



Mise en place d'un serveur Zabbix

Table des matières

Cahier des charges – Expression des besoins	3
Descriptif de l'existant 	3
Besoin(s) 	3
Contrainte(s) 	3
Ressources	3
Ressources mises à disposition 	3
Ressources nécessaires à la mise en place 	4
Gestion des ressources 	4
Analyse	4
Descriptifs des solutions 	4
Comparaison des solutions 	5
Choix d'une solution – Argumentation 	5
Plan d'adressage réseau 	6
Etude de l'impact sur le SI existant 	7
Phasage de l'intervention 	7
Prévision des tests et validation 	7
Mise en place	8
Partitionnement serveur Linux 	8
Déploiement de la solution 	8
Installation et enregistrement de l'agent Zabbix sur le client Linux 	11
Installation et enregistrement de l'agent Zabbix sur le client Windows 	11
Création des hôtes dans l'interface Zabbix 	12
Vérification de la collecte des données 	13
Test du déclenchement d'une alerte 	14
Conclusion	14
Auto-évaluation.....	14

Cahier des charges – Expression des besoins

| Descriptif de l'existant |

L'infrastructure de départ étant inexistante, je ne dispose d'aucun service réseau préconfiguré. Mon environnement se limite à un accès à Internet via le réseau local (LAN) du GRETA, qui me permet de télécharger les ressources nécessaires à la mise en place du projet.

| Besoin(s) |

Dans le cadre d'un travail pratique à visée pédagogique, j'ai pour objectif d'installer Zabbix sur une machine virtuelle fonctionnant sous Debian 13, dans le but de superviser de manière centralisée un parc mixte composé de serveurs Linux et Windows. La solution devra permettre de collecter les indicateurs de performance (CPU, mémoire, disques, réseau), de générer des alertes en cas de dépassement de seuils, et d'historiser les données pour analyse.

| Contrainte(s) |

Dans le cadre de ce travail pratique, je dois mettre en place, en un temps imparti de quatre heures, une machine virtuelle sous Debian 13 destinée à héberger le serveur Zabbix ainsi que sa base de données et son interface web. Parallèlement, je dois créer une machine virtuelle Linux et une machine virtuelle Windows 10 entièrement configurées, qui serviront de postes clients supervisés par l'agent Zabbix. Je dois également mettre en place un poste d'administration Windows permettant d'accéder à l'interface web Zabbix depuis le réseau interne. Enfin, je dois installer et configurer les agents Zabbix sur chaque poste client afin de garantir la communication avec le serveur de supervision sur le port 10050/10051.

Ressources

| Ressources mises à disposition |

Pour la réalisation de ce travail pratique, je dispose d'une machine hôte équipée de l'hyperviseur Hyper-V, utilisé comme solution de virtualisation. De plus, cette machine est connectée au réseau local (LAN) du GRETA, ce qui me permet de bénéficier d'un accès à Internet nécessaire au téléchargement et à la configuration des différents composants du projet.

| Ressources nécessaires à la mise en place |

Pour la mise en œuvre de ce travail pratique, plusieurs éléments matériels et logiciels sont nécessaires :

- Une machine hôte équipée d'un logiciel de virtualisation (dans ce cas, Hyper-V) afin de créer et gérer les différentes machines virtuelles ;
- Un accès à Internet, indispensable pour le téléchargement des ressources et la mise à jour des paquets ;
- Les images ISO des systèmes d'exploitation Debian 13 et Windows 10, utilisées respectivement pour le serveur Zabbix et les postes clients ;
- Le dépôt officiel Zabbix, disponible à l'adresse : <https://www.zabbix.com/download> ;
- Les agents Zabbix Linux et Windows, téléchargeables depuis : https://www.zabbix.com/download_agents.

| Gestion des ressources |

Étant donné que le temps imparti pour ce travail pratique est limité à quatre heures, et qu'il inclut l'installation complète du serveur Debian 13 destiné à héberger Zabbix ainsi que la mise en place des postes clients Linux et Windows, j'ai pré-téléchargé les images ISO correspondantes. Cette préparation en amont permet de gagner du temps lors de la création des machines virtuelles et d'optimiser le déroulement du TP.

