

Projet E6 – Supervision réseau avec Zabbix 7.4

Informations du projet

Étudiant : Matthis FINOCCHI

Formation : BTS SIO option SISR

Établissement : UTEC Avon-Fontainebleau / Melun

Année : 2025–2026

Navigation du dossier technique

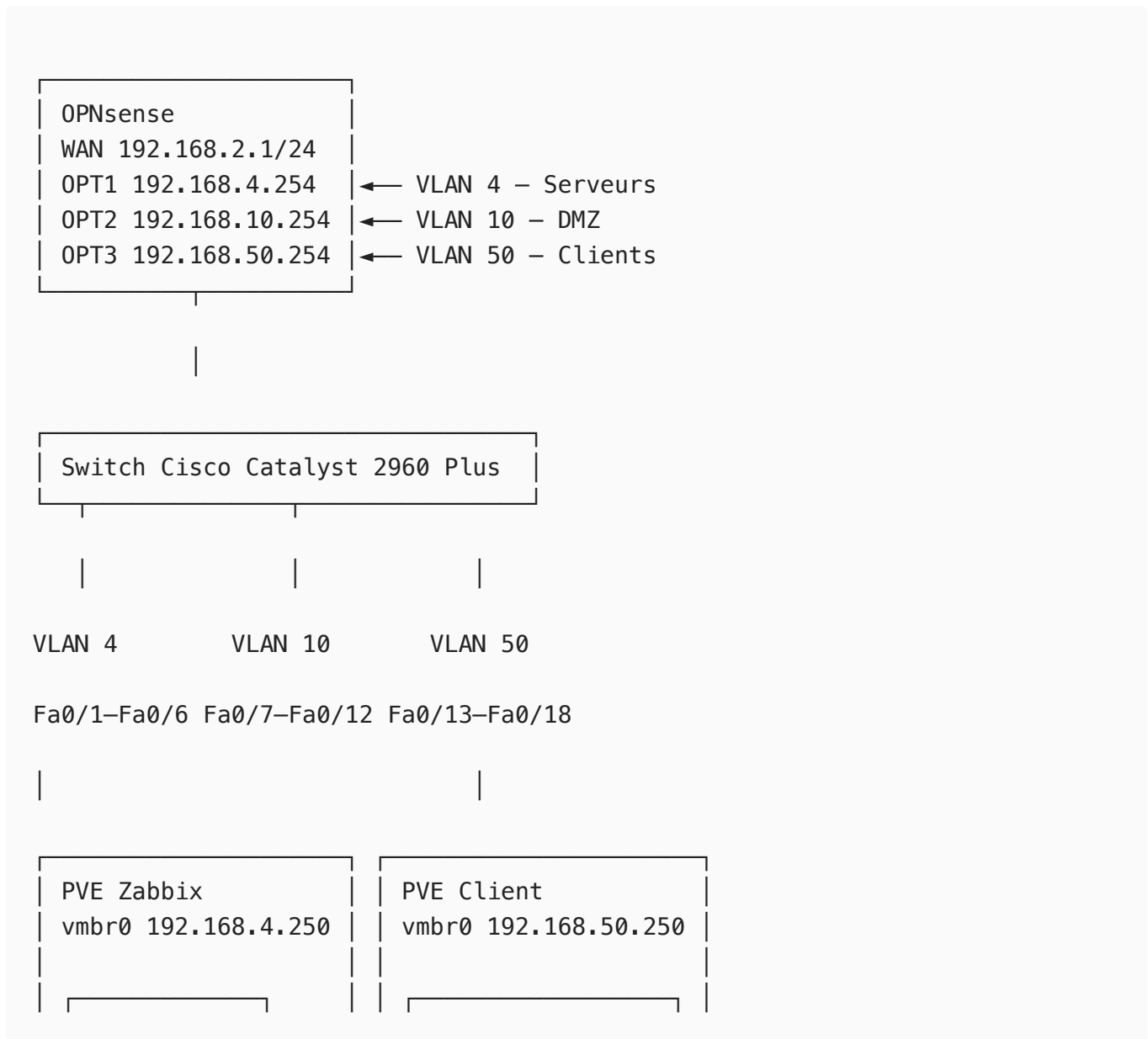
#	Note	Description
01	01 - Architecture	Schéma réseau, tableau des éléments
02	Z02 - Installation Zabbix Server	Installation Ubuntu Server + MySQL + Zabbix 7.4
03	Z03 - Interface web	Configuration de l'interface web Zabbix
04	Z04 - Alertes Gmail	Mot de passe applicatif, type de média, action
05	Z05 - Agent macOS	Installation et configuration sur MacBook
06	Z06 - Ajout MacBook dans Zabbix	Déclaration de l'hôte dans l'interface
07	Z07 - VLAN Aware Proxmox	Activation VLAN Aware sur vmbr0
08	Z08 - VM Cliente VLAN 50	Création VM, IP statique NetworkManager
09	Z09 - Règles OPNsense	Règles inter-VLAN OPT1 et OPT3
10	Z10 - Agent VM Cliente	Installation et config agent VLAN 50
11	Z11 - Ajout VM dans Zabbix	Déclaration de la VM cliente
12	Z12 - Tests et vérifications	Tableau de tests complet
13	Z13 - Dépannage	Problèmes courants et solutions

Avancement

- ✓ Architecture définie
- ✓ Zabbix Server installé et opérationnel
- ✓ Interface web configurée
- ✓ Alertes Gmail fonctionnelles
- ✓ Agent macOS installé
- ✓ VM cliente VLAN 50 créée et supervisée
- ✓ Règles OPNsense inter-VLAN configurées
- ✓ Tests validés

Z01 – Architecture de l'infrastructure

Schéma réseau



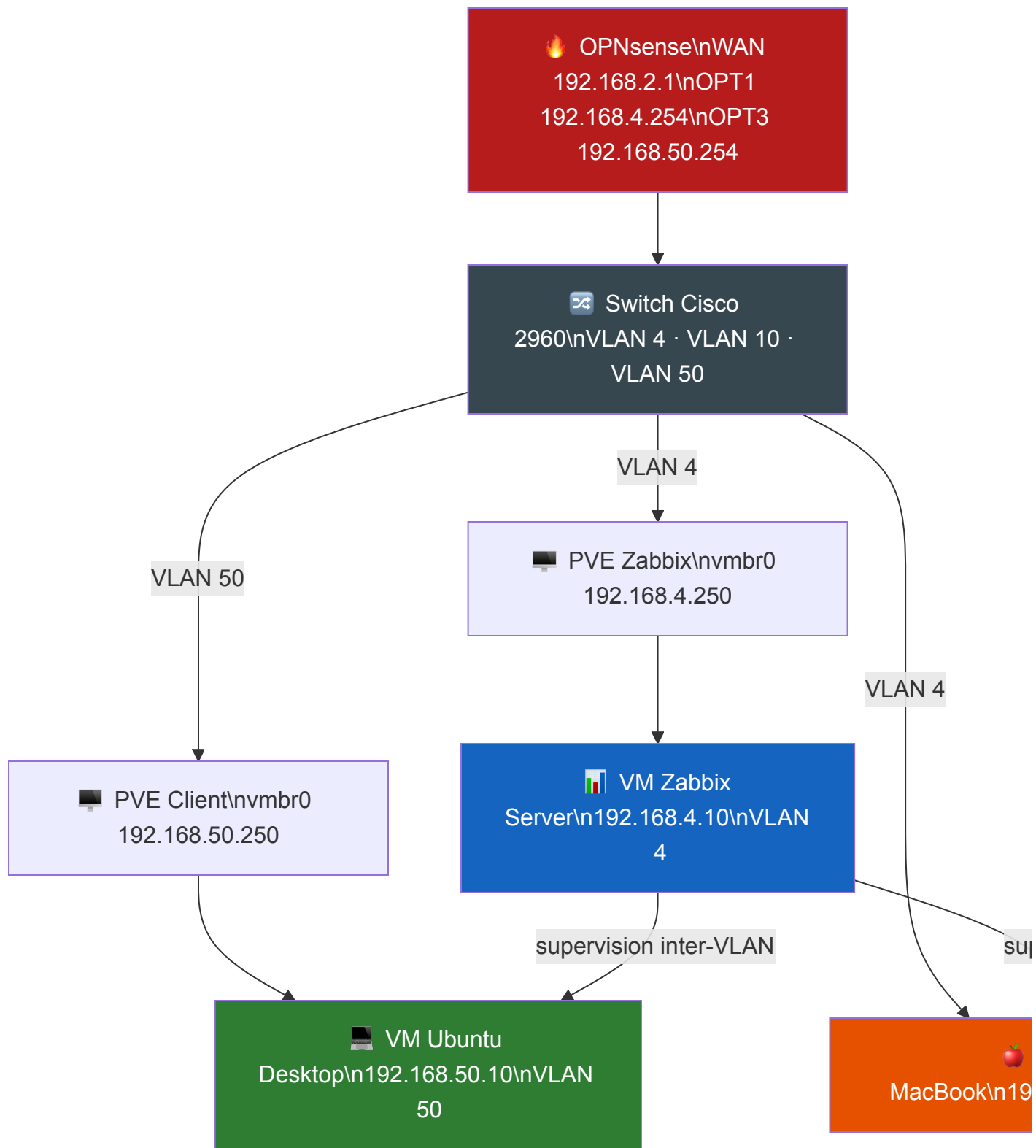
VM Zabbix 668

192.168.4.10

VM Ubuntu Desktop

192.168.50.10

Diagramme Mermaid



📄 Tableau des éléments

Élément	VLAN	Adresse IP	Rôle
VM Zabbix Server	4	192.168.4.10	Supervision centrale

Élément	VLAN	Adresse IP	Rôle
OPNsense OPT1	4	192.168.4.254	Gateway VLAN 4
OPNsense OPT3	50	192.168.50.254	Gateway VLAN 50
VM Ubuntu Desktop Cliente	50	192.168.50.10	Hôte supervisé
MacBook	—	192.168.4.20 (examen)	Hôte supervisé

Plan d'adressage réseau

VLAN	Nom	Réseau	Gateway	Usage
4	Serveurs	192.168.4.0/24	192.168.4.254	VM Zabbix, MacBook
10	DMZ	192.168.10.0/24	192.168.10.254	Services exposés
50	Clients	192.168.50.0/24	192.168.50.254	VM Ubuntu Desktop

Ports du switch Cisco 2960

- **Fa0/1 – Fa0/6** → VLAN 4 (Serveurs)
- **Fa0/7 – Fa0/12** → VLAN 10 (DMZ)
- **Fa0/13 – Fa0/18** → VLAN 50 (Clients)

Z02 – Installation du serveur Zabbix

Machine concernée

Toutes les commandes s'exécutent sur la **VM Ubuntu Server 24.04** (IP : 192.168.4.10), accessible via la console Proxmox.

