouvrez GLPI via https://IP_SRV_INFRA. Naviguez. Vérifiez dans les logs Apache des serveurs Web (tail -f /var/log/apache2/access.log) que les requêtes arrivent bien tantôt sur l'un, tantôt sur l'autre.
2. **Test de panne (Chaos Monkey)** :
 - Connectez-vous à GLPI.
 - Éteignez brutalement la VM **SRV-WEB-01**.
 - Cliquez sur un menu dans GLPI.
 - **Résultat attendu** : Aucune erreur. Vous avez basculé de façon transparente sur SRV-WEB-02.
3. **Test de l'Agent** :
 - Reprenez votre client du TP "Inventaire".
 - Pointez l'agent vers https://IP_SRV_INFRA/front/inventory.php.
 - L'inventaire doit remonter correctement, quel que soit le serveur Web qui traite la demande.