Procédure de mise en place d'une stratégie de sauvegarde 3-2-1

Ce document détaille la mise en œuvre d'une stratégie de sauvegarde 3-2-1 pour un serveur de fichiers Ubuntu virtualisé sur Proxmox VE, en utilisant les outils BorgBackup et `vzdump`.

Contexte de l'Infrastructure :

Composant	Nom	Rôle
Hyperviseur	`srv-prox01`	Proxmox VE
Machine Virtuelle (VM)	`vm-files01`	Ubuntu Server, Partage Samba
Dossier Critique	`/srv/partage`	Contient les données avec des ACLs

Objectif de la Stratégie 3-2-1 :

1. **3** copies de données (Original + 2 Sauvegardes).
2. **2** types de supports différents (Disque interne dédié + Disque USB externe).
3. **1** copie hors site/hors ligne (Disque USB externe, débranché après usage).

Table des matières

PARTIE 1 : Sauvegarde Quotidienne (Interne) avec BorgBackup

L'objectif est de sauvegarder les données du partage `/srv/partage` de `vm-files01` vers un disque physique dédié, en préservant impérativement les ACLs.

1. Configuration Proxmox (Passthrough de Disque)

Cette étape permet d'attribuer un disque physique du serveur hôte (`srv-prox01`) directement à la VM `vm-files01`, garantissant performance et simplicité de gestion.

1.1. Identification du Disque Physique

Connectez-vous en SSH à l'hôte Proxmox (`srv-prox01`).

1. Listez les périphériques par ID pour une identification stable et récupérer le nom du disque voulue (appuyez-vous sur l'interface web avec ***nœud***=>***Disks*** au besoin) :
   ls -l /dev/disk/by-id/ | head

Exemple de résultat : `ata-WDC_WD5000LPVX-08V0TT5_WD-WX61AC3U2707`

2. Identifiez le chemin complet du disque à utiliser (ex: `/dev/disk/by-id/ata-WDC_WD5000LPVX-08V0TT5_WD-WX61AC3U2707`).

1.2. Attachement du Disque à la VM (Passthrough)

Utilisez la commande `qm set` pour attacher le disque directement à la VM 101 (`vm-files01`). Nous utilisons `scsi0` comme bus.
Syntaxe : qm set <***VMID***> -scsi<***N***> /dev/disk/by-id/<***DISK_ID***>

qm set 101 -scsi0 /dev/disk/by-id/ata-WDC_WD5000LPVX-08V0TT5_WD-WX61AC3U2707

2. Préparation de la VM (Ubuntu)

Connectez-vous en SSH à la VM cible (`vm-files01`). Le disque devrait apparaître comme `/dev/sda`, `/dev/sdb`, etc. (probablement `/dev/sdb` si `/dev/sda` est le disque système).

2.1. Formatage en EXT4

ATTENTION : Cette commande effacera TOUTES les données existantes sur ce disque. Assurez-vous d'avoir le bon périphérique (ici, `/dev/sdb`).

Vérifiez d'abord :
sudo fdisk -l

Si le disque est /dev/sdb (à adapter si nécessaire) :
sudo mkfs.ext4 -F /dev/sdb

2.2. Montage Permanent (fstab)

1. Créez le point de montage : ***sudo mkdir -p /mnt/backup-disk***
2. Récupérez l'UUID du nouveau disque pour un montage stable : ***sudo blkid /dev/sdb***

Exemple : `UUID=" a85061dd-79ca-43c4-ba02-78202f9db66e"`

3. Éditez `/etc/fstab` pour ajouter une entrée de montage automatique. Remplacez l'UUID par le vôtre : ***sudo nano /etc/fstab***

Ajoutez la ligne suivante :
UUID=a85061dd-79ca-43c4-ba02-78202f9db66e /mnt/backup-disk ext4 defaults 0 2

4. Montez le disque :
   sudo mount -a
   df -h /mnt/backup-disk

3. Installation & Configuration de BorgBackup

3.1. Installation de BorgBackup et Zabbix sender

1. Télécharger le paquet de configuration du dépôt Zabbix (version 7.0 LTS ou 6.4)
wget https://repo.zabbix.com/zabbix/7.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ubuntu24.04_all.deb
2. Installer le dépôt
sudo dpkg -i zabbix-release_latest+ubuntu24.04_all.deb
3. Mettre à jour la liste des paquets
sudo apt update && sudo apt upgrade -y
4. Installer les outils
sudo apt install borgbackup fuse3 python3-pyfuse3 zabbix-sender -y

3.2. Initialisation du Dépôt (Repository)

Nous initialisons le dépôt sur le disque dédié (`/mnt/backup-disk`).
Note : Il est recommandé d'utiliser une passphrase forte pour protéger le dépôt.
Initialisation du dépôt. La commande demandera la passphrase.
sudo borg init --encryption=repokey /mnt/backup-disk/borg-repo

4. Le Script de Sauvegarde Quotidienne

Ce script garantit la préservation des ACLs et met en œuvre la politique de rétention.

4.1. Création du Script

Créez le fichier `/usr/local/bin/backup-borg.sh` et rendez-le exécutable.
sudo nano /usr/local/bin/backup-borg.sh

Script `/usr/local/bin/backup-borg.sh` :

```
#!/bin/bash
# Configuration
REPO="/mnt/backup-disk/borg-repo"
SOURCE="/srv/partage"
LOGFILE="/var/log/borg-backup.log"
DATE=$(date +%Y-%m-%d_%H-%M-%S)
ARCHIVE_NAME="vm-files01-$DATE"
```

```
PASSPHRASE="passphrase" # REMPLACEZ MOI !
ZABBIX_SERVER="FQDN_Zabbix" # REMPLACEZ MOI !
HOST_NAME="vm-files01"
echo "--- Démarrage de la sauvegarde Borg : $DATE ---" >> $LOGFILE
# Sauvegarde avec préservation des ACLs
sudo BORG_PASSPHRASE=$PASSPHRASE borg create \
    --stats \
    --progress \
    --compression zstd,5 \
    $REPO::$ARCHIVE_NAME \
    $SOURCE 2>&1
EXIT_CODE=$?
if [ $EXIT_CODE -eq 0 ]; then
    echo "Sauvegarde $ARCHIVE_NAME réussie." >> $LOGFILE
    zabbix_sender -z "$ZABBIX_SERVER" -s "$HOST_NAME" -k "borg.backup.status" -o 0
    # Politique de rétention
    # (7 jours, 4 semaines, 12 mois)
    echo "Démarrage de la rétention..." >> $LOGFILE
    sudo BORG_PASSPHRASE=$PASSPHRASE borg prune \
        --list \
        --stats \
        $REPO \
        --keep-daily 7 \
        --keep-weekly 4 \
        --keep-monthly 12 >> $LOGFILE 2>&1
    echo "Rétention terminée. Sauvegarde réussie." >> $LOGFILE
else
    echo "ERREUR : La sauvegarde Borg a échoué (Code $EXIT_CODE)." >> $LOGFILE
    zabbix_sender -z "$ZABBIX_SERVER" -s "$HOST_NAME" -k "borg.backup.status" -o 1
fi
echo "--- Fin de la sauvegarde Borg ---" >> $LOGFILE
```

Rendez le script exécutable : ***sudo chmod +x /usr/local/bin/backup-borg.sh***

4.2. Automatisation avec CRON

Planifiez l'exécution du script tous les jours à 22h00.

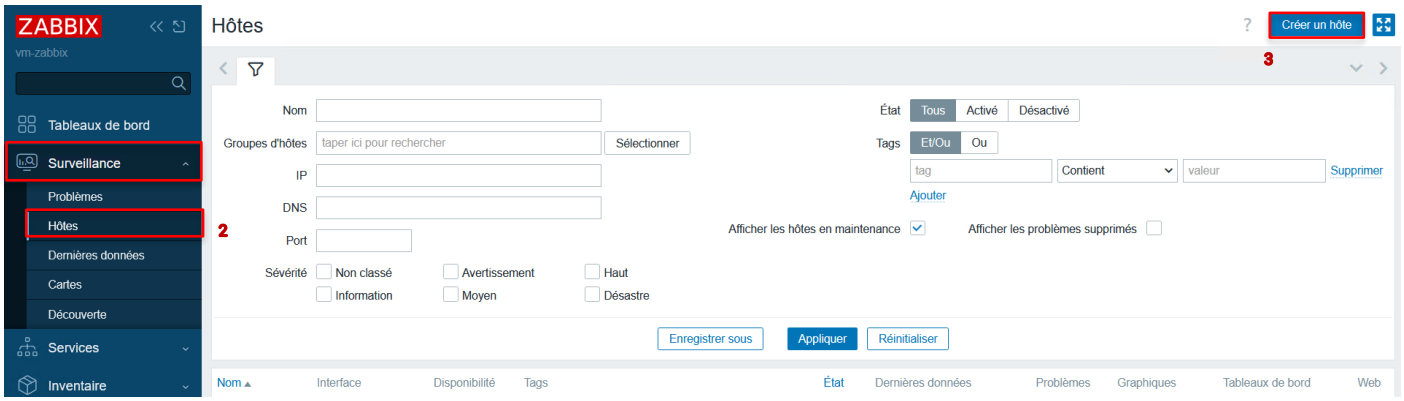
1. Ouvrez la table cron de l'utilisateur root : ***sudo crontab -e***
2. Ajoutez la ligne suivante à la fin du fichier :

```
# Sauvegarde quotidienne des fichiers à 22h00
0 22 * * * /usr/local/bin/backup-borg.sh
```

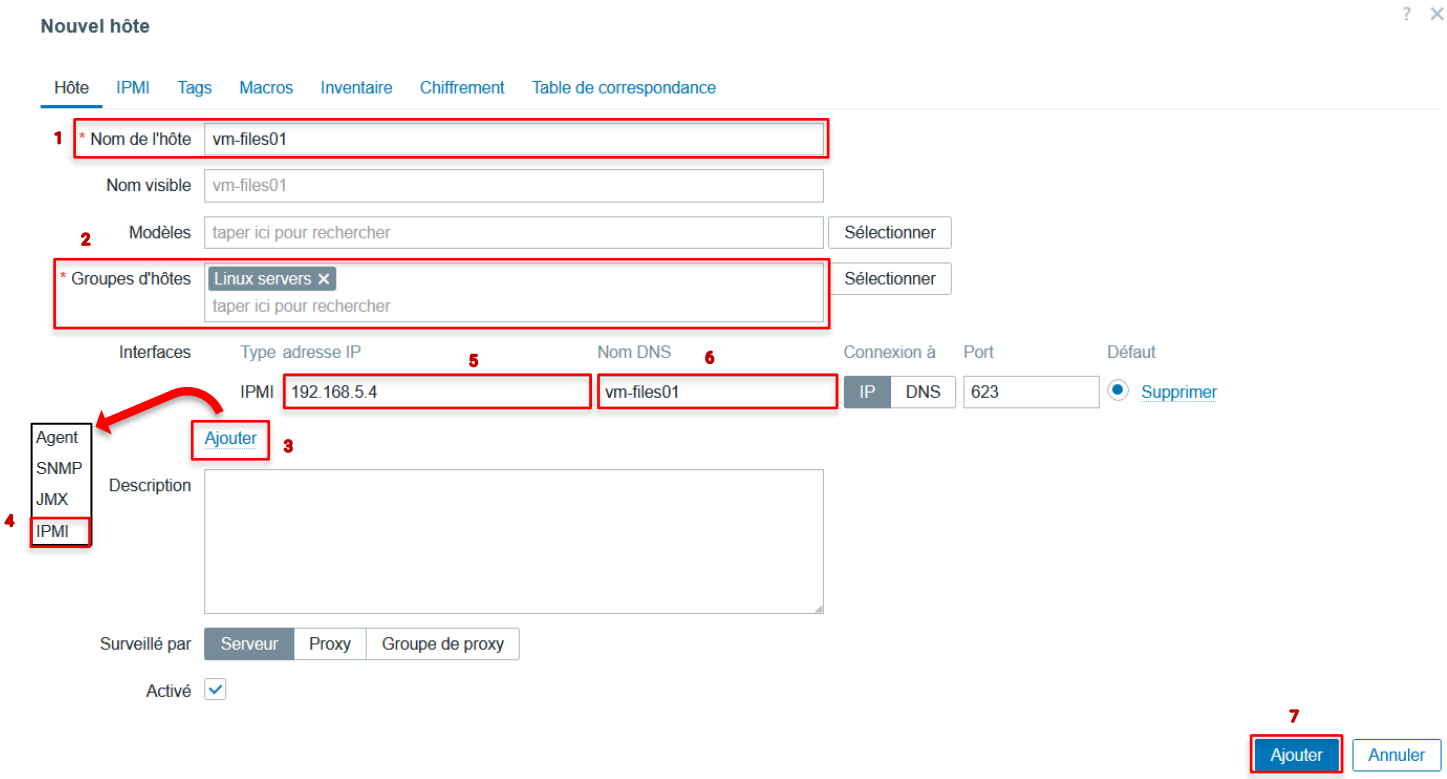
5. Configuration sur l'interface Web Zabbix

Nous devons configurer Zabbix afin de savoir comment réceptionner les données envoyées par nôtre précédent script.