Étape 1 – Configuration réseau (Netplan)

L'IP statique est indispensable : les agents ont besoin de savoir à quelle IP envoyer leurs données, et l'interface web doit rester accessible à la même adresse

(<http://192.168.4.10/zabbix>).

```
sudo nano /etc/netplan/00-installer-config.yaml
```

Contenu du fichier :

```
network:
version: 2
ethernets:
ens18:
dhcp4: false
addresses:
- 192.168.4.10/24
routes:
- to: default
via: 192.168.4.254
nameservers:
addresses: [8.8.8.8, 1.1.1.1]
```

Paramètre	Valeur	Rôle
-----------	--------	------

| dhcp4: false | — | Désactive le DHCP |

| addresses | 192.168.4.10/24 | IP fixe de la VM |

| via | 192.168.4.254 | Passerelle OPNsense |

| nameservers | 8.8.8.8, 1.1.1.1 | DNS Google + Cloudflare |

```
sudo chmod 600 /etc/netplan/00-installer-config.yaml
```

```
sudo netplan apply
```

🔥 Pourquoi chmod 600 ?

`chmod 600` sécurise le fichier (lecture/écriture uniquement par root). Sans ça, Netplan affiche un avertissement et peut refuser d'appliquer la configuration.

“Capture d'écran 2026-03-12 à 18.11.37.png” ne peut être trouvé.

📦 Étape 2 – Ajout du dépôt Zabbix 7.4

Ubuntu ne connaît pas Zabbix 7.4 dans ses dépôts officiels — il faut ajouter la source manuellement.

```
wget https://repo.zabbix.com/zabbix/7.4/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_7.4+ubuntu24.04_all.deb
```

```
sudo dpkg -i zabbix-release_latest_7.4+ubuntu24.04_all.deb
```

```
sudo apt update
```

| Commande | Rôle |

|---|---|

| `wget` | Télécharge le paquet `.deb` du dépôt Zabbix |

| `dpkg -i` | Enregistre le dépôt Zabbix sur le système |

| `apt update` | Recharge la liste des paquets pour voir Zabbix 7.4 |

📦 Étape 3 – Installation des paquets

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

| Paquet | Rôle |

|---|---|

| zabbix-server-mysql | Serveur Zabbix qui collecte les données |

| zabbix-frontend-php | Interface web accessible depuis le navigateur |

| zabbix-apache-conf | Configuration Apache pour servir l'interface web |

| zabbix-sql-scripts | Scripts SQL pour initialiser la base de données |

| zabbix-agent | Agent local pour superviser la VM Zabbix elle-même |

Étape 4 – Configuration MySQL

Zabbix a besoin d'une base de données dédiée. On crée un utilisateur spécifique plutôt que d'utiliser root par sécurité.

```
sudo apt install mysql-server -y
```

```
sudo mysql
```

```
CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
```

```
CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
```

```
SET GLOBAL log_bin_trust_function_creators = 1;
```

```
EXIT;
```

| Commande SQL | Rôle |

|---|---|

| CREATE DATABASE | Crée la base `zabbix` en UTF8 (obligatoire pour Zabbix) |

| CREATE USER | Crée un utilisateur dédié avec mot de passe |

| GRANT ALL PRIVILEGES | Donne tous les droits sur cette base uniquement |

| `log_bin_trust_function_creators = 1` | Requis temporairement pour l'import du schéma |

Étape 5 – Import du schéma de base de données

```
zcat /usr/share/zabbix/sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -uzabbix -p zabbix
```

Durée d'exécution

Cette commande prend **plusieurs minutes** — c'est normal. Zabbix initialise des milliers de tables, index et données de configuration.

`zcat` décompresse le fichier SQL fourni par Zabbix et le passe directement à MySQL via le pipe `|`.

Étape 6 – Configuration du mot de passe DB

Le serveur Zabbix doit connaître le mot de passe MySQL pour se connecter.

```
sudo nano /etc/zabbix/zabbix_server.conf
```

Modifier la ligne suivante :

```
DBPassword=password
```

“Capture d'écran 2026-03-12 à 18.10.59.png” ne peut être trouvé.

Étape 7 – Démarrage des services

```
sudo systemctl restart zabbix-server zabbix-agent apache2
```

```
sudo systemctl enable zabbix-server zabbix-agent apache2
```

| Option | Rôle |

|---|---|

| restart | Démarre les services en prenant en compte les configurations |

| enable | Démarre automatiquement les services à chaque reboot |

Interdépendance des services

Les trois services sont liés : **Apache** sert l'interface web, **zabbix-server** collecte les métriques, **zabbix-agent** supervise la VM locale.

Étape 8 – Vérification

```
sudo systemctl status zabbix-server
```

Résultat attendu

Le statut doit afficher **active (running)** en vert. En cas d'erreur, les logs s'affichent directement pour diagnostiquer.

“Capture d'écran 2026-03-12 à 18.12.34.png” ne peut être trouvé.

Navigation

← Retour : [01 - Architecture](#)

→ Suite : [Z03 - Interface web](#)

tags:

- BTS-SIO
- SISR
- E6
- Zabbix
- Interface
- Web

parents: "[00 - Index du projet Zabbix](#)"

Z03 – Configuration de l'interface web

Accès à l'interface

Depuis un navigateur, ouvrir :

```
http://192.168.4.10/zabbix
```

Assistant d'installation

Suivre l'assistant en complétant les champs suivants :

| Étape | Champ | Valeur |

|---|---|---|

| Connexion DB | Type de base de données | MySQL |

| Connexion DB | Hôte | localhost |

| Connexion DB | Utilisateur | zabbix |

| Connexion DB | Mot de passe | password |

| Connexion DB | Nom de la base | zabbix |

Fuseau horaire

Sélectionner **Europe/Paris** à l'étape de configuration du fuseau horaire.

“Capture d'écran 2026-03-12 à 18.13.36.png” ne peut être trouvé.

Identifiants par défaut

 **Changer le mot de passe après la première connexion !**

| Champ | Valeur |

|---|---|

| Utilisateur | Admin |

| Mot de passe | zabbix |

Vérification

Après connexion, le tableau de bord Zabbix s'affiche. Vérifier que :

- Le serveur Zabbix apparaît en **vert** dans **Collecte de données** → **Hôtes**
- Aucune erreur n'est visible dans **Rapports** → **Journal système**



Navigation

← Retour : [Z02 - Installation Zabbix Server](#)

→ Suite : [Z04 - Alertes Gmail](#)

tags:

- BTS-SIO
- SISR
- E6
- Zabbix
- Gmail
- Alertes

parents: "[00 - Index du projet Zabbix](#)"

Z04 – Configuration des alertes Gmail

Vue d'ensemble du flux d'alerte



1 Créer un mot de passe d'application Google

Prérequis

La validation en deux étapes doit être activée sur le compte Google pour accéder aux mots de passe d'application.

1. Aller sur myaccount.google.com
2. **Sécurité** → **Validation en 2 étapes** → Activer

3. **Sécurité** → **Mots de passe des applications** → Créer
4. Nom de l'application : Zabbix
5. Copier le mot de passe de **16 caractères** généré (sans espaces)

Tip

Ce mot de passe n'est affiché qu'une seule fois — le noter immédiatement dans un endroit sûr.

2 Configurer le type de média dans Zabbix

Chemin : Alertes → Types de média → Gmail

“Capture d'écran 2026-03-12 à 18.16.15.png” ne peut être trouvé.

| Champ | Valeur |

|---|---|

| Fournisseur | Gmail |

| Courriel expéditeur | alertezabbixmatthis@gmail.com |

| Mot de passe | (*mot de passe d'application 16 caractères*) |

| Activé | ✓ |

3 Ajouter le média à l'utilisateur Admin

Chemin : Utilisateurs → Admin → Médias → Ajouter

| Champ | Valeur |

|---|---|

| Type | Gmail |

| Envoyer à | alertezabbixmatthis@gmail.com |

| Activé | ✓ |

“Capture d'écran 2026-03-12 à 18.17.38.png” ne peut être trouvé.

4 Activer l'action d'alerte

Chemin : Alertes → Actions → Actions de déclencheur

Activer l'action : **Report problems to Zabbix administrators**

“Capture d'écran 2026-03-12 à 18.19.01.png” ne peut être trouvé.

“Capture d'écran 2026-03-12 à 18.19.38.png” ne peut être trouvé.

✓ Test de l'alerte

Pour tester que tout fonctionne :

1. Éteindre la VM cliente (`sudo shutdown now` sur la VM Ubuntu Desktop)
2. Attendre environ **5 minutes** (délai du déclencheur par défaut)
3. Vérifier la réception d'un email sur `alertezabbixmatthis@gmail.com`

✓ Résultat attendu

Un email avec le sujet `PROBLEM: ...` doit arriver, suivi d'un email `RESOLVED: ...` quand la VM redémarre.

Navigation

← Retour : [Z03 - Interface web](#)

→ Suite : [Z05 - Agent macOS](#)

tags:

- BTS-SIO
- SISR

- E6
- Zabbix
- macOS
- Agent
- Homebrew

parents: "[00 - Index du projet Zabbix](#)"

Z05 – Installation de l'agent sur macOS

Machine concernée

Toutes les commandes s'exécutent sur le **MacBook**, dans le terminal macOS.

Prérequis réseau pour l'examen

Pour que la supervision fonctionne, le Mac doit être **branché physiquement sur le switch** (VLAN 4) le jour de l'examen, avec un adaptateur **USB-C → RJ45**, et configuré avec l'IP statique `192.168.4.20`.

Étape 1 – Installation de Homebrew

Si Homebrew n'est pas encore installé :

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Apple Silicon

Sur MacBook Apple Silicon (M1/M2/M3), Homebrew s'installe dans `/opt/homebrew/` (au lieu de `/usr/local/` sur Intel).

Étape 2 – Installation de l'agent Zabbix

```
brew install zabbix
```

Homebrew télécharge et installe automatiquement `zabbix_agentd` et ses dépendances.

Étape 3 – Configuration de l'agent

```
nano /opt/homebrew/etc/zabbix/zabbix_agentd.conf
```

Modifier ces trois lignes :

```
Server=192.168.4.10,127.0.0.1
```

```
ServerActive=192.168.4.10
```

```
Hostname=MacBook
```

| Paramètre | Valeur | Rôle |

|---|---|---|

| `Server` | 192.168.4.10,127.0.0.1 | IP(s) autorisées à interroger l'agent |

| `ServerActive` | 192.168.4.10 | IP vers laquelle l'agent envoie des données en mode actif |

| `Hostname` | MacBook | Doit correspondre exactement au nom d'hôte dans Zabbix |

Correspondance du Hostname

La valeur de `Hostname=` dans ce fichier doit être **identique** au champ "Nom d'hôte" saisi dans l'interface Zabbix ([Z06 - Ajout MacBook dans Zabbix](#)).