Analyse

| Descriptifs des solutions |

- **Zabbix** est une solution open source de supervision orientée entreprise, capable de surveiller des serveurs, des équipements réseau, des bases de données, des applications et des services cloud. Elle repose sur une architecture serveur/agent et propose une interface web complète permettant la configuration, la consultation des données et la gestion des alertes. Zabbix se distingue par ses templates prêts à l'emploi pour Linux, Windows, SNMP, JMX, ainsi que par son moteur d'alerting puissant (notifications mail, SMS, webhooks, escalades). Son auto-découverte et ses dashboards personnalisables en font une solution moderne et évolutive.
- **Nagios Core** est une solution historique de supervision open source, considérée comme l'une des plus matures du marché. Elle fonctionne par exécution périodique de plugins (NRPE, check_nt, etc.) qui retournent un statut OK / WARNING / CRITICAL. Sa très large communauté met à disposition des milliers de plugins. En revanche, sa configuration repose principalement sur des fichiers texte, son interface web est limitée par défaut, et la gestion des graphiques nécessite des extensions tierces (PNP4Nagios, Grafana). Bien adapté aux environnements Unix traditionnels, Nagios Core demande davantage d'efforts d'intégration qu'une solution moderne.

| Comparaison des solutions |

Solutions	Zabbix	Nagios Core
Coûts	Gratuit (open source)	Gratuit (open source)
OS d'installation	Linux (Debian, Ubuntu, RHEL, etc.)	Linux uniquement
Points forts	Interface web moderne, supervision agent et SNMP, auto-découverte, dashboards personnalisables, alerting avancé, templates intégrés Linux/Windows	Très grande maturité, très large communauté de plugins, supervision basée sur les checks actifs/passifs
Interface / utilisation	Interface web complète et intuitive, configuration entièrement graphique	Interface web vieillissante, configuration principalement par fichiers texte
Mise en place	Installation rapide via dépôt officiel, assistant d'installation web	Plus longue, demande l'édition manuelle de plusieurs fichiers de configuration
Communauté / Support	Communauté active, documentation officielle complète, support payant disponible	Communauté historique, nombreuses ressources mais documentation moins centralisée
Idéal pour	PME, écoles, entreprises souhaitant une supervision moderne et centralisée multi-OS	Environnements unix historiques, supervision réseau pure
Limites	Consomme plus de ressources que Nagios sur de très petites infrastructures	Pas d'interface graphique de configuration native, ergonomie datée

| Choix d'une solution – Argumentation |

J'ai choisi d'utiliser **Zabbix** pour ce travail pratique car il s'agit d'une solution libre, moderne et particulièrement adaptée à la supervision d'un parc mixte Linux/Windows. Sa configuration se fait intégralement depuis l'interface web, ce qui simplifie grandement la prise en main par rapport à Nagios. De plus, Zabbix fournit nativement des templates prêts à l'emploi pour les systèmes Linux et Windows, ce qui réduit considérablement le temps de paramétrage. Enfin, Zabbix dispose d'un moteur d'alerting et de tableaux de bord personnalisables qui répondent parfaitement aux besoins exprimés. Ce choix s'inscrit dans une logique d'efficacité, de rapidité et de maîtrise de l'environnement technique, tout en utilisant une solution gratuite, accessible et largement répandue en milieu professionnel.