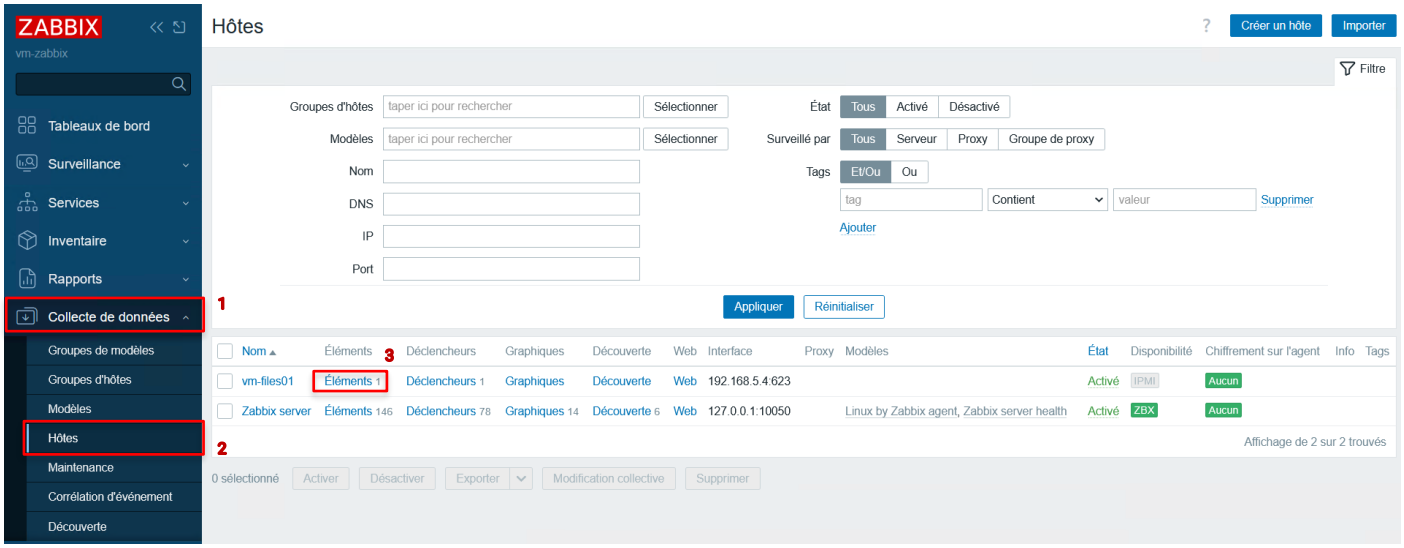
1. Création de l'Hôte
    - Allez sur le bandeau de gauche : ***Surveillance*** (1) > ***Hôtes*** (2) > ***Créer un hôte*** (3)



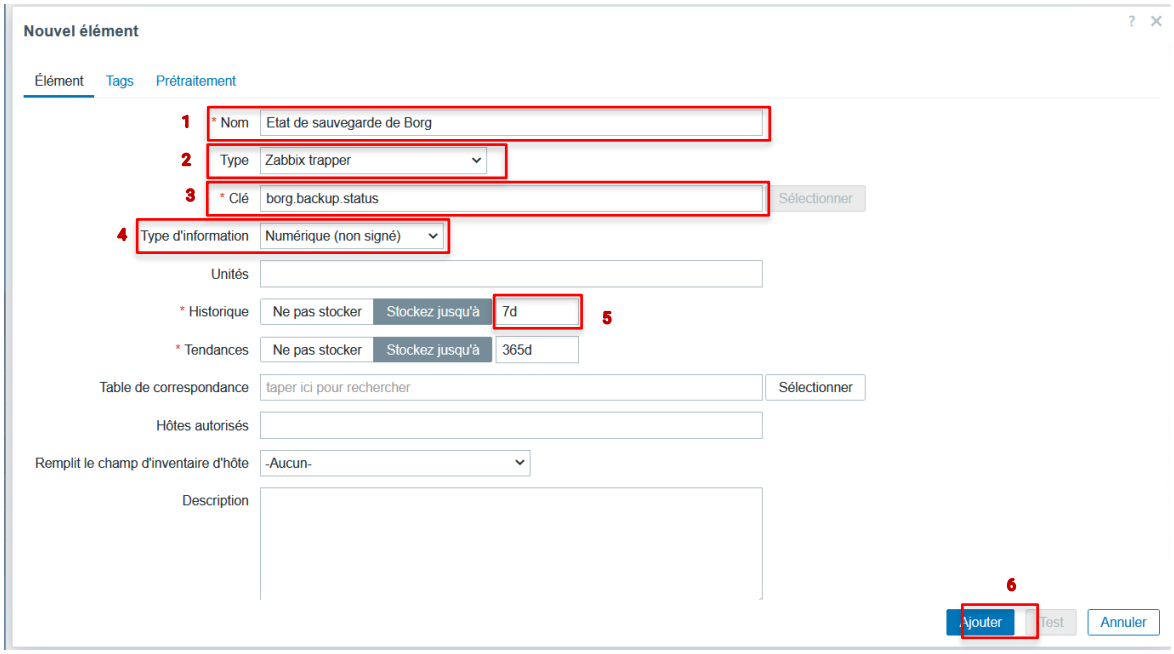
    - Nom de l'Hôte (1) : ***FQDN_SRV-FICHIER***
    - Groupes d'hôtes (2): ***Linux servers***
    - Interfaces : cliquer sur ***Ajouter*** (3)> ***IPMI*** (4)
        - Adresse ip (5): ***IP_SRV-FICHIER***
        - Nom DNS (6): ***FQDN_SRV-FICHIER***
    - ***Ajouter*** (7)



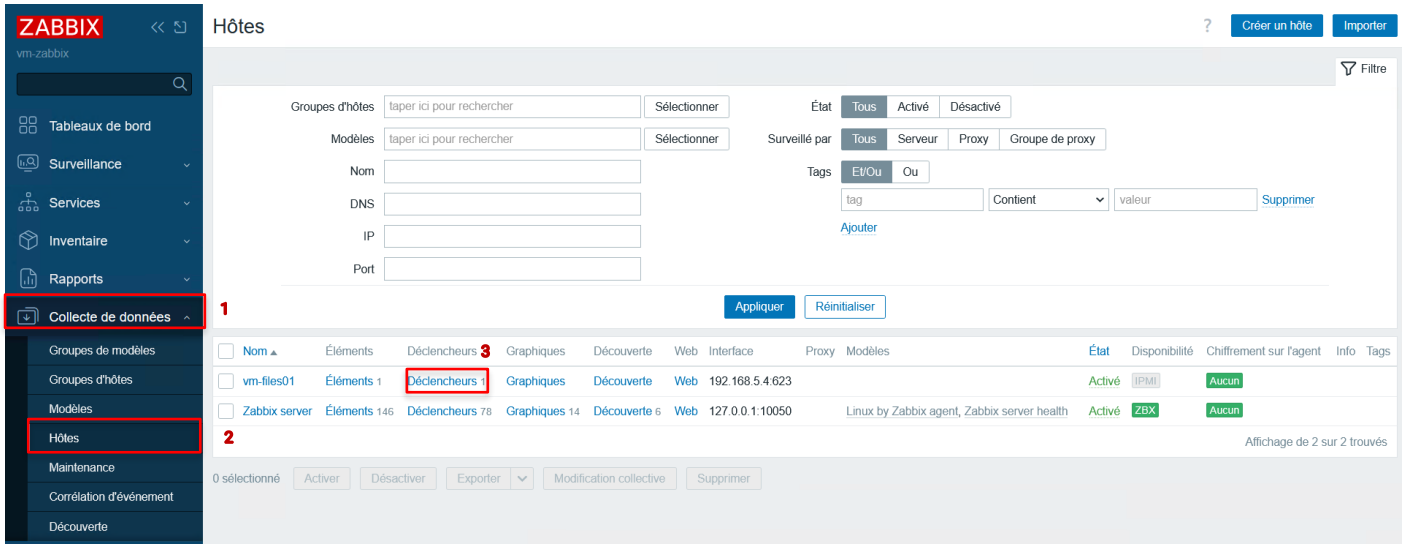
2. Création de l'éléments
   - Allez sur le bandeau de gauche : ***Collecte de données*** (1) > ***Hôtes*** (2) > ***Éléments*** (3) (au niveau de vm-files01)



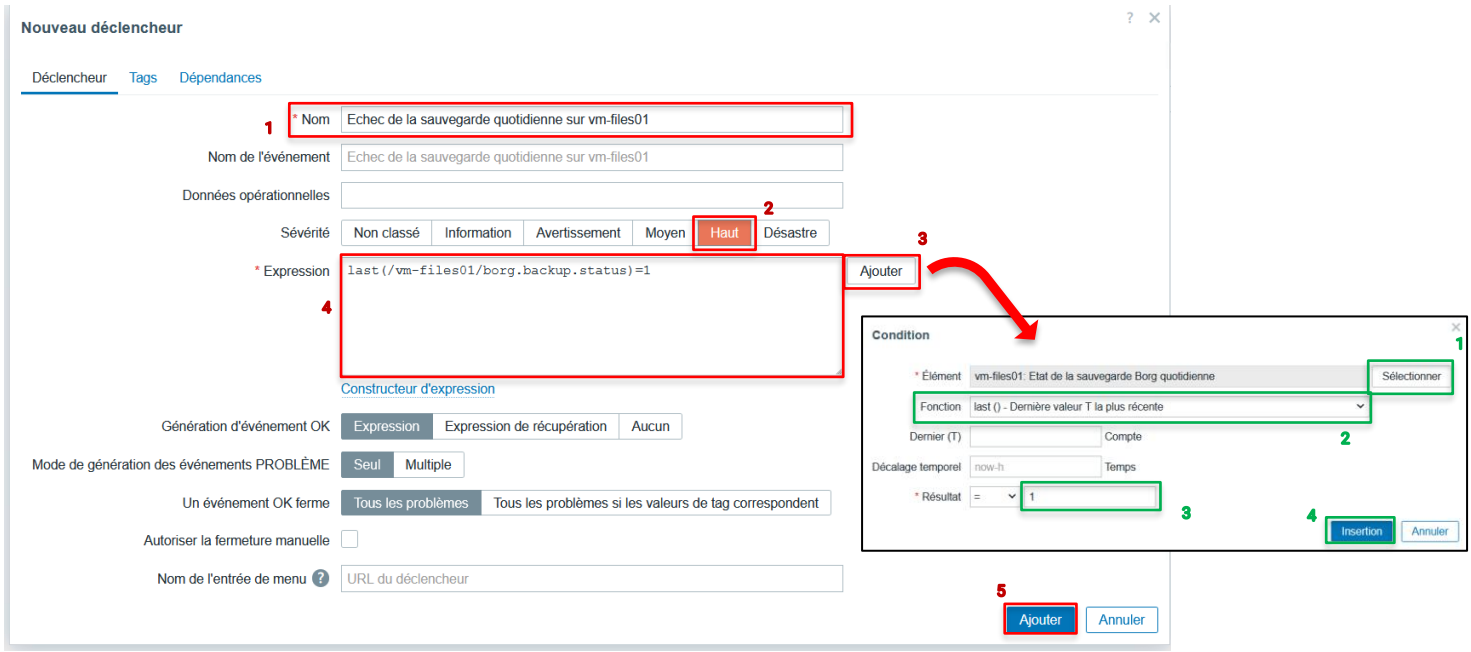
   - ***Créer un élément*** (en haut à gauche)
     - Nom (1) : ***Etat de sauvegarde de Borg***
     - Type (2) : ***Zabbix trapper***
     - Clés (3) : ***borg.backup.status***
     - Type d'information (4) : ***Numérique (non signé)***
     - Historique (5) : ***7d*** (7 jours)
     - ***Ajouter*** (6)



3. Création du déclencheur
    - Allez sur le bandeau de gauche : ***Collecte de données*** (1) > ***Hôtes*** (2) > ***Déclencheurs*** (3) (au niveau de vm-files01)