Étape 4 – Démarrage de l'agent

```
sudo /opt/homebrew/sbin/zabbix_agentd -c
/opt/homebrew/etc/zabbix/zabbix_agentd.conf
```

Étape 5 – Démarrage automatique au boot (LaunchDaemon)

Pour que l'agent redémarre automatiquement après chaque reboot du Mac :

```
sudo nano /Library/LaunchDaemons/com.zabbix.agentd.plist
```

Contenu du fichier :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Label</key>
<string>com.zabbix.agentd</string>
<key>ProgramArguments</key>
<array>
<string>/opt/homebrew/sbin/zabbix_agentd</string>
<string>-c</string>
<string>/opt/homebrew/etc/zabbix/zabbix_agentd.conf</string>
```

```
</array>

<key>RunAtLoad</key>

<true/>

<key>KeepAlive</key>

<true/>

</dict>

</plist>
```

```
sudo launchctl load /Library/LaunchDaemons/com.zabbix.agentd.plist
```

| Clé plist | Rôle |

|---|---|

| RunAtLoad | Démarre l'agent au chargement du daemon |

| KeepAlive | Redémarre l'agent automatiquement s'il s'arrête |

✓ Étape 6 – Vérification

```
ps aux | grep zabbix
```

✓ **Résultat attendu**

| Une ligne contenant `zabbix_agentd` doit apparaître, confirmant que le processus tourne.

 **Navigation**

← Retour : [Z04 - Alertes Gmail](#)

→ Suite : [Z06 - Ajout MacBook dans Zabbix](#)

tags:

- BTS-SIO
- SISR
- E6
- Zabbix
- macOS
- Hôte

parents: "[00 - Index du projet Zabbix](#)"

Z06 – Ajout du MacBook dans Zabbix

Chemin dans l'interface

Collecte de données → Hôtes → Créer un hôte

Paramètres de l'hôte

| Champ | Valeur |

|---|---|

| Nom d'hôte | MacBook |

| Groupe | Linux servers |

| Interface Agent – IP | 192.168.4.20 |

| Interface Agent – Port | 10050 |

| Template | Linux by Zabbix agent |

 **Jour de l'examen**

Brancher le Mac sur un port **VLAN 4** (Fa0/1 – Fa0/6) du switch Cisco et lui attribuer l'IP **192.168.4.20** en statique avant de déclarer l'hôte.

Correspondance Hostname

Le champ **Nom d'hôte** doit correspondre exactement à `Hostname=MacBook` dans le fichier de config de l'agent ([Z05 - Agent macOS](#)).

Vérification

Après création de l'hôte, attendre ~1 minute puis vérifier dans **Collecte de données** → **Hôtes** :

| Indicateur | Statut attendu |

|---|---|

| Icône ZBX |  Verte |

| Icône JMX | Grise (normal, non configuré) |

| Icône SNMP | Grise (normal, non configuré) |

Résultat attendu

L'icône **ZBX verte** confirme que le serveur Zabbix communique avec l'agent sur le MacBook.

Navigation

← Retour : [Z05 - Agent macOS](#)

→ Suite : [Z07 - VLAN Aware Proxmox](#)

tags:

- BTS-SIO
- SISR
- E6

- Zabbix
- Proxmox
- VLAN

parents: "[00 - Index du projet Zabbix](#)"

Z07 – Activation VLAN Aware sur Proxmox

Machine concernée

Cette opération est à réaliser sur le **PVE Client** — le nœud Proxmox qui hébergera la VM Ubuntu Desktop sur le VLAN 50.

Pourquoi activer VLAN Aware ?

Par défaut, un bridge Proxmox (`vmbr0`) transmet le trafic réseau **sans tenir compte des tags VLAN**. Pour qu'une VM puisse être placée sur un VLAN spécifique (ici VLAN 50), il faut activer l'option "**Gère les VLAN**" sur le bridge, qui lui permet de comprendre et transmettre les tags 802.1Q.

Procédure

Dans l'interface Proxmox

Chemin : nœud pve → Système → Réseau → `vmbr0` → Éditer

Vérifier l'état de la case "**Gère les VLAN**" :

| État | Action |

|---|---|

| Cochée | Rien à faire |

| Décochée | Cocher → OK → **Appliquer la configuration** |

“Capture d'écran 2026-03-12 à 18.23.59.png” ne peut être trouvé.

✓ Vérification

Après application, la configuration réseau de Proxmox doit afficher :

```
bridge-vlan-aware yes
```

dans les propriétés de `vmb0`. Cette ligne confirme que le bridge accepte maintenant les VMs avec des tags VLAN configurés.

⚠ Redémarrage possible

Sur certaines versions de Proxmox, un redémarrage du nœud peut être nécessaire pour appliquer le changement.

🔗 Navigation

← Retour : [Z06 - Ajout MacBook dans Zabbix](#)

→ Suite : [Z08 - VM Cliente VLAN 50](#)

tags:

- BTS-SIO
- SISR
- E6
- Zabbix
- Proxmox
- VLAN
- Ubuntu
- NetworkManager

parents: "[00 - Index du projet Zabbix](#)"

Z08 – Création de la VM cliente VLAN 50

Étape 1 – Création dans Proxmox

Chemin : Créer VM

| Paramètre | Valeur |

|---|---|

| Nom | Ubuntu-Client-VLAN50 |

| ISO | Ubuntu Desktop 24.04 LTS |

| RAM | 4 Go minimum |

| Disque | 20 Go minimum |

| Bridge | vmbr0 |

| **VLAN Tag** | **50** |

VLAN Tag dans Proxmox

Le champ **VLAN Tag** dans les paramètres réseau de la VM est ce qui place la VM sur le VLAN 50. Sans ce tag, la VM serait sur le réseau par défaut sans segmentation.

“Capture d’écran 2026-03-12 à 18.26.59.png” ne peut être trouvé.

Étape 2 – Installation Ubuntu Desktop

1. Démarrer la VM et suivre l'installateur Ubuntu
2. À l'étape "**Connexion Internet**" → sélectionner "**Je ne souhaite pas me connecter à internet pour l'instant**"
3. Terminer l'installation et redémarrer

Étape 3 – Configuration IP statique (NetworkManager)

❗ Pourquoi NetworkManager et pas Netplan ?

Contrairement à Ubuntu Server (qui utilise Netplan avec des fichiers YAML), Ubuntu Desktop utilise **NetworkManager** pour gérer les connexions réseau dans les environnements graphiques. Il faut donc passer par `nmcli` et non éditer un fichier Netplan.

L'objectif est le même qu'à l'étape de la VM Zabbix : donner une IP fixe (`192.168.50.10`) pour que le serveur Zabbix puisse toujours joindre cette machine.

```
nmcli con mod "Wired connection 1" ipv4.addresses 192.168.50.10/24
nmcli con mod "Wired connection 1" ipv4.gateway 192.168.50.254
nmcli con mod "Wired connection 1" ipv4.dns "8.8.8.8 1.1.1.1"
nmcli con mod "Wired connection 1" ipv4.method manual
nmcli con up "Wired connection 1"
```

| Paramètre | Valeur | Rôle |

|---|---|---|

| `ipv4.addresses` | `192.168.50.10/24` | IP statique de la VM sur VLAN 50 |

| `ipv4.gateway` | `192.168.50.254` | OPNsense comme passerelle inter-VLAN |

| `ipv4.dns` | `8.8.8.8 1.1.1.1` | DNS Google + Cloudflare |

| `ipv4.method manual` | — | Désactive le DHCP |

| `nmcli con up` | — | Applique immédiatement sans redémarrer |

⚠ Orthographe

Attention à bien écrire `ipv4.addresses` avec deux `s`. Une faute de frappe ici retourne une erreur "propriété non valide".

✅ Étape 4 – Vérification de la connectivité

```
ping 192.168.50.254 # Gateway OPNsense VLAN 50
```

```
ping 192.168.4.10 # Serveur Zabbix (inter-VLAN)
```

✓ Résultat attendu

Les deux pings doivent répondre. Si `192.168.4.10` ne répond pas, vérifier les [règles OPNsense](#).

“Capture d'écran 2026-03-12 à 18.42.48.png” ne peut être trouvé.

Navigation

← Retour : [Z07 - VLAN Aware Proxmox](#)

→ Suite : [Z09 - Règles OPNsense](#)

tags:

- BTS-SIO
- SISR
- E6
- Zabbix
- OPNsense
- Firewall
- VLAN

parents: "[00 - Index du projet Zabbix](#)"

Z09 – Configuration OPNsense — Règles inter-VLAN

 **Comportement par défaut d'OPNsense**

OPNsense **bloque par défaut tout le trafic inter-VLAN**. Sans règles explicites, la VM VLAN 50 ne peut pas joindre le serveur Zabbix en VLAN 4, et inversement.

🎯 Objectif

Autoriser le trafic Zabbix entre les deux VLANs dans les deux sens :



📄 Règle 1 – OPT1 (VLAN 4) — Serveur vers clients

Chemin : Firewall → Règles → OPT1 → Ajouter

| Champ | Valeur |

|---|---|

| Action | Pass |

| Interface | OPT1 |

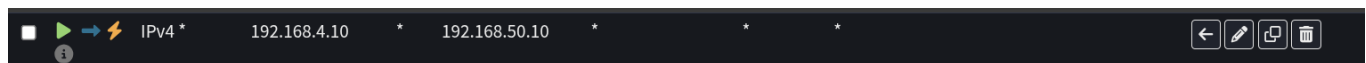
| Protocole | TCP |

| Source | 192.168.4.10 |

| Destination | 192.168.50.10/24 |

| Port destination | any |

| Description | Zabbix serveur vers agents VLAN50 |



📄 Règle 2 – OPT3 (VLAN 50) — Clients vers serveur

Chemin : Firewall → Règles → OPT3 → Ajouter

| Champ | Valeur |

|---|---|

| Action | Pass |

| Interface | OPT3 |

| Protocole | TCP |

| Source | 192.168.50.10/24 |

| Destination | 192.168.4.10 |

| Port destination | any |

| Description | Zabbix agents VLAN50 vers serveur |



Ports Zabbix utilisés

| Port | Protocole | Usage |

|---|---|---|

| **10050** | TCP | Agent passif (serveur interroge l'agent) |

| **10051** | TCP | Agent actif (agent envoie vers le serveur) |

Affiner les règles

Pour une meilleure sécurité, on peut restreindre les règles aux ports **10050** et **10051** uniquement au lieu de `any`. Pour un projet BTS, `any` est suffisant.

Vérification

Depuis la VM cliente (192.168.50.10) :

```
ping 192.168.4.10
```

```
nc -zv 192.168.4.10 10051
```

Depuis le serveur Zabbix (192.168.4.10) :