| Plan d'adressage réseau |

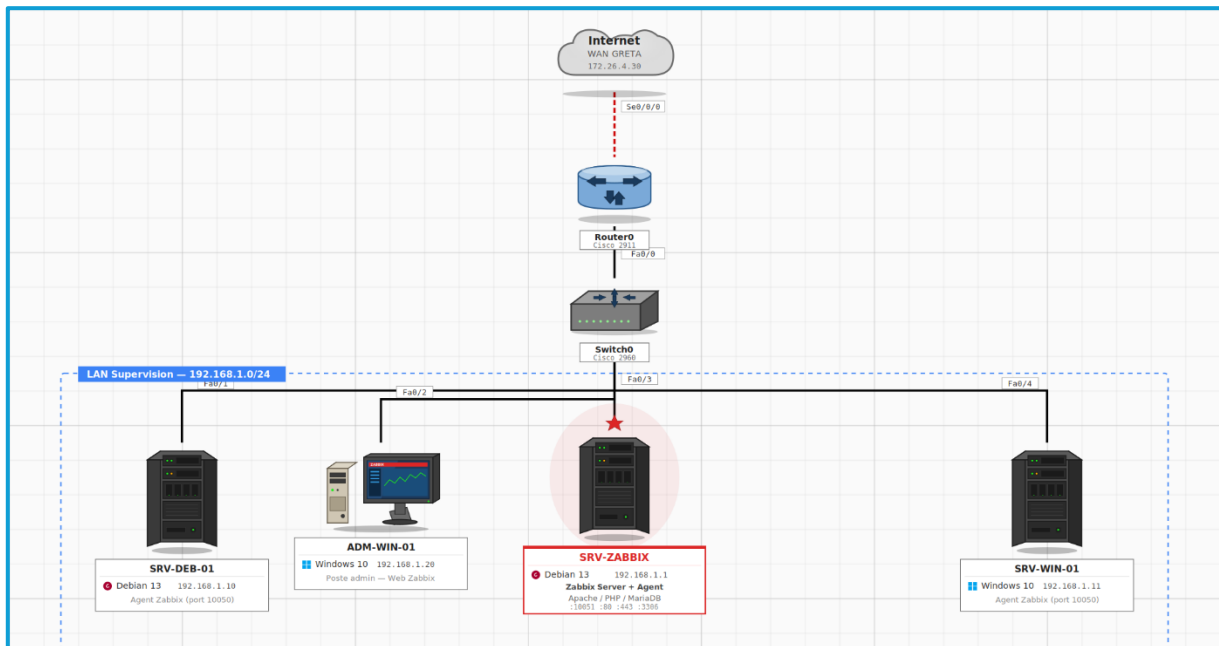
Tableau d'adressage :

Nom du réseau	Adresse réseau	Masque de sous réseau	Première adresse	Dernière adresse	Adresse de broadcast
WAN	172.26.0.0	255.255.248.0	172.26.0.1	172.26.7.254	172.26.7.255
LAN	192.168.1.0	255.255.255.0	192.168.1.1	192.168.1.254	192.168.1.255

Tableau des adresses IP :

Machines	Adresses IP	Masque de sous réseau	Passerelle par défaut	DNS
SRV-ZABBIX (WAN)	172.26.4.30	255.255.248.0	172.26.7.254	1.1.1.1
SRV-NEXTCLOUD (LAN)	192.168.1.1	255.255.255.0	X	X
SRV-DEB-01	192.168.1.10	255.255.255.0	192.168.1.1	X
SRV-WIN-01	192.168.1.11	255.255.255.0	192.168.1.1	X
ADM-WIN-01	192.168.1.20	255.255.255.0	192.168.1.1	X

Schéma réseau :



| Etude de l'impact sur le SI existant |

Étant donné que ce travail s'inscrit dans un contexte de formation, il n'existe aucun système d'information (SI) préexistant, mis à part l'accès à Internet. L'ensemble de l'infrastructure nécessaire au déploiement de Zabbix doit donc être créée et configurée intégralement dans le cadre de ce TP. La mise en place d'un serveur Zabbix dans une entreprise apporterait un gain significatif en matière de visibilité sur l'état du parc informatique, en permettant de détecter les pannes en temps réel, d'anticiper les incidents (saturation disque, pic CPU, baisse de mémoire) et de réduire le temps moyen de résolution grâce à un système d'alertes ciblé.