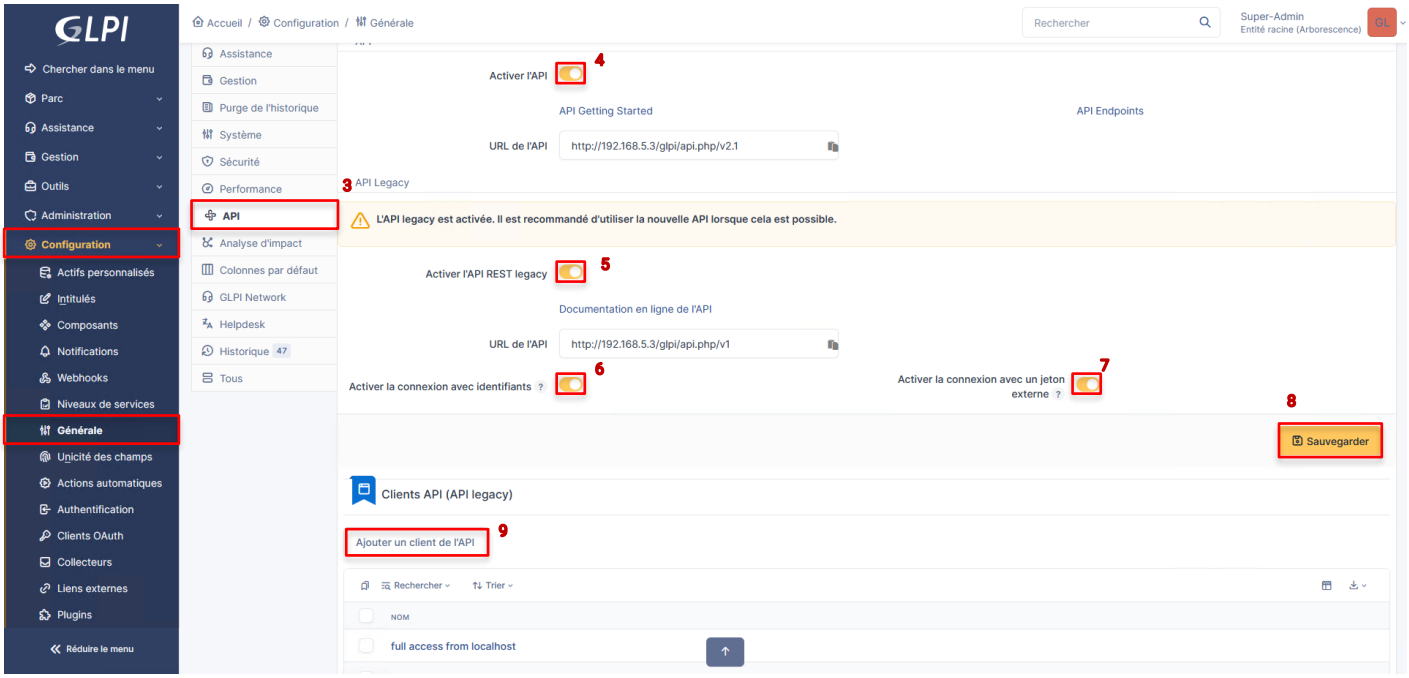
- ***Créer un déclencheur*** (en haut à gauche)
    - Nom (1) : ***Echec de la sauvegarde quotidienne sur FDND_SRV-FILE***
    - Sévérité (2) : ***Haut***
    - Expression (3) : ***Ajouter***
        - Élément : **Sélectionner** (1) > ***Etat de la sauvegarde borg quotidienne***
        - Fonction (2) : ***last () – Dernière valeur T la plus récente***
        - Résultat (3) : 1
        - ***Insertion*** (4)
        - Le résultat final doit être (4): `last(/vm-files01/borg.backup.status)=1`
    - ***Ajouter*** (5)



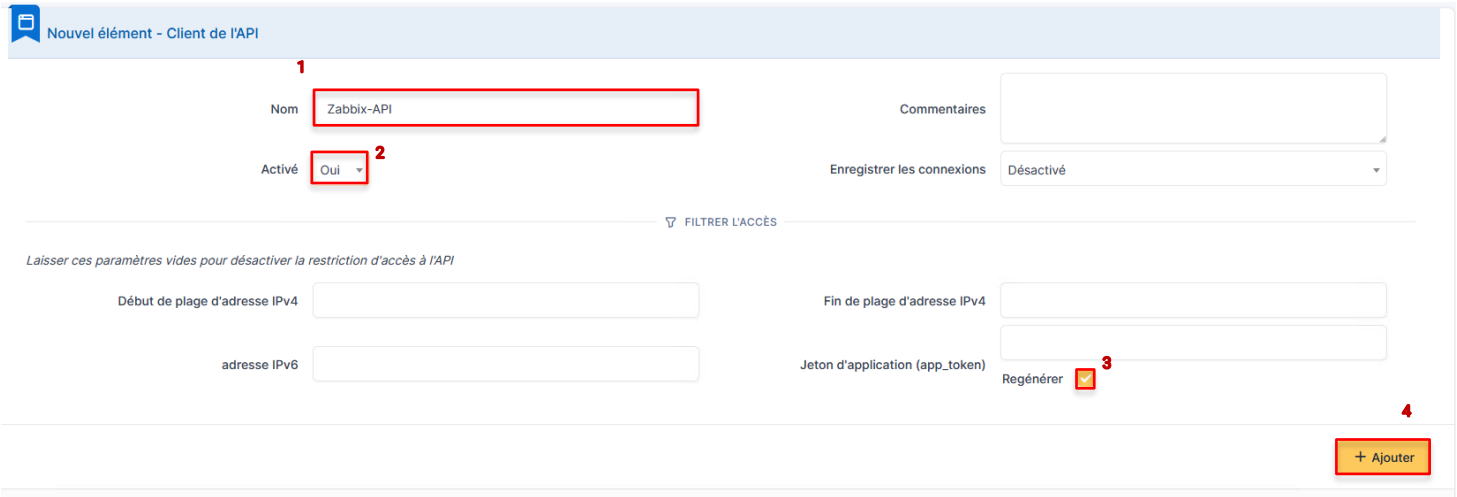
6. Configuration de GLPI pour la création automatique de ticket

Nous devons configurer GLPI afin de permettre la création de ticket dès qu'un incident est déclaré. **Sur GLPI :**

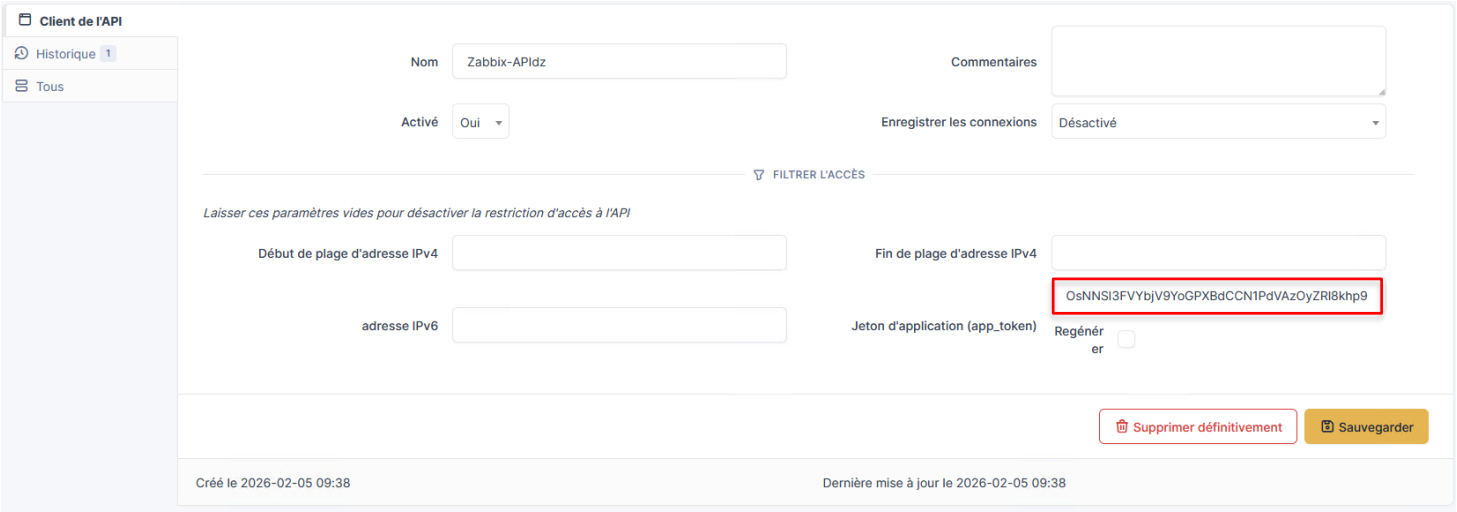
1. Activer l'API
    - Allez sur le bandeau de gauche : ***Configuration*** (1) > ***Générale*** (2) > ***API*** (3). Cocher ***Activer l'API*** (4), ***Activer l'API REST legacy*** (5), ***Activer la connexion avec identifiants*** (6) et ***Activer la connexion avec un jeton externe*** (7). Cliquer sur ***Sauvegarder*** (8) et ***Ajouter un client de l'API*** (9).



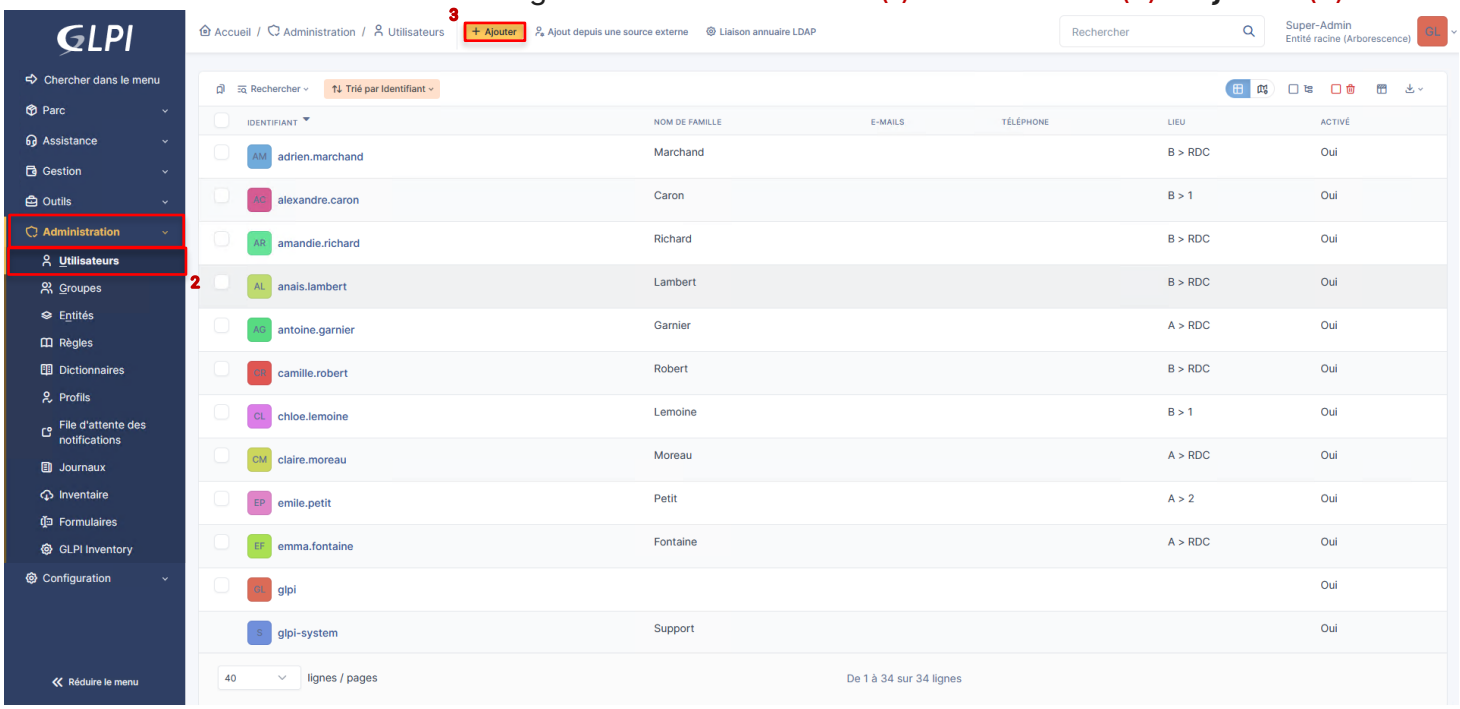
    - Nom : ***Zabbix-API*** (1)
    - Actif : ***Oui*** (2)
    - Laisser coche ***Régénérer le jeton (App-Token)*** (3)
    - Cliquer sur ***Ajouter*** (4)