```
nc -zv 192.168.50.10 10050
```

✓ **Résultat attendu**

Les deux commandes `nc` doivent retourner `succeeded` ou `open` .

Navigation

← Retour : [Z08 - VM Cliente VLAN 50](#)

→ Suite : [Z10 - Agent VM Cliente](#)

tags:

- BTS-SIO
- SISR
- E6
- Zabbix
- Agent
- Linux
- VLAN

parents: "[00 - Index du projet Zabbix](#)"

Z10 – Installation de l'agent sur la VM cliente

Machine concernée

Toutes les commandes s'exécutent sur la **VM Ubuntu Desktop** (IP : 192.168.50.10).

Étape 1 – Ajout du dépôt Zabbix

```
wget https://repo.zabbix.com/zabbix/7.4/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_7.4+ubuntu24.04_all.deb
```

```
sudo dpkg -i zabbix-release_latest_7.4+ubuntu24.04_all.deb
```

```
sudo apt update
```

Tip

Même procédure que sur le serveur Zabbix ([étape 2](#)) — on ajoute le dépôt officiel Zabbix pour obtenir la version 7.4.

Étape 2 – Installation de l'agent

```
sudo apt install zabbix-agent -y
```

Ici, on installe uniquement `zabbix-agent` — pas le serveur ni l'interface web.

Étape 3 – Configuration de l'agent

```
sudo nano /etc/zabbix/zabbix_agentd.conf
```

Modifier ces lignes :

```
Server=192.168.4.10,192.168.50.10
```

```
ServerActive=192.168.4.10
```

```
Hostname=pc1-Standard-PC-i440FX-PIIX-1996
```

| Paramètre | Valeur | Rôle |

|---|---|---|

| Server | 192.168.4.10,192.168.50.10 | IPs autorisées à interroger l'agent |

| ServerActive | 192.168.4.10 | IP du serveur Zabbix pour le mode actif |

| Hostname | pc1-Standard-PC-i440FX-PIIX-1996 | Doit correspondre exactement au nom dans Zabbix |

Double Hostname

Si le fichier contient `Hostname=Hostname=xxx`, supprimer la ligne en double et ne garder que `Hostname=xxx`.

“Capture d'écran 2026-03-12 à 18.58.34.png” ne peut être trouvé.

“Capture d'écran 2026-03-12 à 18.51.26.png” ne peut être trouvé.

Étape 4 – Démarrage de l'agent

```
sudo systemctl restart zabbix-agent
```

```
sudo systemctl enable zabbix-agent
```

```
sudo systemctl status zabbix-agent
```

✓ **Résultat attendu**

Le statut doit afficher `active (running)` .

✓ Étape 5 – Vérification depuis le serveur Zabbix

Depuis la VM Zabbix Server (192.168.4.10) :

```
sudo apt install zabbix-get -y  
  
zabbix_get -s 192.168.50.10 -p 10050 -k agent.hostname
```

✓ Résultat attendu

pc1-Standard-PC-i440FX-PIIX-1996

Si cette valeur s'affiche, l'agent répond correctement au serveur Zabbix.

“Capture d'écran 2026-03-12 à 18.57.01.png” ne peut être trouvé.

Navigation

← Retour : [Z09 - Règles OPNsense](#)

→ Suite : [Z11 - Ajout VM dans Zabbix](#)

tags:

- BTS-SIO
- SISR
- E6
- Zabbix

- Hôte
- VLAN

parents: "[00 - Index du projet Zabbix](#)"

Z11 – Ajout de la VM cliente dans Zabbix

Chemin dans l'interface

Collecte de données → Hôtes → Créer un hôte

Paramètres de l'hôte

| Champ | Valeur |

|---|---|

| Nom d'hôte | pc1-Standard-PC-i440FX-PIIX-1996 |

| Groupe | Linux servers |

| Interface Agent – IP | 192.168.50.10 |

| Interface Agent – Port | 10050 |

| Template | Linux by Zabbix agent |

Correspondance exacte du Hostname

Le champ **Nom d'hôte** doit être **identique** à la valeur `Hostname=` dans le fichier de configuration de l'agent ([Z10 - Agent VM Cliente](#)). La moindre différence (majuscule, espace, tiret) empêche la supervision.

“Capture d'écran 2026-03-12 à 19.00.04.png” ne peut être trouvé.

Vérification

Après création de l'hôte, attendre ~1 minute puis aller dans **Collecte de données** → **Hôtes** :

| Indicateur | Statut attendu |

|---|---|

| Icône ZBX | ● Verte |

| Disponibilité | Activé |

Récapitulatif des hôtes supervisés

| Hôte | IP | Template | Statut |

|---|---|---|---|

| Zabbix Server (local) | 127.0.0.1 | Linux by Zabbix agent | ● ZBX vert |

| pc1-Standard-PC-i440FX-PIIX-1996 | 192.168.50.10 | Linux by Zabbix agent | ● ZBX vert |

| MacBook | 192.168.4.20 (examen) | Linux by Zabbix agent | ⚠ Branchement physique requis

|

Navigation

← Retour : [Z10 - Agent VM Cliente](#)

→ Suite : [Z12 - Tests et vérifications](#)

tags:

- BTS-SIO
- SISR
- E6
- Zabbix
- Tests
- Validation

parents: "[00 - Index du projet Zabbix](#)"


Z12 – Tests et vérifications


Tableau de tests complet


| # | Test | Commande / Action | Résultat attendu | Statut |


|---|---|---|---|---|

| 1 | Ping gateway VLAN 50 | `ping 192.168.50.254` | Réponse TTL |  |

| 2 | Ping serveur Zabbix (inter-VLAN) | `ping 192.168.4.10` | Réponse TTL |  |

| 3 | Port agent accessible | `nc -zv 192.168.50.10 10050` | Connection succeeded |  |

| 4 | Agent répond au serveur | `zabbix_get -s 192.168.50.10 -p 10050 -k agent.hostname` | Hostname retourné |  |

| 5 | Supervision active dans Zabbix | Zabbix → Hôtes → Disponibilité | Icône ZBX verte |  |

| 6 | Alerte email fonctionnelle | Éteindre la VM, attendre 5 min | Email reçu sur Gmail |  |

Détail des tests

Test 3 – Vérification du port 10050

Depuis le serveur Zabbix (192.168.4.10) :

```
nc -zv 192.168.50.10 10050
```

```
Connection to 192.168.50.10 10050 port [tcp/*] succeeded!
```

Test 4 – Interrogation de l'agent

Depuis le serveur Zabbix (192.168.4.10) :

```
zabbix_get -s 192.168.50.10 -p 10050 -k agent.hostname
```

```
pc1-Standard-PC-i440FX-PIIX-1996
```

D'autres clés utiles à tester :

```
zabbix_get -s 192.168.50.10 -p 10050 -k system.uptime
```

```
zabbix_get -s 192.168.50.10 -p 10050 -k vm.memory.size[available]
```

```
zabbix_get -s 192.168.50.10 -p 10050 -k system.cpu.load[all,avg1]
```

Test 6 – Simulation d'une alerte

```
# Sur la VM cliente (192.168.50.10)
```

```
sudo shutdown now
```

Attendre 5 minutes → vérifier la boîte mail `alartezabbixmatthis@gmail.com`.

✓ Emails attendus

1. **PROBLEM** : Zabbix agent on pc1-... is unreachable for 5 minutes
2. **RESOLVED** : (après redémarrage de la VM)

Checklist finale avant présentation E6

- ☑ VM Zabbix Server démarrée et accessible sur <http://192.168.4.10/zabbix>
- ☑ Icône ZBX verte pour pc1-Standard-PC-i440FX-PIIX-1996
- ☑ Alertes Gmail testées et fonctionnelles
- ☑ Règles OPNsense OPT1 et OPT3 présentes
- ☑ VLAN Aware activé sur vmbr0 du PVE Client
- ☑ MacBook prêt (agent installé, config faite)
- ☑ Adaptateur USB-C → RJ45 disponible pour le jour J

Navigation

← Retour : [Z11 - Ajout VM dans Zabbix](#)

→ Suite : [Z13 - Dépannage](#)

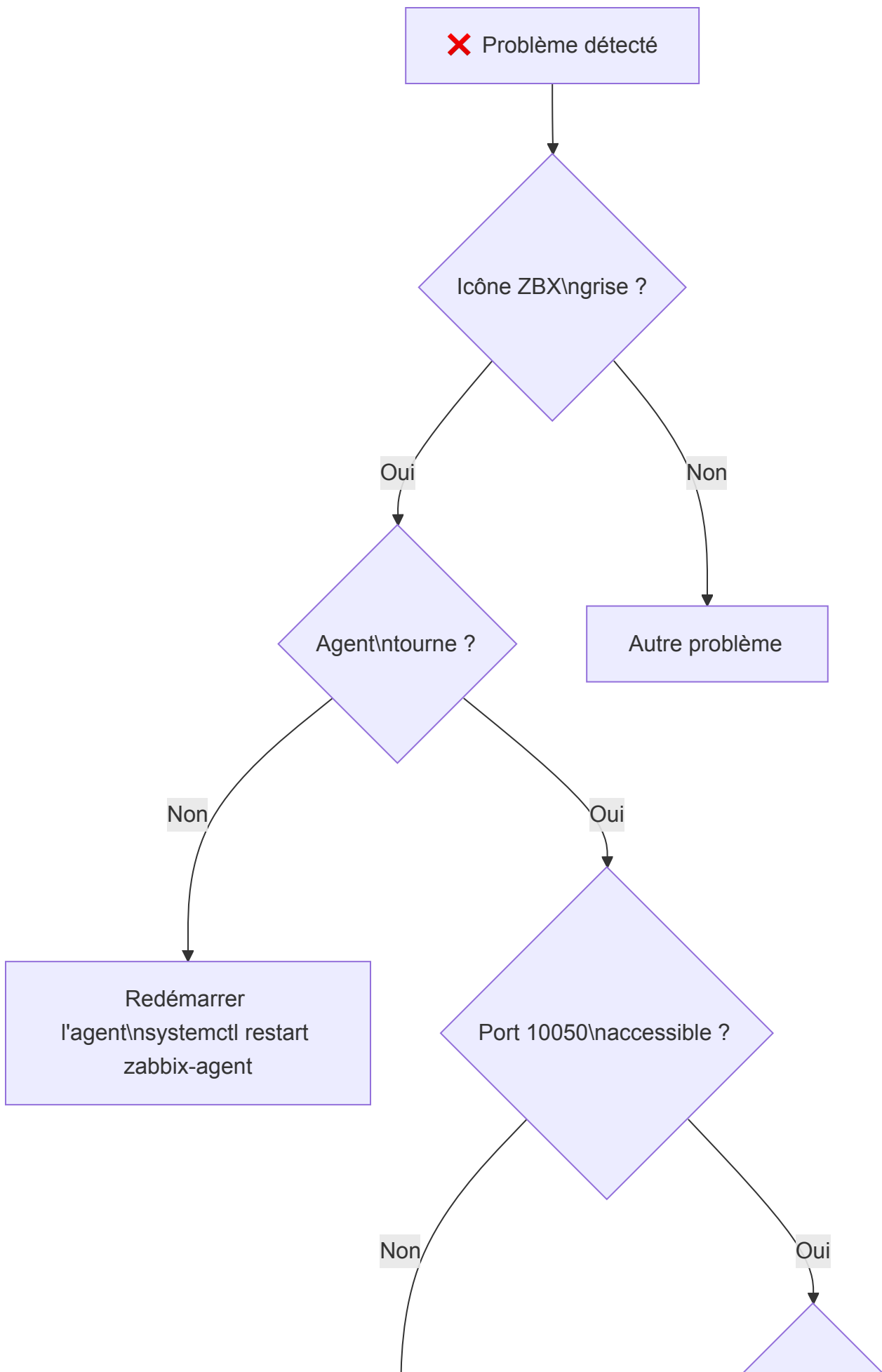
tags:

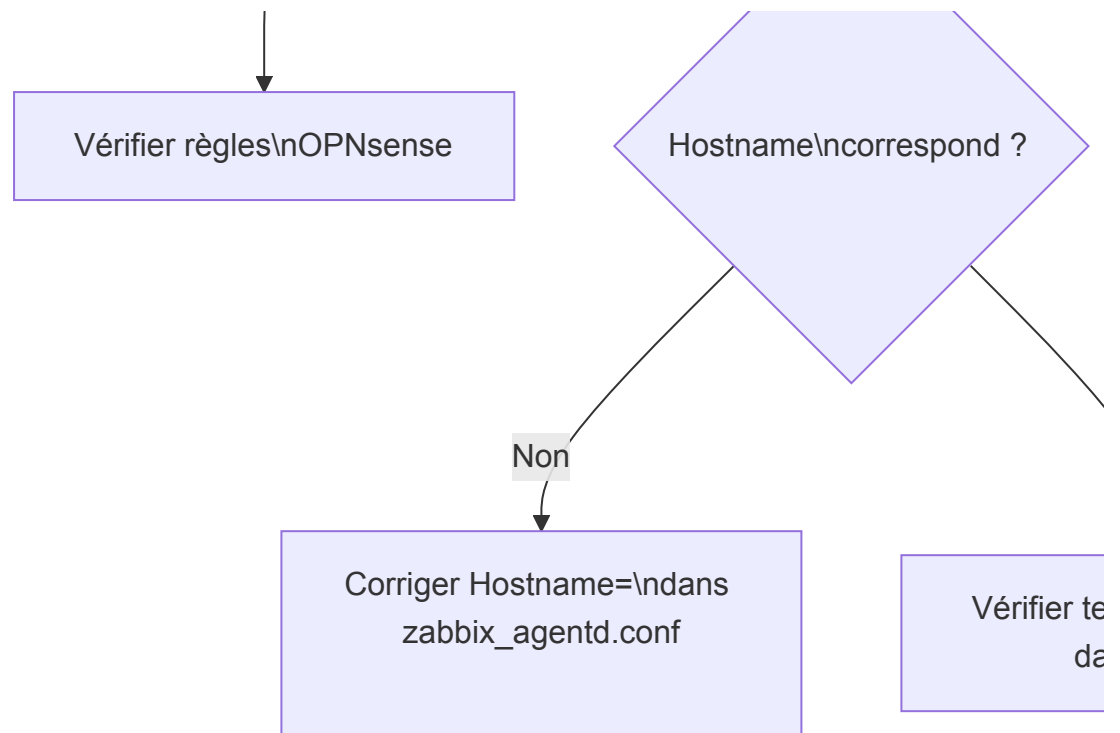
- BTS-SIO
- SISR
- E6
- Zabbix
- Dépannage
- Troubleshooting

parents: "[00 - Index du projet Zabbix](#)"

Z13 – Dépannage

 **Arbre de décision général**





● Problème 1 – L'agent ne démarre pas

Symptôme : `systemctl status zabbix-agent` affiche `failed` ou `inactive`

Diagnostic :

```
# Lancer l'agent en mode foreground pour voir l'erreur exacte
sudo /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf -f
```

Causes fréquentes :

| Erreur affichée | Solution |

|---|---|

| `cannot bind to socket` | Un autre processus utilise le port 10050 → `sudo kill $(lsof -t -i:10050)` |

| `configuration file not found` | Vérifier le chemin du fichier de config |

| `invalid hostname` | Supprimer les caractères spéciaux du champ `Hostname=` |

| `double Hostname=Hostname=xxx` | Supprimer la ligne en double, ne garder que `Hostname=xxx` |

● Problème 2 – Icône ZBX grise dans Zabbix

Symptôme : L'hôte est créé mais l'icône ZBX reste grise

Vérification dans l'ordre :

```
# 1. L'agent tourne-t-il ?
```

```
sudo systemctl status zabbix-agent
```

```
# 2. Le port est-il accessible depuis le serveur Zabbix ?
```

```
nc -zv <IP_AGENT> 10050
```

```
# 3. L'agent répond-il correctement ?
```

```
zabbix_get -s <IP_AGENT> -p 10050 -k agent.hostname
```

```
# 4. Le hostname retourné correspond-il à celui dans Zabbix ?
```

```
# Comparer avec : Collecte de données → Hôtes → Nom d'hôte
```

Checklist complète :

- L'agent tourne (active (running))
- Le port 10050 est accessible (nc -zv → succeeded)
- Le Hostname= dans le fichier conf = Nom d'hôte dans Zabbix
- Un template est bien assigné à l'hôte
- L'IP de l'hôte dans Zabbix correspond à la vraie IP de la machine

● Problème 3 – Pas de connectivité inter-VLAN

Symptôme : ping 192.168.4.10 ne répond pas depuis la VM VLAN 50

Vérification dans l'ordre :

```
# 1. La gateway VLAN 50 répond-elle ?
```

```
ping 192.168.50.254
```

```
# 2. La configuration IP est-elle correcte ?
```

```
ip a
```

```
ip route show
```

Causes fréquentes :

| Cause | Vérification | Solution |

|---|---|---|

| VLAN Aware désactivé | Proxmox → vmbr0 → Propriétés | Cocher "Gère les VLAN" → Appliquer |

| Règles OPNsense manquantes | Firewall → Règles → OPT1 et OPT3 | Créer les règles ([Z09 - Règles OPNsense](#)) |

| VLAN Tag incorrect sur la VM | Proxmox → VM → Hardware → Network | Vérifier que VLAN Tag = 50 |

| Port switch mal configuré | Accès console switch | Vérifier le port en access VLAN 50 ou trunk |

🟡 Problème 4 – Erreur `nmcli` propriété non valide

Symptôme :