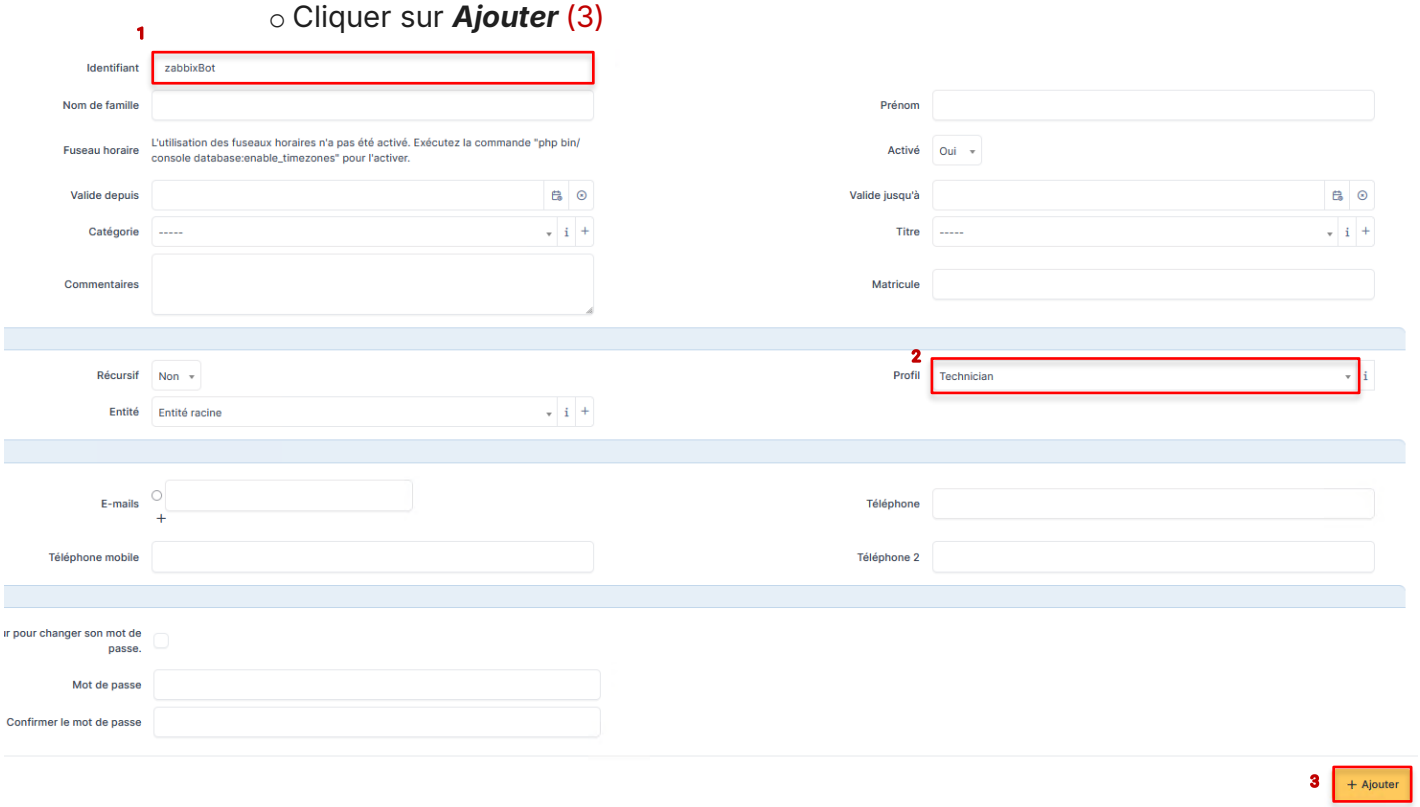
- Copier le jeton et **le garder de côté**



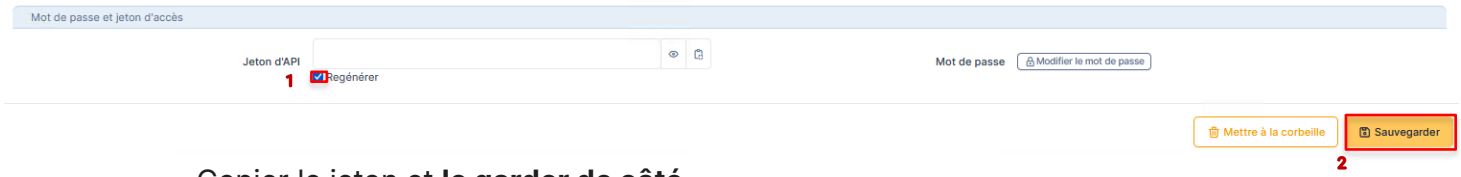
2. Créer un utilisateur "Bot"
   - Allez sur le bandeau de gauche : ***Administration*** (1) > ***Utilisateurs*** (2) > ***Ajouter*** (3)



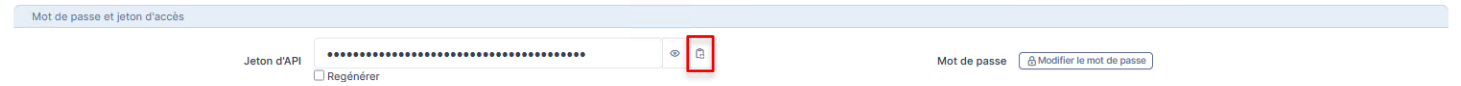
- Identifiant (1) : ***zabbixBot***
- Profil (2) : ***Technician***
- Cliquer sur ***Ajouter*** (3)



- En bas, cliquer sur ***Regénérer*** (1) puis ***Sauvegarder*** (2)



- Copier le jeton et **le garder de côté**



Sur Zabbix :

Nous allons créer un petit script afin d'envoyer des infos à GLPI depuis Zabbix

1. Créer le script d'alerte
   - Se connecter en SSH sur le serveur Zabbix
   - Ce placer au répertoire des scripts d'alerte : ***cd /usr/lib/zabbix/alertscripts/***
   - Créer le script d'alerte : ***nano glpiTicket.sh***

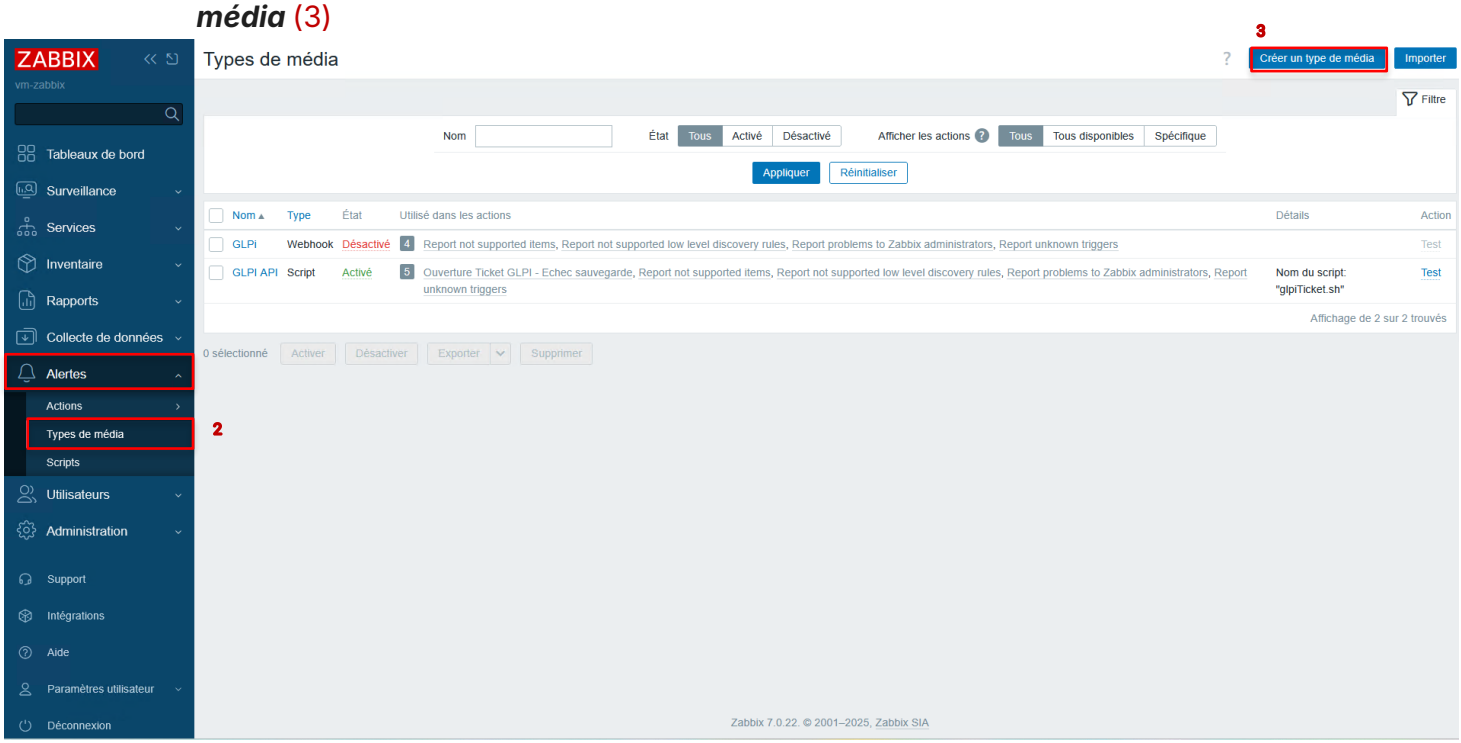
Script `/usr/lib/zabbix/alertscripts/glpiTicket.sh` :

```
#!/bin/bash
# Configuration
URL="http://IP_GLPI/apirest.php"
APP_TOKEN="TOKEN_DU_CLIENT_API"
USER_TOKEN="TOKEN_DU_BOT"

# Variables envoyées par Zabbix
SUBJECT=$(echo "$1" | tr -d '"' | tr '\n' ' ' | tr '\r' ' ')
MESSAGE=$(echo "$2" | tr -d '"' | tr '\n' ' ' | tr '\r' ' ')

# Initialisation de la session (Récupération du Session Token)
SESSION_TOKEN=$(curl -s -X GET \
-H "Content-Type: application/json" \
-H "App-Token: $APP_TOKEN" \
-H "Authorization: user_token $USER_TOKEN" \
"$URL/initSession" | grep -oP '(?<="session_token":")[^"]*')

# Création du ticket
curl -s -X POST \
-H "Content-Type: application/json" \
-H "App-Token: $APP_TOKEN" \
-H "Session-Token: $SESSION_TOKEN" \
-d "{ \"input\": { \"name\": \"$SUBJECT\", \"content\": \"$MESSAGE\", \"status\": 1, \"urgency\": 4 } } " \
"$URL/Ticket"

# Fermeture de session
curl -s -X GET \
-H "Content-Type: application/json" \
-H "App-Token: $APP_TOKEN" \
-H "Session-Token: $SESSION_TOKEN" \
"$URL/killSession"
```

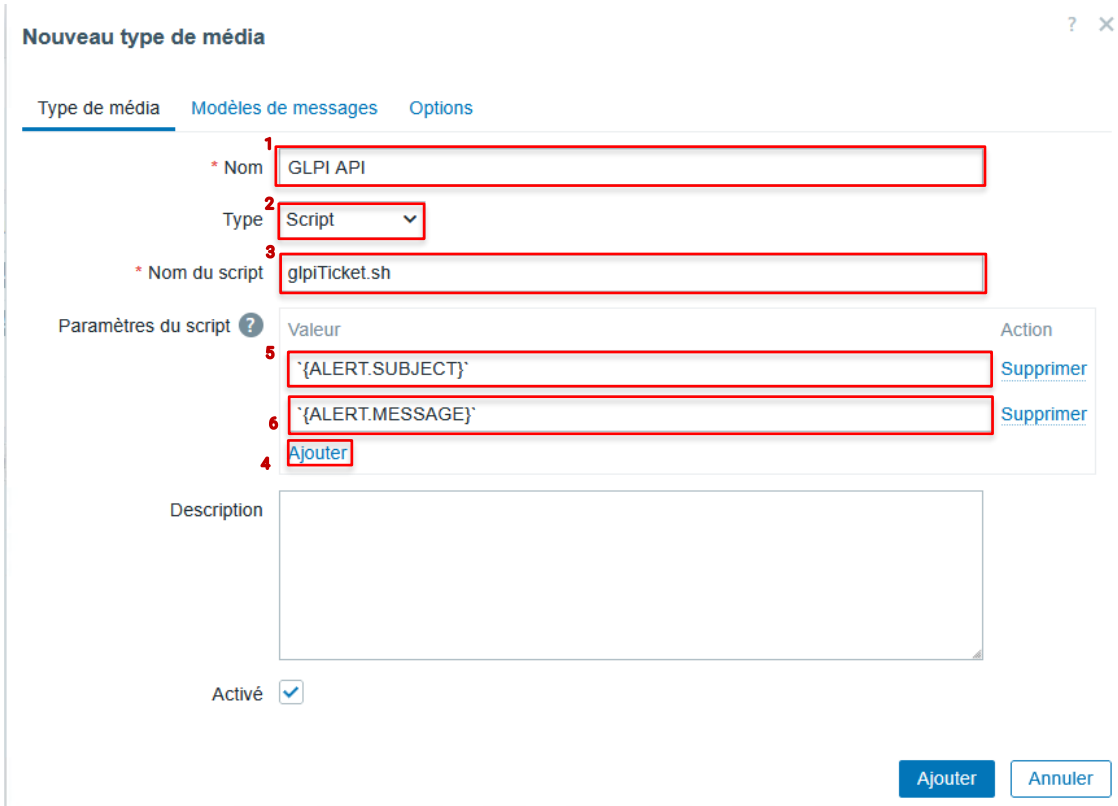
- Rendre le script executable : ***chmod +x glpiTicket.sh***

Configurons maintenant Zabbix pour exécuter se script dès que le problème survient **sur l'interface web de Zabbix**.