```
Error: Failed to modify connection '...': 'ipv4.adresses' is not a valid...
```

Solution : Vérifier l'orthographe exacte des propriétés nmcli :

| ❌ Incorrect | ✅ Correct |

|---|---|

| ipv4.adresses | ipv4.addresses (deux s) |

| ipv4.gatway | ipv4.gateway |

| ipv4.methode | ipv4.method |

🟢 Problème 5 – Pas d'email d'alerte reçu

Symptôme : La VM est éteinte mais aucun email n'arrive

Vérification dans l'ordre :

1. **Action activée ?** → Alertes → Actions → Actions de déclencheur → Report problems to Zabbix administrators doit être **Activé**
 2. **Média configuré sur l'utilisateur ?** → Utilisateurs → Admin → Médias → Gmail présent
 3. **Mot de passe d'application correct ?** → Retester avec un nouveau mot de passe applicatif Google
 4. **Délai du déclencheur ?** → Le déclencheur par défaut attend **5 minutes** avant d'alerter
-

🔗 Navigation

← Retour : [Z12 - Tests et vérifications](#)

← Index : [00 - Index du projet Zabbix](#)

tags:

- BTS-SIO
- SISR
- E6
- Grafana
- Supervision
- Zabbix

created: 2026-03-12

statut: En cours

Complément E6 – Visualisation avec Grafana

Informations du projet

Étudiant : Matthis FINOCCHI

Formation : BTS SIO option SISR

Établissement : UTEC Avon-Fontainebleau

Alternance : Elionis

Année : 2024–2025

Lien avec le projet Zabbix

Ce complément s'appuie sur l'infrastructure déjà en place dans le [projet Zabbix](#). Grafana est installé sur la **même VM** que Zabbix Server (192.168.4.10) et se connecte directement à la base MySQL Zabbix.

Navigation du dossier

| # | Note | Description |

|---|-----|-----|

| 01 | [G01 - Contexte et objectifs](#) | Pourquoi Grafana en complément de Zabbix |

| 02 | [G02 - Installation Grafana](#) | Installation sur la VM Zabbix (192.168.4.10) |

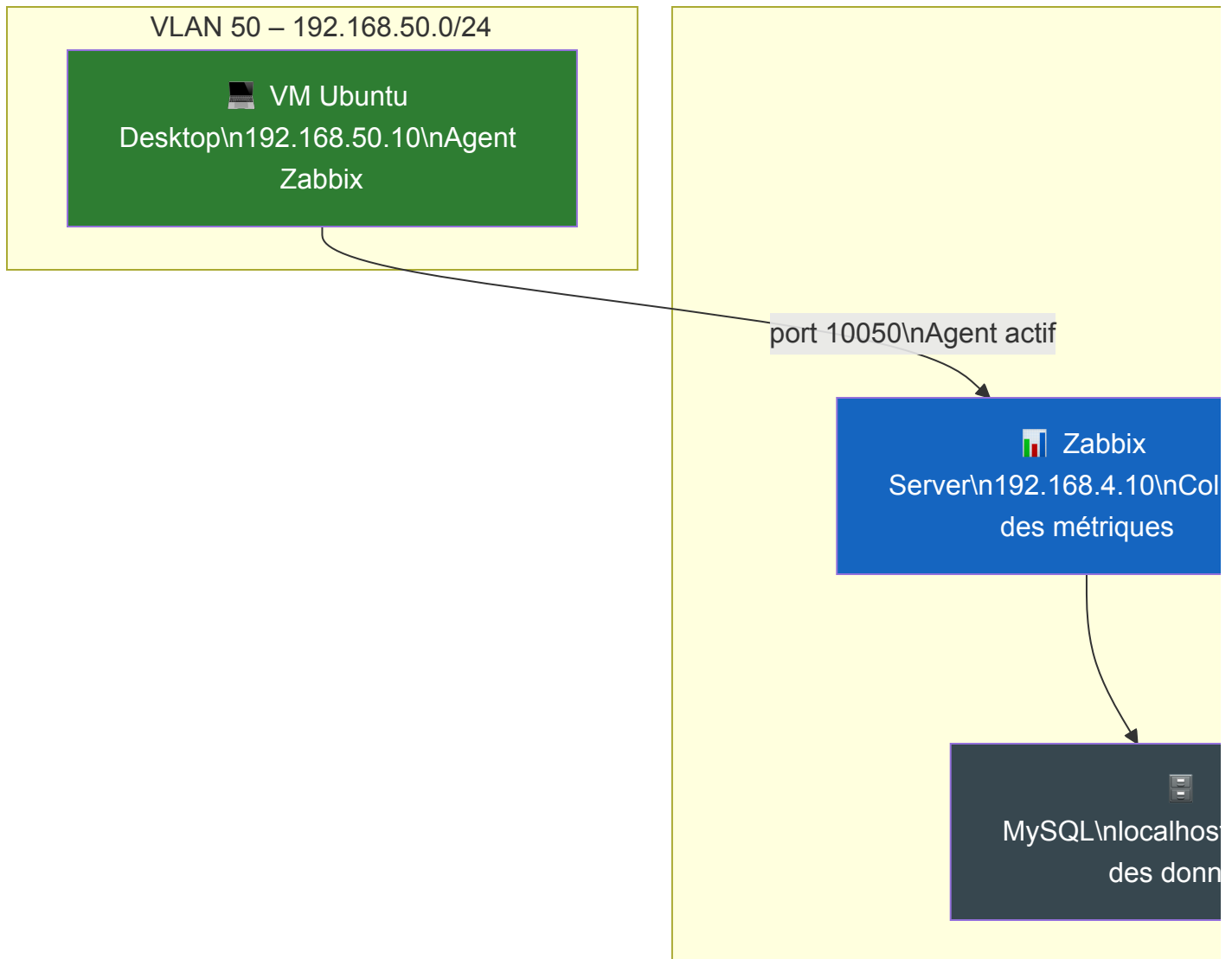
| 03 | [G03 - Datasource Zabbix](#) | Connexion Grafana → MySQL Zabbix |

| 04 | [G04 - Dashboards](#) | Création des tableaux de bord |

| 05 | [G05 - Alertes Grafana](#) | Configuration des alertes visuelles |

| 06 | [G06 - Tests et vérifications](#) | Validation du fonctionnement complet |

Architecture complète (Zabbix + Grafana)



Comparaison Zabbix vs Grafana

| Fonctionnalité | Zabbix | Grafana |

|---|---|---|

| Collecte des métriques | Natif | Non (utilise Zabbix) |

| Alertes | Email, SMS... | Email, Slack, PagerDuty... |

| Dashboards | Basiques | Riches et personnalisables |

| Visualisation graphique | Correcte | Excellente |

| Corrélation multi-sources | ❌ Limité | ✅ Multi-datasource |

| Interface moderne | ⚠️ Fonctionnelle | ✅ Très moderne |

📄 Complémentarité

Zabbix collecte, Grafana affiche. Les deux outils sont complémentaires : Zabbix reste le moteur de supervision, Grafana apporte une couche de visualisation professionnelle.

📊 Avancement

- Grafana installé
 - Datasource Zabbix connectée
 - Dashboard système créé
 - Dashboard réseau créé
 - Alertes configurées
 - Tests validés
-

🔗 Liens

Projet principal : [Z00 - Index du projet Zabbix](#) | Portfolio : [mfiportfolio.fr](#)

tags:

- BTS-SIO
- SISR
- E6
- Grafana
- Contexte

parents: "[G00 - Index Grafana](#)"

G01 – Contexte et objectifs

Pourquoi ajouter Grafana à l'infrastructure Zabbix ?

Le projet [Zabbix](#) est déjà fonctionnel : les agents remontent les métriques, les alertes Gmail fonctionnent, et les hôtes sont supervisés. Cependant, l'interface de Zabbix reste **orientée administration** — les graphiques sont fonctionnels mais peu lisibles pour un responsable IT non technique.

Grafana apporte une couche de **visualisation professionnelle** sans toucher à l'infrastructure Zabbix existante.

Contexte TechNova Solutions

Suite au déploiement de Zabbix, la direction de **TechNova Solutions** souhaite disposer d'un **tableau de bord de supervision** accessible à l'équipe IT et aux managers, présentant :

- L'état en temps réel des serveurs
 - L'utilisation CPU, RAM et disque des VMs supervisées
 - L'historique des incidents réseau
 - Des alertes visuelles en cas de dépassement de seuil
-

Objectifs du complément Grafana

1 – Visualisation avancée

- Créer des dashboards clairs avec graphiques en courbes, jauges et heatmaps
- Afficher les métriques collectées par Zabbix dans une interface moderne
- Permettre le filtrage par hôte, par VLAN, par période

2 – Centralisation

- Connecter Grafana directement à la base MySQL de Zabbix (même VM)
- Un seul point d'accès pour toutes les métriques : `http://192.168.4.10:3000`

3 – Alertes visuelles

- Configurer des seuils d'alerte directement dans Grafana
 - Alertes par email en complément de celles de Zabbix
-

Choix techniques

| Choix | Justification |

|---|---|

| **Grafana OSS** (open-source) | Gratuit, communauté active, parfait pour un labo BTS |

| **Datasource : Zabbix plugin** | Plugin officiel Grafana pour lire directement les données Zabbix |

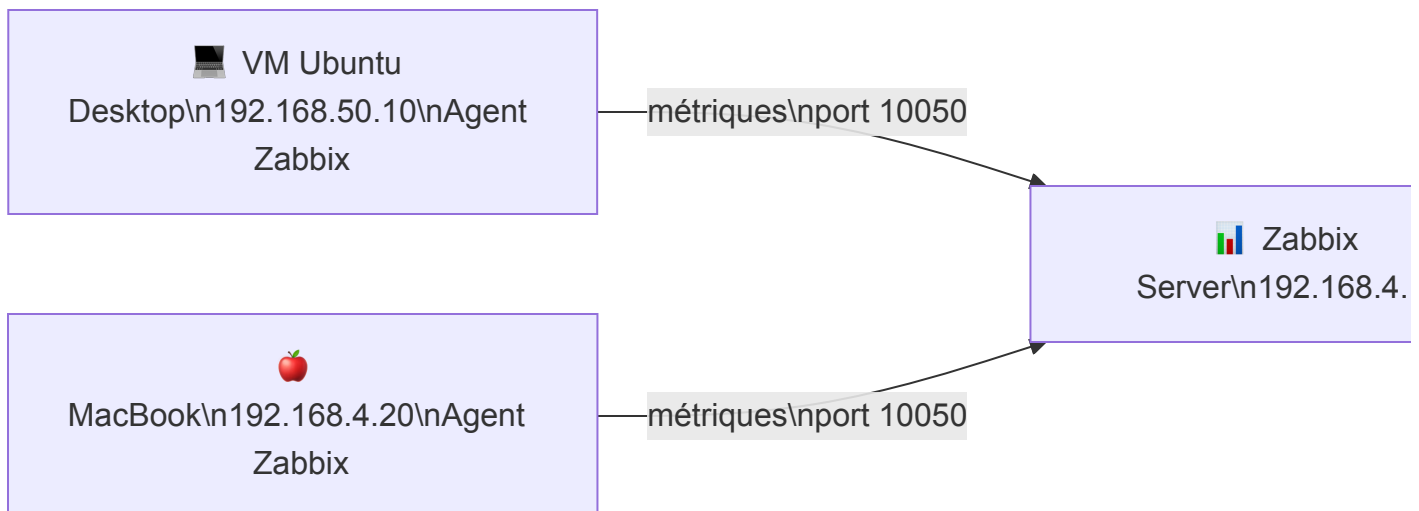
| **Même VM que Zabbix** | Évite de créer une VM supplémentaire, simplifie l'architecture |

| **Port 3000** | Port par défaut de Grafana, distinct du port 80 d'Apache/Zabbix |

Version utilisée

Grafana OSS 10.x — dernière version stable au moment du projet.

Flux de données



Navigation

← Index : [G00 - Index Grafana](#)

→ Suite : [G02 - Installation Grafana](#)

tags:

- BTS-SIO
- SISR
- E6
- Grafana
- Installation
- Linux

parents: "[G00 - Index Grafana](#)"

G02 – Installation de Grafana

Machine concernée

Toutes les commandes s'exécutent sur la **VM Zabbix Server** (192.168.4.10, Ubuntu Server 24.04), en SSH ou via la console Proxmox.

Prérequis

Le projet [Zabbix Server](#) doit être installé et fonctionnel avant de commencer.

Étape 1 – Ajout du dépôt Grafana

Grafana n'est pas dans les dépôts Ubuntu par défaut. On ajoute le dépôt officiel :

```
# Installation des dépendances pour HTTPS

sudo apt install -y apt-transport-https software-properties-common wget

# Ajout de la clé GPG officielle Grafana

sudo mkdir -p /etc/apt/keyrings

wget -q -O - https://apt.grafana.com/gpg.key | \
```

```
gpg --dearmor | \  
  
sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null  
  
# Ajout du dépôt stable Grafana  
  
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] \  
https://apt.grafana.com stable main" | \  
  
sudo tee /etc/apt/sources.list.d/grafana.list  
  
# Mise à jour de la liste des paquets  
  
sudo apt update
```

Étape 2 – Installation de Grafana OSS

```
sudo apt install -y grafana
```

Grafana OSS vs Enterprise

On installe **Grafana OSS** (Open Source) — la version gratuite, largement suffisante pour un projet BTS. La version Enterprise est payante.

Étape 3 – Démarrage et activation du service

```
sudo systemctl start grafana-server
```

```
sudo systemctl enable grafana-server
```

| Commande | Rôle |

|---|---|

| start | Lance Grafana immédiatement |

| enable | Démarre Grafana automatiquement à chaque reboot |

✅ Étape 4 – Vérification du service

```
sudo systemctl status grafana-server
```

✓ Résultat attendu

| Le statut doit afficher **active (running)** en vert.

Vérifier aussi que le port 3000 est bien en écoute :

```
ss -tlnp | grep 3000
```

Résultat attendu :

```
LISTEN 0 128 0.0.0.0:3000 0.0.0.0:* users:(("grafana",pid=XXXX,...))
```

🔥 Étape 5 – Ouverture du port dans le pare-feu (si UFW actif)

```
# Vérifier si UFW est actif
```

```
sudo ufw status
```

```
# Si actif, autoriser le port 3000
```

```
sudo ufw allow 3000/tcp
```

```
sudo ufw reload
```

Étape 6 – Première connexion

Depuis un navigateur sur le réseau VLAN 4 :

```
http://192.168.4.10:3000
```

Identifiants par défaut :

| Champ | Valeur |

|---|---|

| Utilisateur | admin |

| Mot de passe | admin |

Changer le mot de passe

Grafana demande immédiatement de changer le mot de passe par défaut. Choisir un mot de passe sécurisé.

Étape 7 – Configuration de base (grafana.ini)

Le fichier de configuration principal de Grafana :

```
sudo nano /etc/grafana/grafana.ini
```

Modifier les lignes suivantes (décommenter en supprimant le ;) :

```
[server]