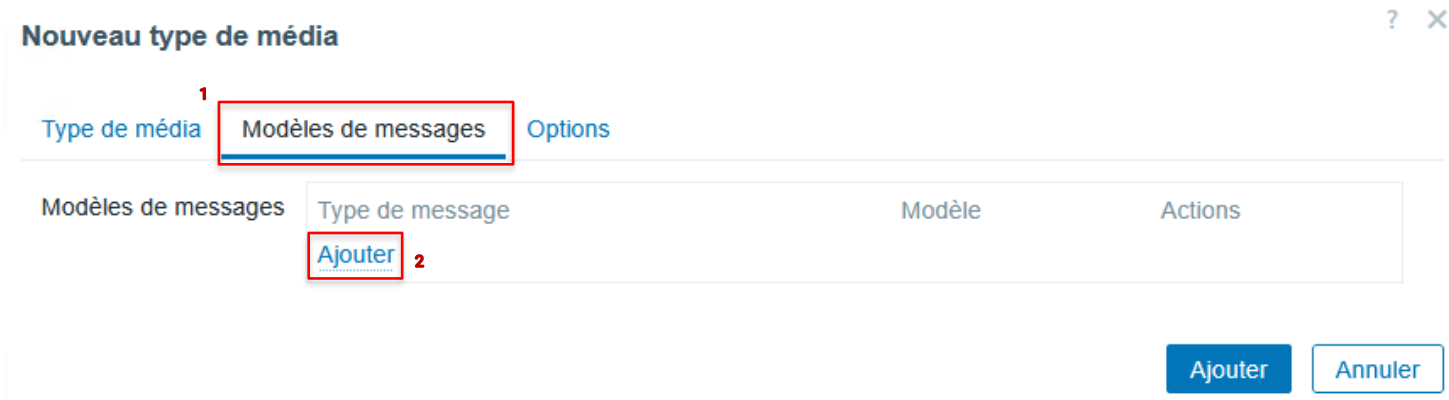
2. Créer le "Type de média"
    - Allez sur le bandeau de gauche : ***Alertes*** (1) > ***Types de média*** (2) > ***Créer un type de média*** (3)



    - Nom (1) : ***GLPI API***
    - Type (2) : ***Script***
    - Nom du script (3) : ***glpiTicket.sh*** (même nom que le script créer plus tôt)
    - Paramètres du script (ajouter ces 2 variables **dans l'ordre**) : cliquer sur ***Ajouter*** (4)
        1. (5) `{ALERT.SUBJECT}`
        2. (6) `{ALERT.MESSAGE}`

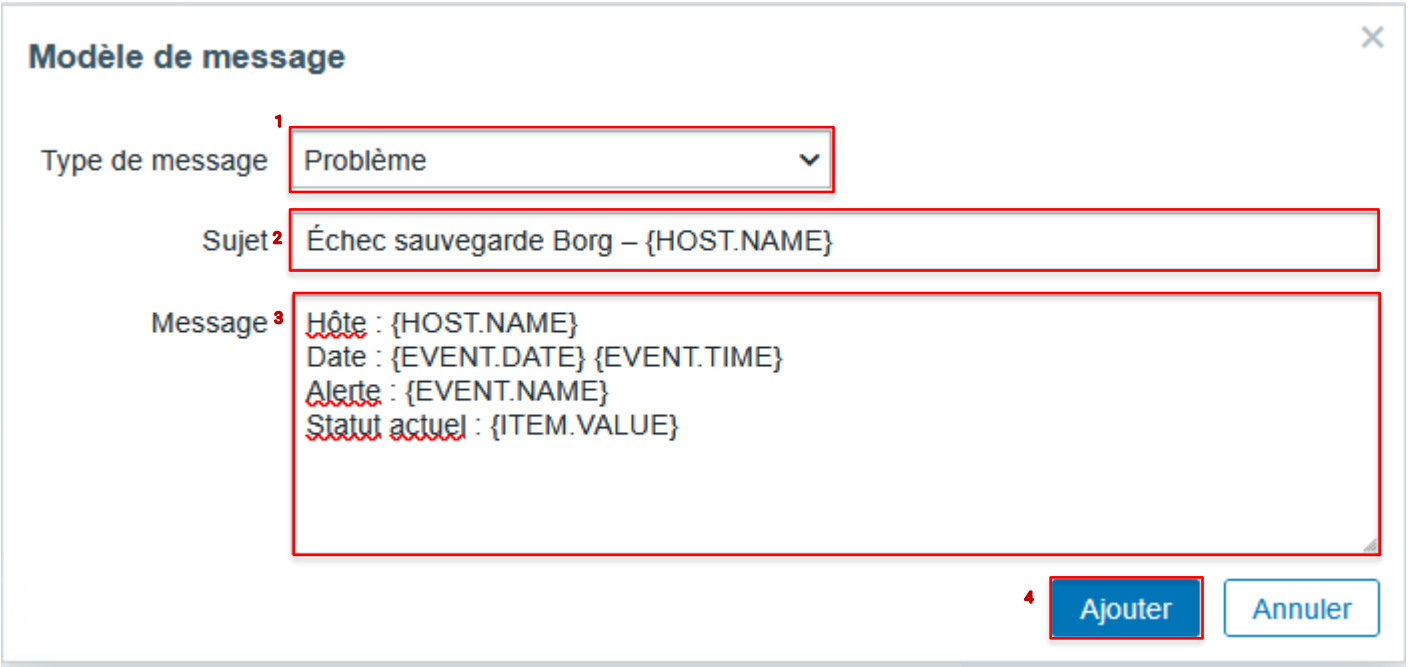


- Cliquer sur ***Modèles de messages*** (1) > ***Ajouter*** (2)

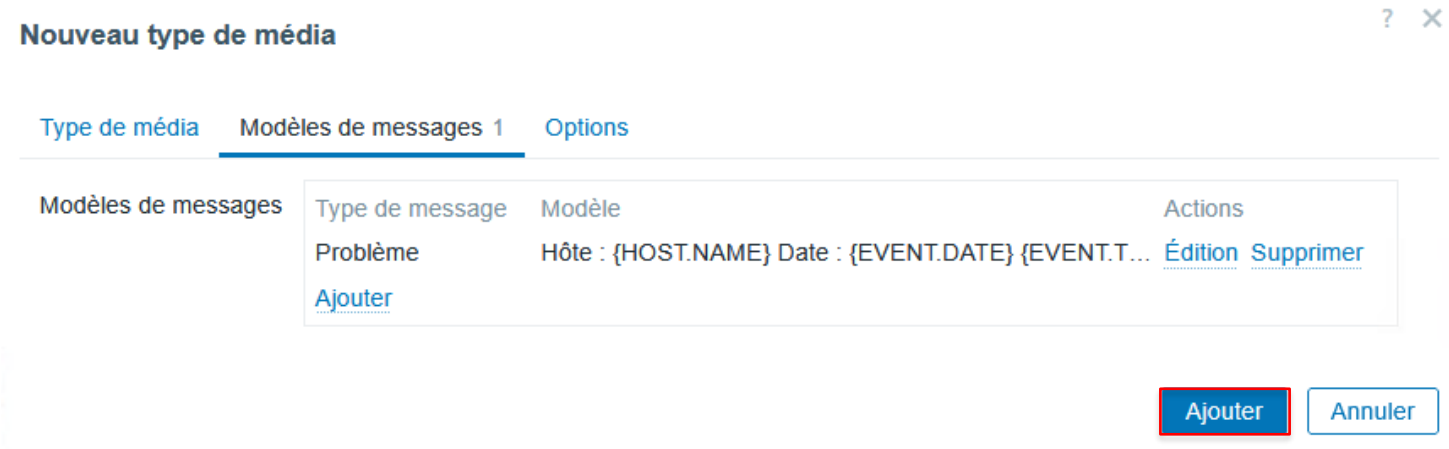


1. Type de message (1) : ***Problème***
2. Sujet (2) : ***Échec sauvegarde Borg – {HOST.NAME}***
3. Message (3) :

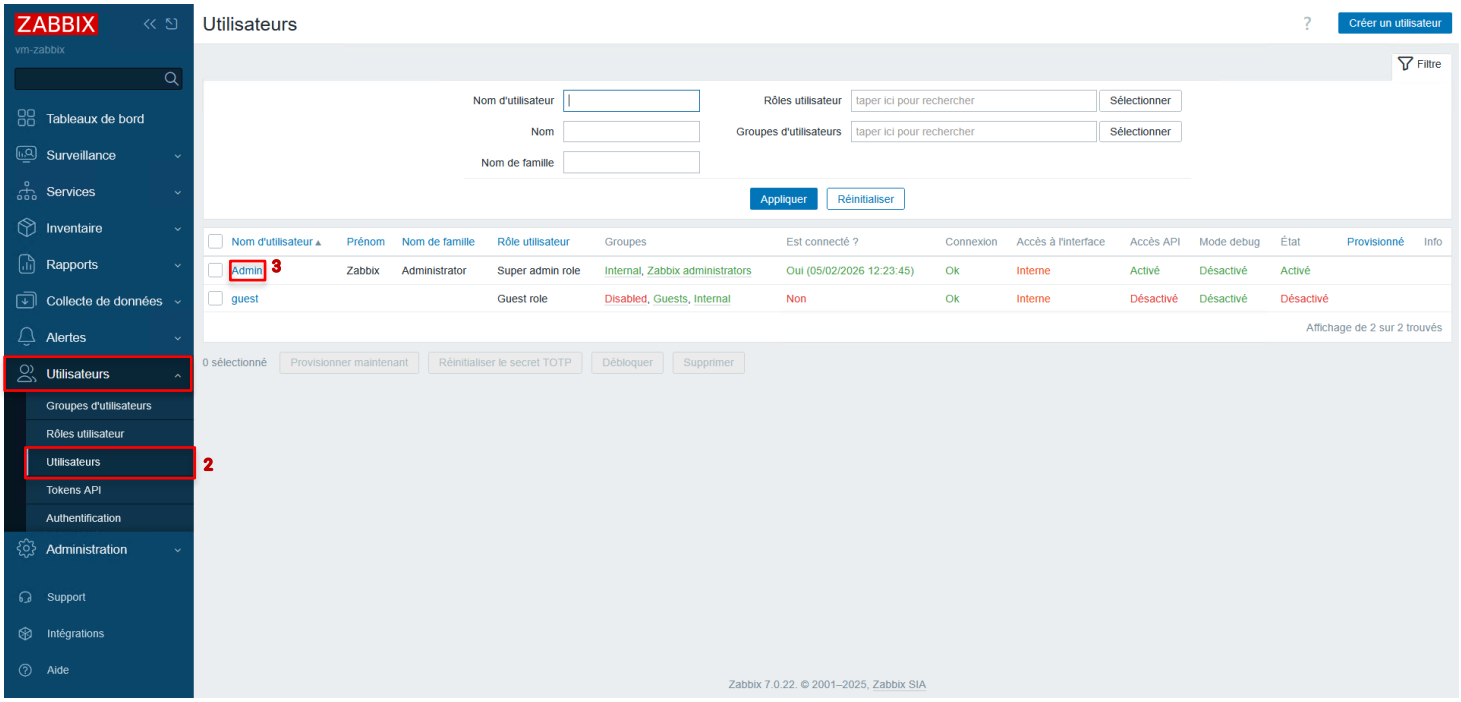
   Hôte : {HOST.NAME}
   Date : {EVENT.DATE} {EVENT.TIME}
   Alerte : {EVENT.NAME}
   Statut actuel : {ITEM.VALUE}
4. Cliquer sur ***Ajouter*** (4)



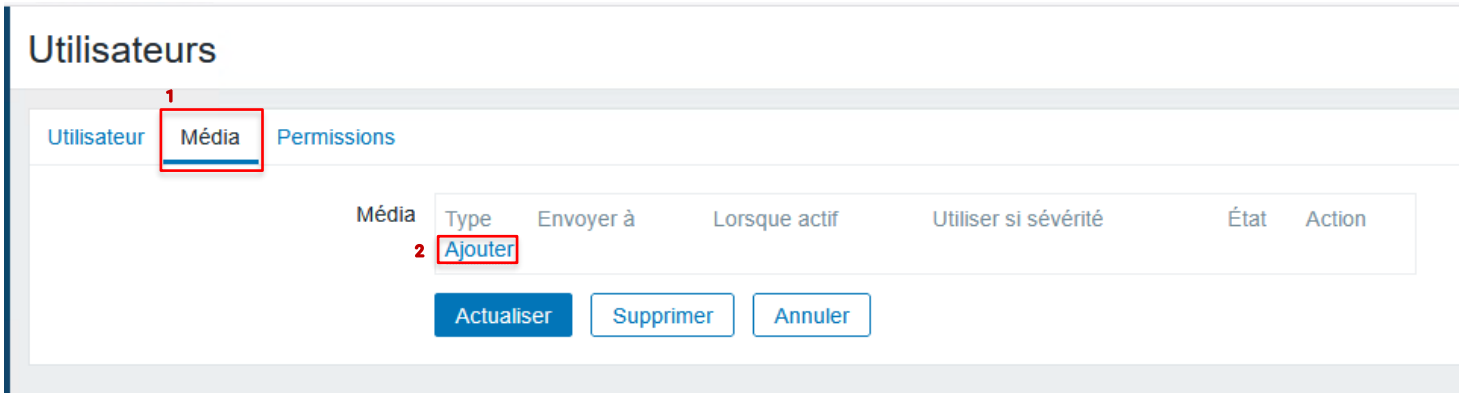
- Cliquer sur ***Ajouter***



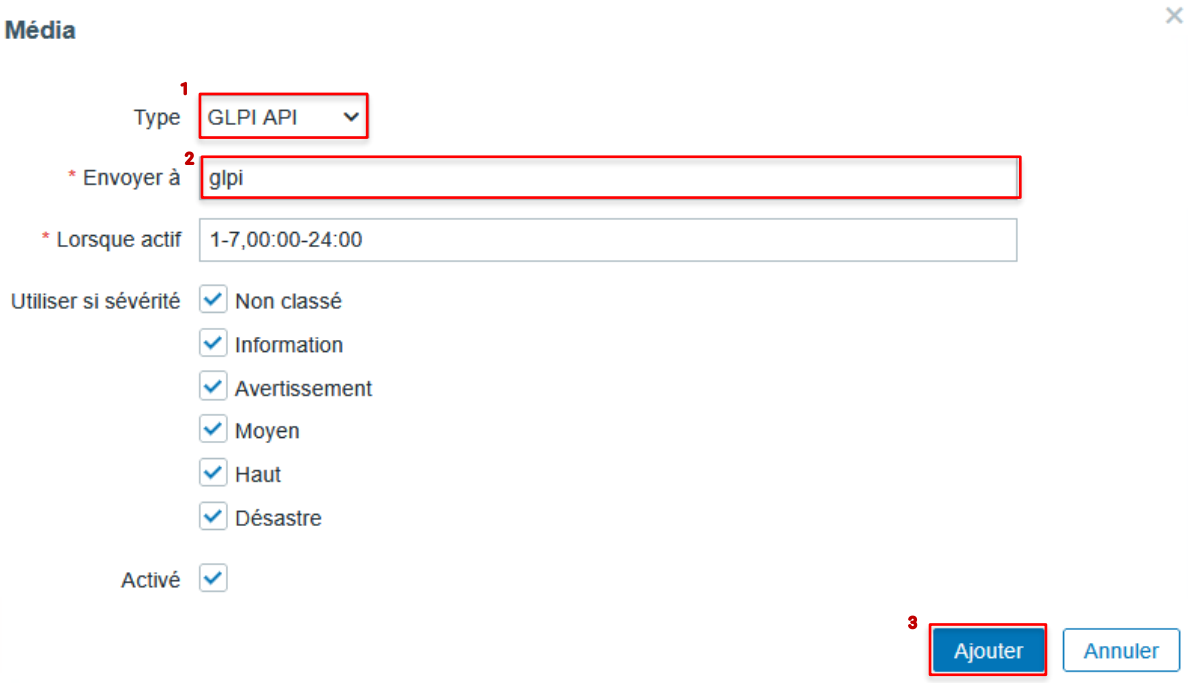
3. Associer le média à l'administrateur
   - Allez sur le bandeau de gauche : ***Utilisateurs*** (1) > ***Utilisateurs*** (2) > ***Admin*** (3)



   - Cliquer sur ***Média*** (1) > ***Ajouter*** (2)

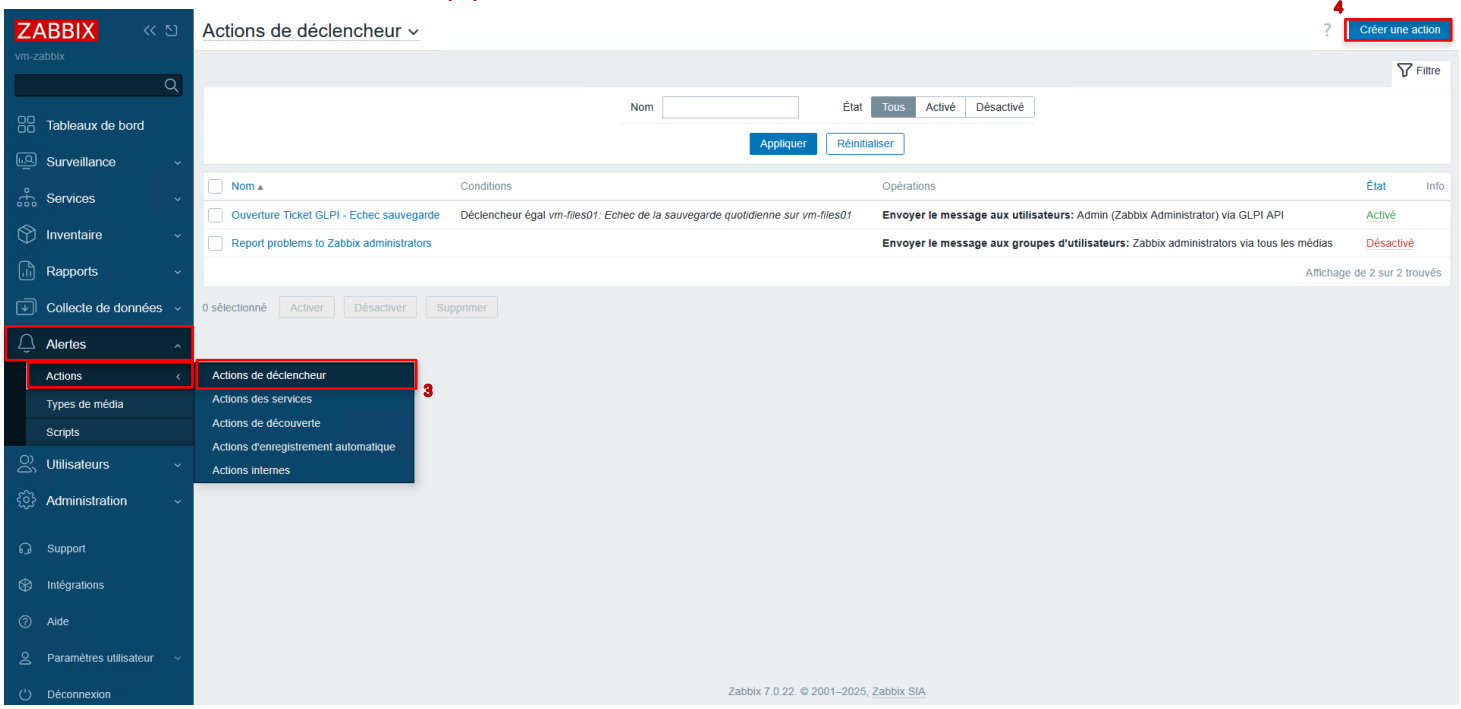


     - Type (1) : ***GLPI API***
     - Envoyer à (2) : ***glpi*** (ou ce que vous voulez, cette valeur est non utilisée)
     - Cliquer sur ***Ajouter*** (3)

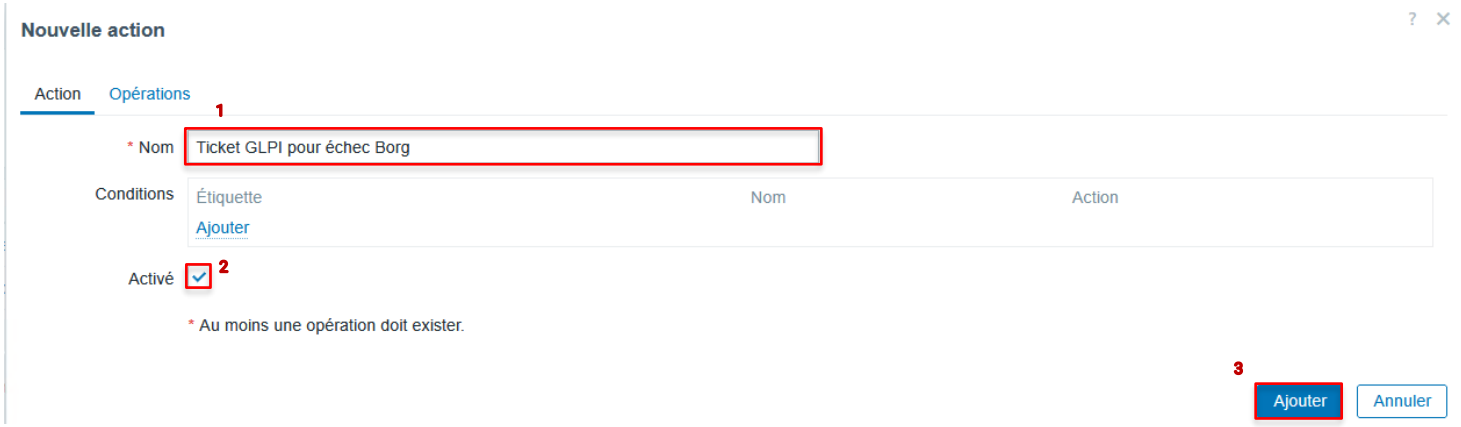


- Cliquer sur ***Actualiser***

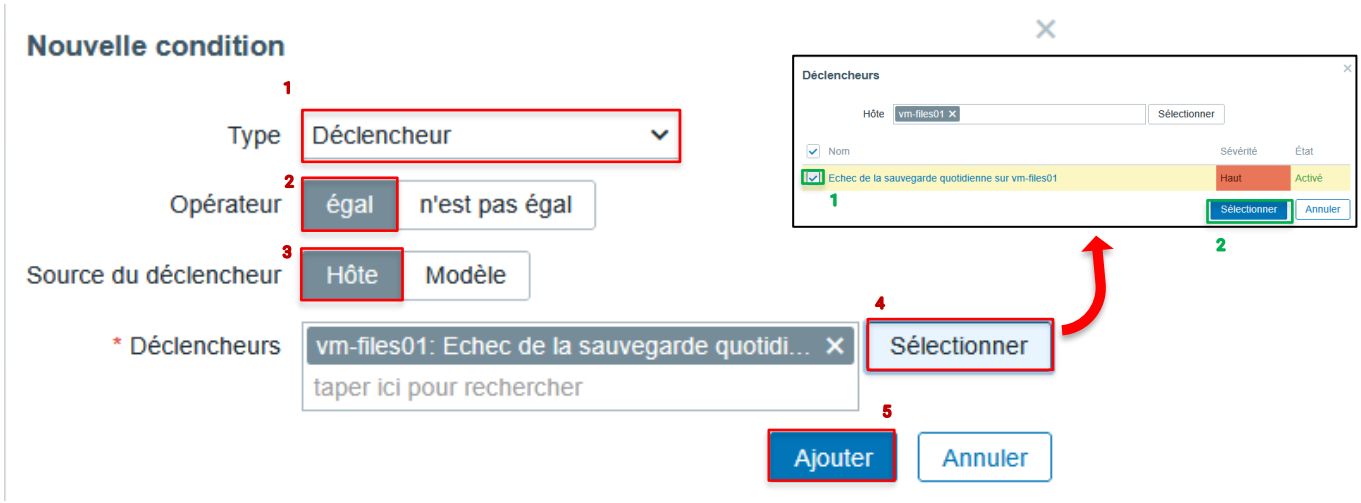
4. Créer l'Action
   - Allez sur le bandeau de gauche : ***Alertes*** (1) > ***Actions*** (2) > ***Actions de déclencher*** (3) > ***Créer une action*** (4)