# Adresse d'écoute (toutes les interfaces)

http_addr = 0.0.0.0

# Port d'écoute

http_port = 3000

# URL publique de Grafana

root_url = http://192.168.4.10:3000

[security]

# Désactiver l'enregistrement public

allow_sign_up = false

# Utilisateur admin

admin_user = admin

[users]

# Empêcher les utilisateurs de créer des organisations

allow_org_create = false
```

```
# Redémarrer pour appliquer
```

```
sudo systemctl restart grafana-server
```

Arborescence des fichiers Grafana

```
/etc/grafana/
```

```
├─ grafana.ini ← configuration principale
```

```
└─ provisioning/
```

```
├─ datasources/ ← datasources auto-provisionnées
```

```
├─ dashboards/ ← dashboards auto-provisionnés
```

```
└─ alerting/ ← alertes auto-provisionnées
```

```
/var/lib/grafana/
```

```
├─ grafana.db ← base SQLite (config Grafana)
```

```
└─ plugins/ ← plugins installés
```

```
/var/log/grafana/
```

```
└─ grafana.log ← logs du service
```

Navigation

← Retour : [G01 - Contexte et objectifs](#)

→ Suite : [G03 - Datasource Zabbix](#)

tags:

- BTS-SIO
- SISR
- E6
- Grafana
- Datasource
- Zabbix
- MySQL

parents: "[G00 - Index Grafana](#)"

G03 – Connexion Grafana → Zabbix (Datasource)

Deux méthodes possibles

Il existe deux façons de connecter Grafana aux données Zabbix :

| Méthode | Description | Recommandée |

|---|---|---|

| **Plugin Grafana-Zabbix** | Plugin officiel qui interroge l'API Zabbix | Oui |

| **Datasource MySQL directe** | Connexion directe à la base MySQL de Zabbix |  Avancé |

On utilise les **deux** : le plugin pour les métriques temps réel, MySQL pour les requêtes personnalisées.

Méthode 1 – Plugin Grafana-Zabbix

Étape 1 – Installation du plugin

```
sudo grafana-cli plugins install alexanderzobnin-zabbix-app
```

```
sudo systemctl restart grafana-server
```

Info

`grafana-cli` est l'outil en ligne de commande de Grafana pour gérer les plugins. Le plugin `alexanderzobnin-zabbix-app` est le plugin Zabbix officiel, maintenu par la communauté.

Étape 2 – Activer le plugin dans l'interface

Chemin : Administration → Plugins → Rechercher "Zabbix" → **Enable**

Étape 3 – Ajouter la datasource Zabbix

Chemin : Connexions → Sources de données → Ajouter une source de données → **Zabbix**

| Champ | Valeur |

|---|---|

| Nom | `Zabbix-Local` |

| URL | `http://192.168.4.10/zabbix/api_jsonrpc.php` |

| Utilisateur Zabbix | `Admin` |

| Mot de passe Zabbix | `zabbix` (ou ton mot de passe modifié) |

| Zabbix version | `7.x` |

URL de l'API Zabbix

L'API JSON-RPC de Zabbix est accessible à l'adresse `/zabbix/api_jsonrpc.php`. C'est l'interface que Grafana utilise pour interroger Zabbix sans passer par la base de données directement.

Cliquer **Save & Test** → doit afficher "**Datasource is working**" 

Méthode 2 – Datasource MySQL (complément)

Cette méthode permet d'écrire des requêtes SQL personnalisées pour des visualisations avancées.

Étape 1 – Créer un utilisateur MySQL en lecture seule pour Grafana

```
sudo mysql
```

```
-- Créer un utilisateur Grafana en lecture seule  
  
CREATE USER 'grafana'@'localhost' IDENTIFIED BY 'grafana_password';  
  
-- Donner uniquement les droits SELECT sur la base Zabbix  
  
GRANT SELECT ON zabbix.* TO 'grafana'@'localhost';  
  
FLUSH PRIVILEGES;  
  
EXIT;
```

Sécurité

On crée un utilisateur **en lecture seule** (SELECT uniquement). Grafana n'a jamais besoin d'écrire dans la base Zabbix — lui donner tous les droits serait un risque inutile.

Étape 2 – Ajouter la datasource MySQL dans Grafana

Chemin : Connexions → Sources de données → Ajouter → **MySQL**

| Champ | Valeur |

|---|---|

| Nom | Zabbix-MySQL |

| Hôte | localhost:3306 |

| Base de données | zabbix |

| Utilisateur | grafana |

| Mot de passe | grafana_password |

| TLS/SSL | Désactivé (réseau local) |


Cliquer **Save & Test** → doit afficher "**Database Connection OK**" 

Vérification des datasources

Dans **Connexions** → **Sources de données**, tu dois voir :

| Datasource | Type | Statut |

|---|---|---|

| Zabbix-Local | Zabbix |  OK |

| Zabbix-MySQL | MySQL |  OK |

Test rapide de la datasource Zabbix

Dans l'explorateur Grafana (**Explore** dans le menu) :

1. Sélectionner la datasource `Zabbix-Local`
2. **Group** : `Linux servers`
3. **Host** : `pc1-Standard-PC-i440FX-PIIX-1996`
4. **Application** : `CPU`
5. **Item** : `CPU utilization`
6. Cliquer **Run query**

Résultat attendu

Un graphique de l'utilisation CPU de la VM VLAN 50 s'affiche avec les données collectées par Zabbix.

Navigation

← Retour : [G02 - Installation Grafana](#)

→ Suite : [G04 - Dashboards](#)

tags:

- BTS-SIO
- SISR
- E6
- Grafana
- Dashboard
- Visualisation

parents: "[G00 - Index Grafana](#)"

G04 – Création des dashboards

Dashboards à créer

| Dashboard | Description | Hôtes concernés |

|---|---|---|

| **Supervision Globale** | Vue d'ensemble de tous les hôtes | Tous |

| **VM VLAN 50** | Métriques détaillées de la VM Ubuntu Desktop | 192.168.50.10 |

| **MacBook** | Métriques du MacBook (jour examen) | 192.168.4.20 |

Dashboard 1 – Supervision Globale

Création du dashboard

Chemin : Tableaux de bord → Nouveau → Nouveau tableau de bord → Ajouter une visualisation

Panel 1 – État des hôtes (Stat)

Type de visualisation : Stat

Datasource : Zabbix-Local

| Champ | Valeur |

|---|---|

| Group | Linux servers |

| Host | * (tous) |

| Application | Zabbix agent |

| Item | Zabbix agent availability |

Options d'affichage :

- Orientation : Horizontal
 - Mode couleur : Background
 - Seuils : Vert = 1 (disponible), Rouge = 0 (indisponible)
-

Panel 2 – CPU toutes VMs (Time series)

Type de visualisation : Time series

Datasource : Zabbix-Local

| Champ | Valeur |

|---|---|

| Group | Linux servers |

| Host | * |

| Application | CPU |

| Item | CPU utilization |

Options d'affichage :

- Unité : Percent (0-100)
 - Seuil d'alerte : ligne rouge à 80%
 - Légende : activée (un trait par hôte)
-

Panel 3 – RAM disponible (Gauge)

Type de visualisation : Gauge

Datasource : Zabbix-Local

| Champ | Valeur |

|---|---|

| Group | Linux servers |

| Host | pc1-Standard-PC-i440FX-PIIX-1996 |

| Application | Memory |

| Item | Available memory |

Options d'affichage :

- Unité : bytes(IEC)
 - Min : 0 / Max : 4294967296 (4 Go)
 - Seuils : Rouge < 512 Mo, Jaune < 1 Go, Vert > 1 Go
-

Panel 4 – Utilisation disque (Bar gauge)

Type de visualisation : Bar gauge

Datasource : Zabbix-Local

| Champ | Valeur |

|---|---|

| Group | Linux servers |

| Host | * |

| Application | Filesystems |

| Item | Space utilization |

Options d'affichage :

- Unité : Percent (0-100)
 - Seuils : Jaune à 70%, Rouge à 90%
-

Dashboard 2 – VM VLAN 50 (détailé)

Panel 1 – CPU Load (Time series)

Datasource : Zabbix-Local

Group : Linux servers

Host : pc1-Standard-PC-i440FX-PIIX-1996

Item : CPU utilization

Période : Last 1 hour

Panel 2 – Mémoire RAM (Time series)

Datasource : Zabbix-Local

Host : pc1-Standard-PC-i440FX-PIIX-1996

Item : Available memory in %

Unité : Percent (0-100)

Panel 3 – Trafic réseau (Time series)

Datasource : Zabbix-Local

Host : pc1-Standard-PC-i440FX-PIIX-1996

Items : Network interface eth0 – Bits received

Network interface eth0 – Bits sent

Unité : bits/sec

Panel 4 – Charge système (Stat)

Datasource : Zabbix-Local

Host : pc1-Standard-PC-i440FX-PIIX-1996

Items : System uptime

Number of processes

Number of logged in users

Dashboard 3 – Requête MySQL personnalisée

Pour afficher l'historique des alertes Zabbix dans Grafana :

Datasource : Zabbix-MySQL

Type : Table

```
SELECT  
  
FROM_UNIXTIME(e.clock) AS 'Date/Heure',  
  
h.host AS 'Hôte',  
  
e.name AS 'Problème',  
  
CASE e.severity  
  
WHEN 0 THEN 'Non classifié'  
  
WHEN 1 THEN 'Information'  
  
WHEN 2 THEN 'Avertissement'  
  
WHEN 3 THEN 'Moyen'  
  
WHEN 4 THEN 'Élevé'  
  
WHEN 5 THEN 'Critique'
```

```
END AS 'Sévérité'  
  
FROM events e  
  
JOIN hosts h ON e.objectid = h.hostid  
  
WHERE e.source = 0  
  
AND e.object = 0  
  
AND e.clock >= UNIX_TIMESTAMP(NOW() - INTERVAL 24 HOUR)  
  
ORDER BY e.clock DESC  
  
LIMIT 50;
```

Utilité de cette requête

Cette table montre les **50 derniers événements des dernières 24h** directement depuis la base MySQL de Zabbix — idéal pour la présentation E6 !

Sauvegarder et organiser les dashboards