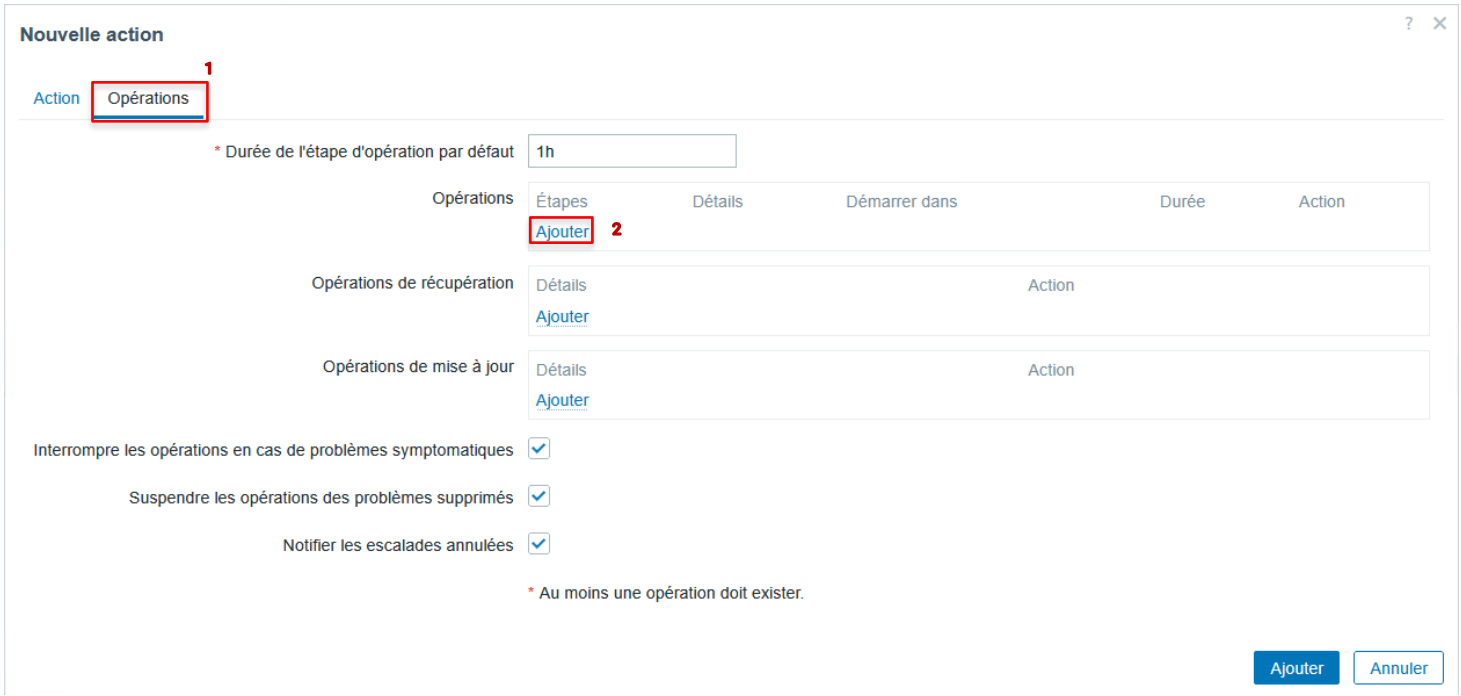
   - Nom (1) : ***Ticket GLPI pour échec Borg***
   - Activé (2) : ***Oui***
   - Condition (3) : ***Ajouter***



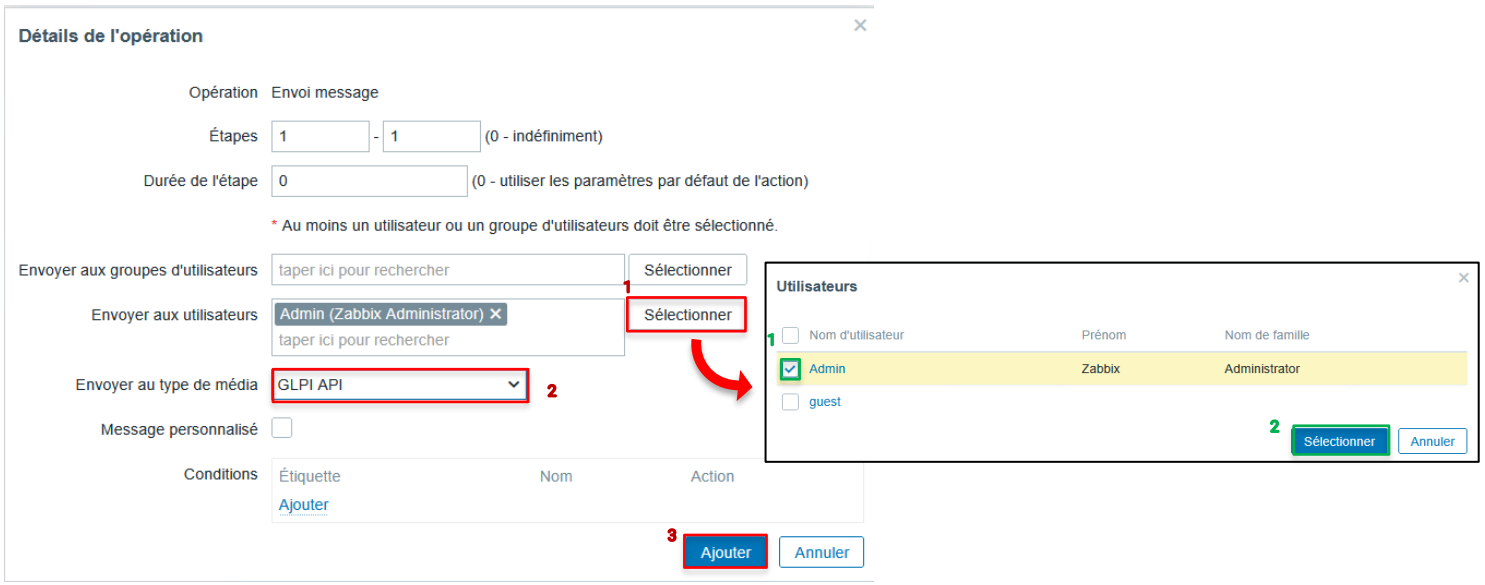
1. Type (1) : ***Déclencheur***
2. Opérateur (2) : ***égal***
3. Source du déclencheur (3) : ***Hôte***
4. Déclencheurs (4) : ***Sélectionner*** > Cocher ***Echec de la sauvegarder quotidienne sur FQDN_SRV-FILE*** (1) > ***Sélectionner*** (2)
5. Cliquer sur ***Ajouter*** (5)



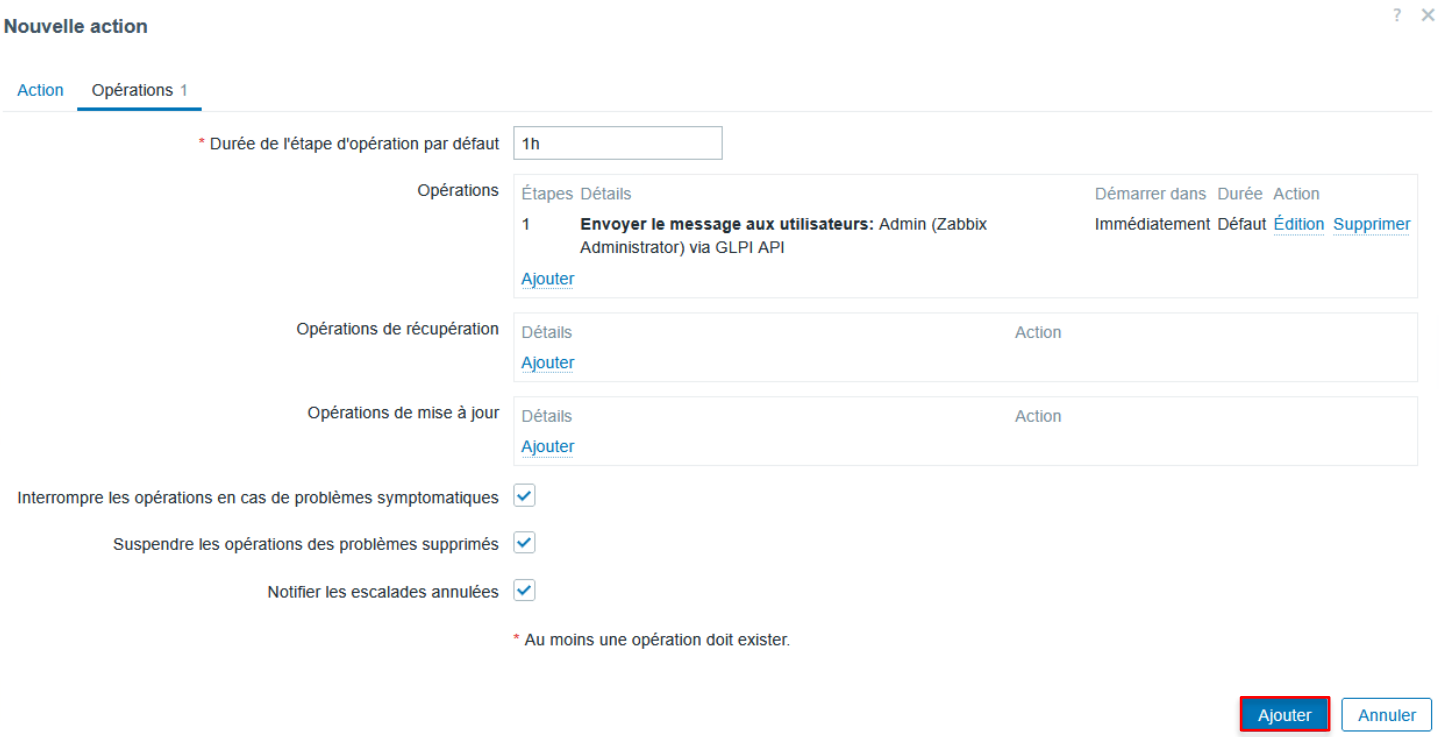
- Cliquer sur ***Opérations*** (1)
- Cliquer sur ***Ajouter*** (2) au niveau d'**Opération**



1. Envoyer aux utilisateurs : ***Sélectionner*** (1) > Cocher ***Admin*** (1) > **Sélectionner** (2)
2. Envoyer au type de média (2): ***GLPI API***
3. Cliquer sur ***Ajouter*** (3)



- Cliquer sur ***Ajouter***

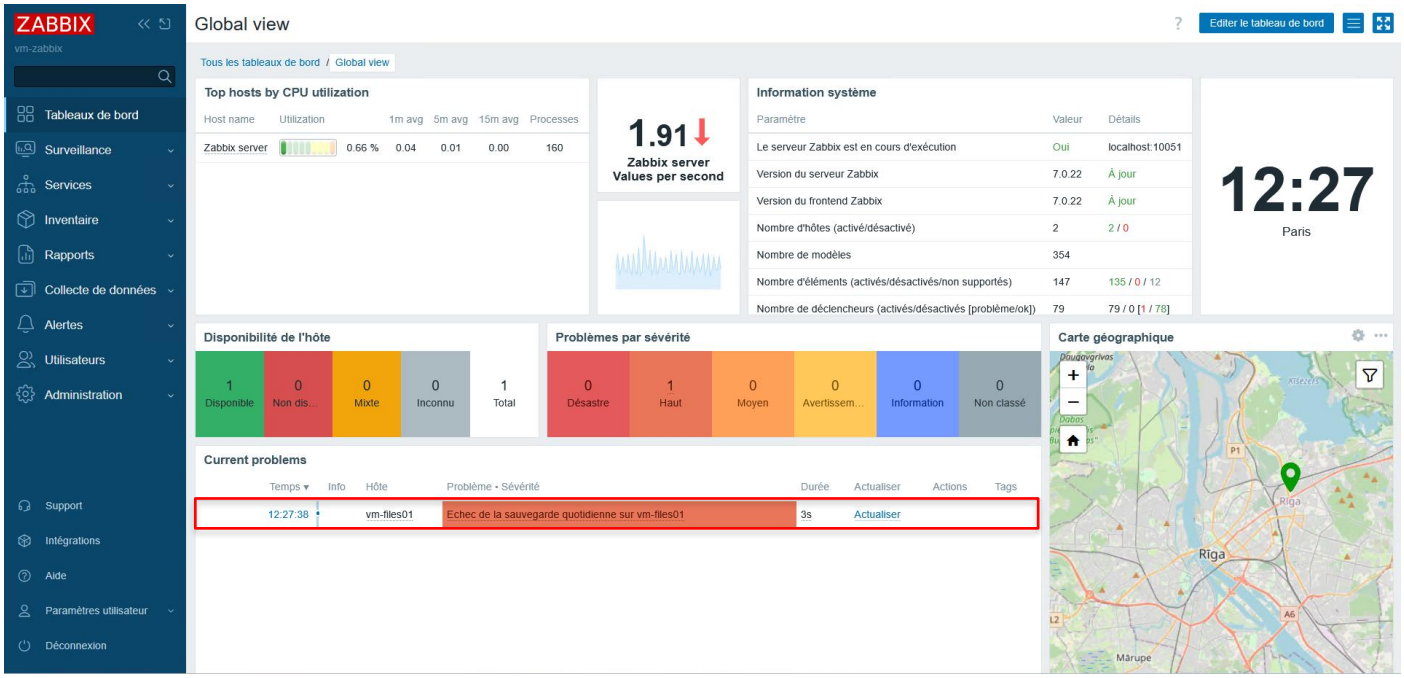


7. Test

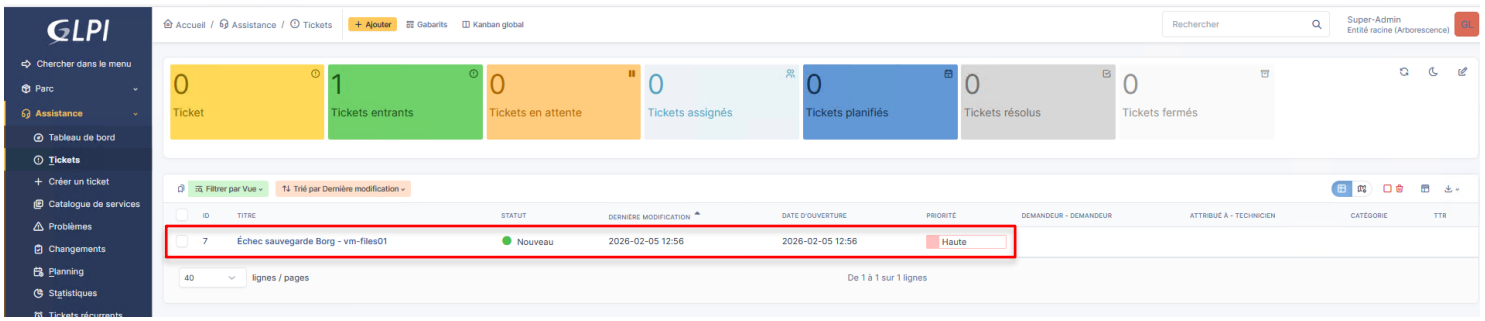
1. Pour s'assurer que les sauvegardes sont fonctionnelles et pour récupérer un fichier :
   - Listez les archives disponibles :
     sudo BORG_PASSPHRASE="PASSPHRASE" borg list /mnt/backup-disk/borg-repo
   - Montez une archive dans un dossier temporaire pour l'explorer (ex: la dernière archive) :
     sudo mkdir /mnt/borg-mount

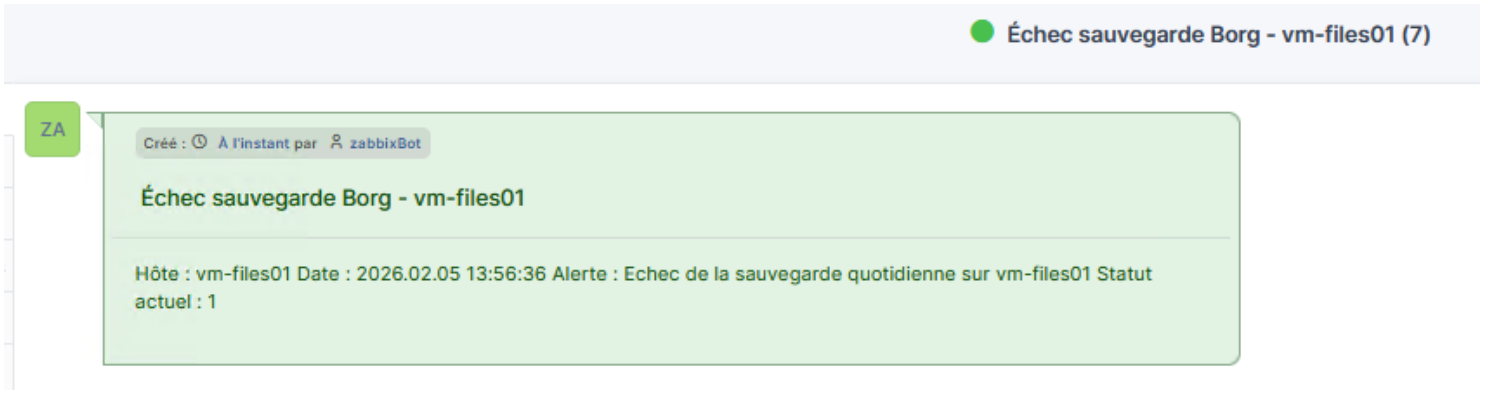
     sudo BORG_PASSPHRASE="PASSPHRASE" borg mount /mnt/backup-disk/borg-repo::NOM_DEPOT /mnt/borg-mount
   - Le partage est accessible dans `/mnt/borg-mount/srv/partage`. Copiez le fichier nécessaire : ***sudo cp /mnt/borg-mount/srv/partage/FICHIER_PERDU.txt /tmp/restored/***
   - Démontez l'archive : ***sudo umount /mnt/borg-mount***

2. Pour s'assurer que l'alerte Zabbix fonctionnelle :
   - Simuler une erreur :
     zabbix_sender -z <IP_ZABBIX> -s "vm-files01" -k "borg.backup.status" -o 0
   - Sur le tableau de bord Zabbix :

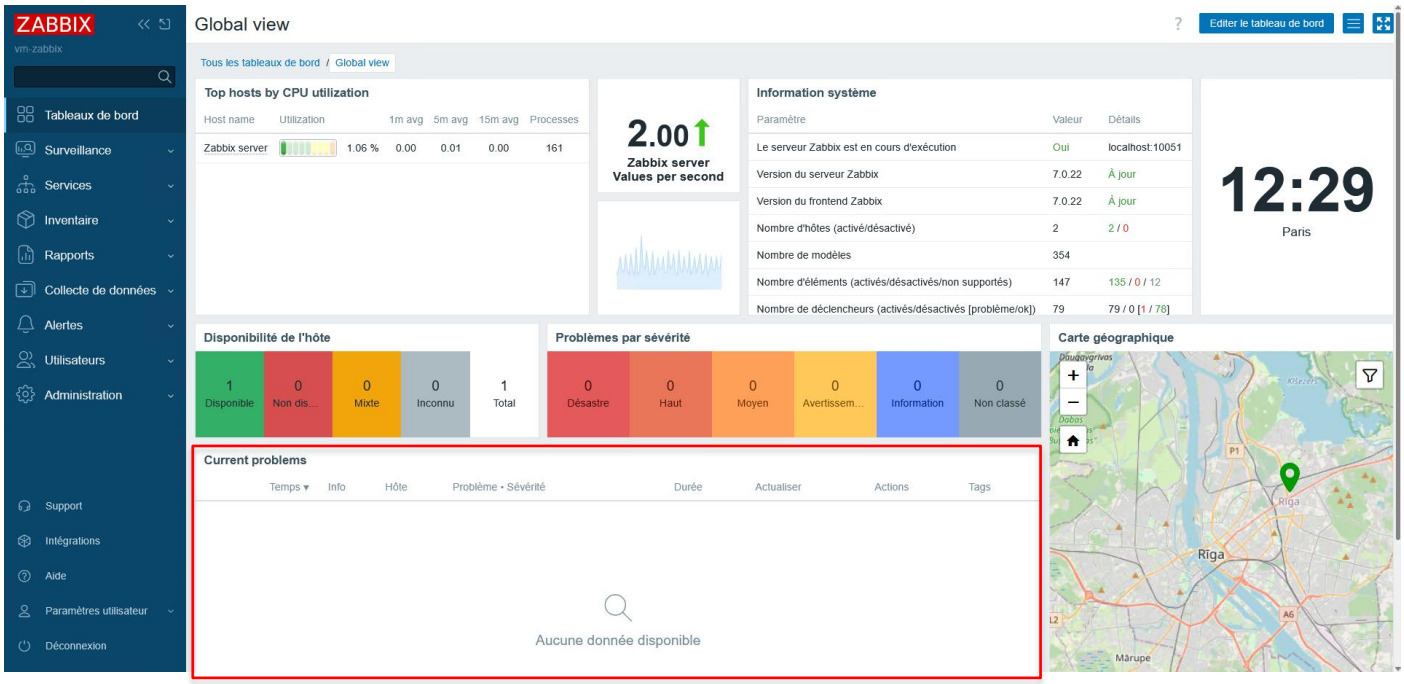


   - Sur les tickets GLPI :





- Revenir à la normale :
  zabbix_sender -z <IP_ZABBIX> -s "vm-files01" -k "borg.backup.status" -o 0
- Sur le tableau de bord Zabbix :



PARTIE 2 : Sauvegarde Hebdomadaire (Externe / Disastre)

L'objectif est de sauvegarder l'image complète de toutes les VMs (y compris `vm-files01`) de l'hôte Proxmox vers un disque USB externe débranché après usage.

1. Préparation du Disque USB (Côté Proxmox)

Cette étape est à réaliser **une seule fois** sur l'hôte Proxmox (`srv-prox01`).

1.1. Identification et Formatage

1. Connectez vous en SSH ou via le shell de l'interface Web au serveur proxmox.
   ssh europacksi@prox
2. Branchez le disque USB, puis identifiez son identifiant stable via `/dev/disk/by-id/` (plus fiable que `/dev/sdX` qui peut changer si d'autres périphériques sont branchés) :
   ls -l /dev/disk/by-id/ | grep -i usb
3. **Exemple de résultat :** `usb-TOSHIBA_EXTERNAL_USB_20221110007491F-0:0` (sans le suffixe `-part1` = disque entier, avec `-part1` = première partition). Notez le chemin complet, par exemple :
   /dev/disk/by-id/usb-TOSHIBA_EXTERNAL_USB_20221110007491F-0:0
4. **ATTENTION :** Formatage en EXT4 du disque USB. Remplacez le chemin by-id par le vôtre (vérifié à l'étape précédente) :

Remplacez le chemin by-id par celui de votre disque USB
sudo mkfs.ext4 -F /dev/disk/by-id/usb-TOSHIBA_EXTERNAL_USB_20221110007491F-0:0
Laissez 262144 blocks en appuyant sur Entrée

```
europacksi@ssi:~$ sudo mkfs.ext4 -F /dev/disk/by-id/usb-TOSHIBA_EXTERNAL_USB_20221110007491F-0:0
[sudo] password for europacksi:
mke2fs 1.47.2 (1-Jan-2025)
Found a dos partition table in /dev/disk/by-id/usb-TOSHIBA_EXTERNAL_USB_20221110007491F-0:0
Creating filesystem with 244190646 4k blocks and 61054976 inodes
Filesystem UUID: 35bdbbcc-4e78-4be9-833f-fec318a91172
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
        102400000, 214990848

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks):
done
Writing superblocks and filesystem accounting information: done
```

2. Le Script de Sauvegarde "Semi-Automatique" (Côté Proxmox)

Ce script gère le montage, la sauvegarde complète des VMs avec `vzdump`, la rétention et le démontage sécurisé.

2.1. Création du Script

Créez le fichier `/root/usb-backup.sh` sur l'hôte Proxmox : ***sudo nano /root/usb-backup.sh***

Script `/root/usb-backup.sh` :

```
#!/bin/bash

# Configuration du Disque USB
USB_DEVICE="/dev/disk/by-id/usb-VOTRE_ID_DISQUE_USB"
MOUNT_POINT="/mnt/usb-backup"
BACKUP_DIR="$MOUNT_POINT/dump"
LOG_FILE="/var/log/usb-backup.log"
DATE=$(date +%Y-%m-%d_%H-%M-%S)
KEEP_VERSIONS=2 # Garder les 2 dernières sauvegardes complètes

echo "--- Démarrage de la sauvegarde USB : $DATE ---" >> $LOG_FILE

# 1. Création du point de montage et vérification
mkdir -p $MOUNT_POINT

if ! mount $USB_DEVICE $MOUNT_POINT; then
    echo "ERREUR : Impossible de monter le disque USB ($USB_DEVICE). Vérifiez le branchement et la partition." | tee -a $LOG_FILE
    exit 1
fi
echo "Disque USB monté avec succès." >> $LOG_FILE
mkdir -p $BACKUP_DIR

# 2. Exécution de VZDUMP pour toutes les VMs
# --dumpdir : chemin direct vers un dossier
# --mode snapshot : rapide, mais nécessite de l'espace sur le stockage source.
# --compress zstd : excellente compression/vitesse pour les backups locaux/USB.
# --maxfiles : conserve uniquement les N dernières sauvegardes par VM.
echo "Démarrage des sauvegardes VZDUMP..." >> $LOG_FILE
vzdump --all 1 --mode snapshot --compress zstd --dumpdir $BACKUP_DIR --maxfiles $KEEP_VERSIONS --quiet 1 2>&1 | tee -a $LOG_FILE


# 3. Démontage sécurisé
echo "Démontage du disque USB..." >> $LOG_FILE
if umount $MOUNT_POINT; then
    echo "SUCCESS : Le disque USB a été démonté. Sauvegarde terminée, vous pouvez débrancher le disque." | tee -a $LOG_FILE
else
    echo "ATTENTION : Le démontage a échoué. Assurez-vous qu'aucun processus n'utilise $MOUNT_POINT, puis débranchez." | tee -a $LOG_FILE
fi
echo "--- Fin de la sauvegarde USB ---" >> $LOG_FILE
```

Rendez le script exécutable : ***sudo chmod +x /root/usb-backup.sh***

2.2. Procédure d'Exécution

La procédure hebdomadaire pour l'administrateur est la suivante :

1. Brancher le disque USB au serveur `srv-prox01`.
2. Se connecter en SSH et lancer le script manuellement : /root/usb-backup.sh
3. Attendre le message de confirmation **"SUCCESS : Le disque USB a été démonté. Sauvegarde terminée, vous pouvez débrancher le disque."**
4. Débrancher le disque USB et le stocker hors ligne/hors site.