```
# Les dashboards sont stockés dans la base SQLite de Grafana  
  
# Pour les exporter en JSON (backup) :  
  
# Tableau de bord → icône Partager → Exporter → Enregistrer en JSON
```

Dossiers de dashboards

Organiser les dashboards dans des dossiers :

-  **TechNova – Supervision** → Dashboard Globale
-  **TechNova – Détails** → VM VLAN 50, MacBook

Navigation

← Retour : [G03 - Datasource Zabbix](#)

→ Suite : [G05 - Alertes Grafana](#)

tags:

- BTS-SIO
- SISR
- E6
- Grafana
- Alertes
- Email

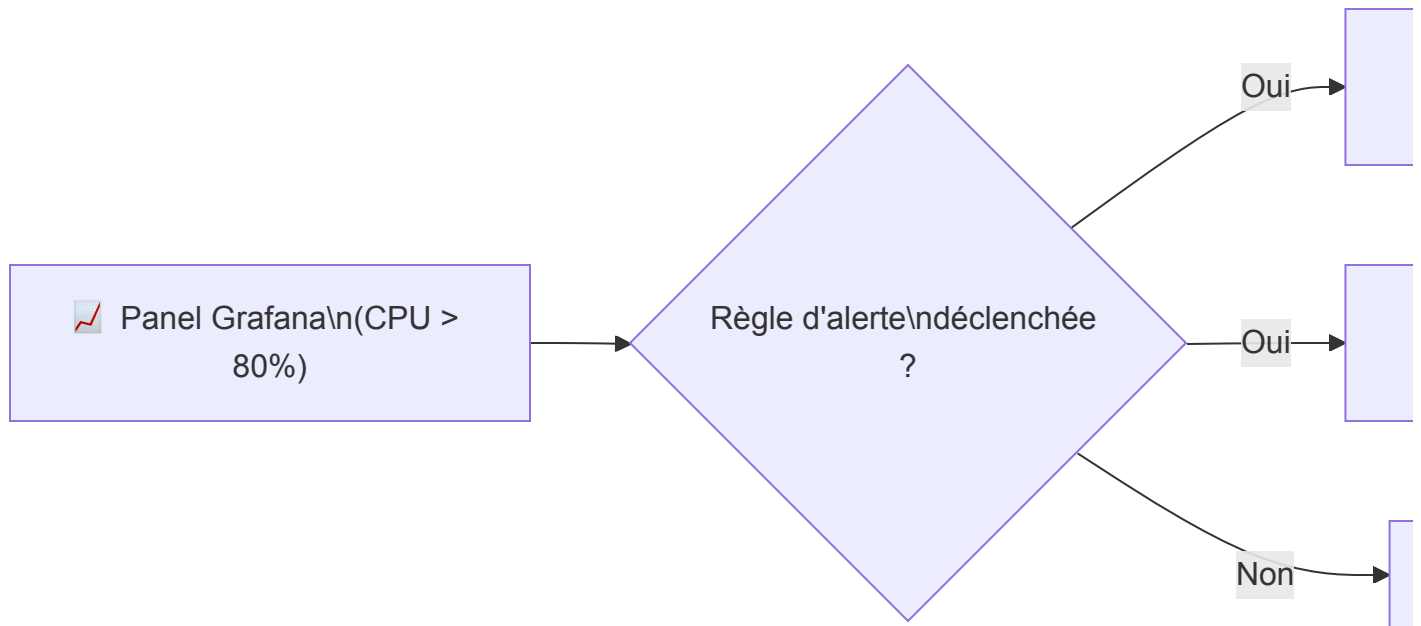
parents: "[G00 - Index Grafana](#)"

G05 – Alertes Grafana

Complémentarité avec Zabbix

Les alertes Grafana complètent celles de [Zabbix](#) — elles sont visuelles (panneau rouge dans le dashboard) et peuvent envoyer des notifications par email, en parallèle de Zabbix.

Vue d'ensemble du système d'alertes



🔧 Étape 1 – Configurer le serveur SMTP (contact point email)

Grafana doit savoir comment envoyer des emails. On réutilise le même compte Gmail que Zabbix.

```
sudo nano /etc/grafana/grafana.ini
```

Modifier la section [smtp] :

```
[smtp]

enabled = true

host = smtp.gmail.com:587

user = alertezabbixmatthis@gmail.com

# Mot de passe d'application Google (16 caractères, sans espaces)

password = xxxx xxxx xxxx xxxx

skip_verify = true
```

```
from_address = alertezabbixmatthis@gmail.com
```

```
from_name = Grafana TechNova
```

```
sudo systemctl restart grafana-server
```

Mot de passe d'application

Utiliser le **même mot de passe d'application Google** que celui configuré pour Zabbix ([étape Gmail Zabbix](#)), ou en créer un nouveau dédié à Grafana.

Étape 2 – Créer un Contact Point

Chemin : Alertes → Contact Points → Nouveau contact point

| Champ | Valeur |

|---|---|

| Nom | Email-TechNova |

| Type | Email |

| Adresse email | alertezabbixmatthis@gmail.com |

Cliquer **Test** pour envoyer un email de test → vérifier la réception 

Cliquer **Save contact point**.

Étape 3 – Créer une politique de notification

Chemin : Alertes → Politique de notification → Modifier la politique par défaut

| Champ | Valeur |

|---|---|

| Contact point par défaut | Email-TechNova |

| Grouper par | alertname , grafana_folder |

| Délai avant envoi | 1 min |

| Délai entre répétitions | 4 h |

● Étape 4 – Créer des règles d'alerte

Alerte 1 – CPU élevé (> 80%)

Chemin : Alertes → Règles d'alerte → Nouvelle règle d'alerte

| Champ | Valeur |

|---|---|

| Nom | CPU élevé – VM VLAN 50 |

| Datasource | Zabbix-Local |

| Group | Linux servers |

| Host | pc1-Standard-PC-i440FX-PIIX-1996 |

| Item | CPU utilization |

Conditions :

```
WHEN : last() OF query(A, 5m, now)
```

```
IS ABOVE : 80
```

| Champ | Valeur |

|---|---|

| Dossier | TechNova – Supervision |

| Groupe d'évaluation | Créer : default (évaluation toutes les 1 min) |

| Délai avant déclenchement | 2 min |

| Résumé | CPU de {{ \$labels.host }} à {{ \$values.A }}% |

Alerte 2 – Mémoire RAM faible (< 15% disponible)

| Champ | Valeur |

|---|---|

| Nom | RAM faible – VM VLAN 50 |

| Item | Available memory in % |

| Condition | IS BELOW 15 |

| Délai | 2 min |

| Résumé | RAM disponible de {{ \$labels.host }} : {{ \$values.A }}% |

Alerte 3 – Hôte inaccessible

| Champ | Valeur |

|---|---|

| Nom | Hôte inaccessible |

| Item | Zabbix agent availability |

| Condition | IS BELOW 1 |


| Délai | 5 min |

| Résumé | L'agent Zabbix sur {{ \$labels.host }} ne répond plus |

Récapitulatif des alertes configurées

| Alerte | Seuil | Délai | Sévérité |

|---|---|---|---|

| CPU élevé | > 80% | 2 min |  Warning |

| RAM faible | < 15% dispo | 2 min |  Warning |

| Disque plein | > 90% utilisé | 5 min |  Critical |

Test des alertes

```
# Simuler une surcharge CPU depuis la VM cible (192.168.50.10)

# Lancer un stress test pendant 5 minutes

sudo apt install -y stress

stress --cpu 4 --timeout 300
```

Résultat attendu

1. Le panel CPU dans Grafana passe au **rouge**
2. Un email d'alerte arrive sur `alertezabbixmatthis@gmail.com`
3. Quand le stress se termine, le panel repasse au **vert** + email de résolution

Navigation

← Retour : [G04 - Dashboards](#)

→ Suite : [G06 - Tests et vérifications](#)

tags:

- BTS-SIO
- SISR
- E6
- Grafana
- Tests
- Validation


parents: "[G00 - Index Grafana](#)"


G06 – Tests et vérifications


Tableau de tests complet

| # | Test | Commande / Action | Résultat attendu | Statut |

|---|---|---|---|---|


| 1 | Service Grafana actif | `systemctl status grafana-server` | active (running) |  |


| 2 | Port 3000 en écoute | `ss -tlnp \| grep 3000` | LISTEN sur 0.0.0.0:3000 |  |


| 3 | Interface web accessible | `http://192.168.4.10:3000` | Page de login Grafana |  |

| 4 | Datasource Zabbix plugin | Save & Test dans Grafana | "Datasource is working" |  |


| 5 | Datasource MySQL | Save & Test dans Grafana | "Database Connection OK" |  |

| 6 | Dashboard Supervision Globale | Ouvrir le dashboard | Données temps réel affichées |  |

| 7 | Panel CPU VM VLAN 50 | Dashboard → VM VLAN 50 | Courbe CPU visible |  |

| 8 | Contact point email | Alertes → Test | Email de test reçu |  |


| 9 | Alerte CPU déclenchée | `stress --cpu 4 --timeout 300` | Email + panel rouge |  |

| 10 | Résolution alerte | Fin du stress test | Email résolution + panel vert |  |

Détail des tests clés

Test 4 – Datasource Zabbix plugin

Connexions → Sources de données → Zabbix-Local → Save & Test

→  "Datasource is working. Zabbix version: 7.4.x"

Test 5 – Datasource MySQL

Connexions → Sources de données → Zabbix-MySQL → Save & Test

→  "Database Connection OK"

Test 9 – Simulation surcharge CPU

Depuis la VM cliente (192.168.50.10) :

```
# Installer l'outil stress

sudo apt install -y stress

# Lancer 4 workers CPU pendant 5 minutes

stress --cpu 4 --timeout 300
```

Depuis Grafana (192.168.4.10:3000) :

- Observer le panel CPU passer de ~5% à ~100%
- Après 2 minutes de dépassement → le panel passe **rouge**
- Vérifier la réception d'un email d'alerte

Checklist finale avant présentation E6

- Grafana installé et accessible sur `http://192.168.4.10:3000`
- Plugin Zabbix installé et activé
- Datasource Zabbix-Local connectée (API Zabbix)
- Datasource Zabbix-MySQL connectée (MySQL local)
- Dashboard "Supervision Globale" créé avec données temps réel

- ☑ Dashboard "VM VLAN 50" créé avec métriques détaillées
- ☑ Requête SQL historique alertes fonctionnelle
- ☑ Contact point email configuré et testé
- ☑ Règles d'alerte CPU et RAM créées
- ☑ Test stress CPU effectué → alerte reçue par email

Récapitulatif des accès

| Service | URL / Commande | Identifiants |

|---|---|---|

| Zabbix | `http://192.168.4.10/zabbix` | Admin / zabbix |

| Grafana | `http://192.168.4.10:3000` | admin / *ton mdp* |

| MySQL | `sudo mysql -uzabbix -p` | zabbix / password |

Pour aller plus loin

Améliorations possibles pour l'oral E6

- Importer un **dashboard pré-fait depuis Grafana.com** (ex: ID 7039 — Zabbix by Alexanderzobnin)
- Ajouter une **variable de template** dans le dashboard pour filtrer par hôte dynamiquement
- Connecter Grafana à **Loki** pour afficher les logs Snort ([projet Snort IDS](#))
- Exposer Grafana en HTTPS avec un certificat auto-signé

Navigation

← Retour : [G05 - Alertes Grafana](#)

← Index Grafana : [G00 - Index Grafana](#)

← Index Zabbix : [Z00 - Index du projet Zabbix](#)