| Phasage de l'intervention |

Dans un premier temps, j'installerai trois machines virtuelles : Debian 13 pour le serveur Zabbix, Debian 13 pour le client Linux, et Windows 10 pour le client Windows. Une quatrième machine Windows 10 servira de poste d'administration pour accéder à l'interface web Zabbix. Sur le serveur Debian, les services nécessaires à l'utilisation de Zabbix sont : Apache (ou Nginx) pour le serveur web, PHP avec ses extensions, MariaDB pour la base de données, le serveur Zabbix (zabbix-server-mysql), et l'agent Zabbix (zabbix-agent) pour la supervision du serveur lui-même. Une fois l'environnement prêt, je terminerai l'installation via l'assistant web Zabbix puis je configurerai les paramètres généraux (langue, fuseau horaire, mot de passe admin). J'installerai ensuite les agents Zabbix sur les postes clients Linux et Windows et je les enregistrerai dans la base d'hôtes du serveur en leur associant les templates correspondants. Pour finir, je créerai des dashboards personnalisés et je validerai la remontée des alertes en simulant une saturation.

| Prévision des tests et validation |

Lors de l'installation des services nécessaires à Zabbix, je procéderai à différentes vérifications afin de m'assurer de leur bon fonctionnement. Je vérifierai notamment que chaque service (apache2, mariadb, zabbix-server, zabbix-agent) a bien démarré, qu'il écoute sur les ports appropriés (80, 3306, 10050, 10051) et qu'il répond correctement aux requêtes. Par la suite, lors de la mise en place des agents Zabbix sur les postes clients, je contrôlerai que chaque machine remonte correctement dans l'interface web de Zabbix et que sa disponibilité (Availability) passe au vert dans la liste des hôtes. Enfin, une fois les templates appliqués, je testerai la collecte effective des données (CPU, mémoire, disques) en consultant les graphes générés et je validerai le déclenchement d'une alerte en provoquant volontairement une saturation (par exemple, une boucle stressant le CPU sur le client Linux).

Mise en place

| Partitionnement serveur Linux |

Point de montage « swap » 1 Go

Le « swap » sert de mémoire d'échange si la RAM est saturée. Permet d'éviter des crashes ou ralentissements lors d'opérations lourdes (ex. forte charge sur le collecteur Zabbix lorsque le nombre d'items augmente).

Point de montage « / » 5 Go

Contient le système d'exploitation (binaires Linux, services, Apache, PHP, configuration de Zabbix). Sépare le système des données changeantes (/var) et permet d'avoir de meilleures performances, une meilleure stabilité et une maintenance facilitée.

Point de montage « /var » 10 Go

Zabbix stocke ici les logs (zabbix_server.log, zabbix_agentd.log), la base de données MariaDB se trouve dans /var/lib/mysql, et Apache y écrit ses fichiers temporaires. Une partition dédiée empêche ces fichiers variables de remplir la partition système et de bloquer le serveur.

Point de montage « /home » 4 Go

Contient les répertoires utilisateurs. Isolé du système pour éviter qu'un fichier utilisateur volumineux ne sature la partition racine.

| Déploiement de la solution |

Mettre à jour les paquets système :

```
sudo apt update && sudo apt upgrade -y
```

Installer le dépôt officiel Zabbix 7.0 LTS pour Debian 13 :

```
wget https://repo.zabbix.com/zabbix/7.0/debian/pool/main/z/zabbix-release/zabbix-release_latest+debian13_all.deb
```

```
sudo dpkg -i zabbix-release_latest+debian13_all.deb
```

```
sudo apt update
```

Installer le serveur Zabbix, l'interface web, l'agent et les scripts SQL :

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

Installer MariaDB et créer la base de données :

```
sudo apt install mariadb-server -y
```

```
sudo systemctl enable --now mariadb
```

```
sudo mysql_secure_installation
```

Une fois MariaDB sécurisé, on crée la base de données et l'utilisateur Zabbix :

FOUR Lucas

Conseil départemental de la Drôme

BTS SIO OPTION SISR | GRETA ARDECHE-DRÔME | Session 2026

```
sudo mysql -uroot -p
```

```
CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;  
CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'MotDePasseZbx';  
GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';  
SET GLOBAL log_bin_trust_function_creators = 1;  
QUIT;
```

Importer le schéma initial de Zabbix :

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

Désactiver le paramètre temporaire après import :

```
sudo mysql -uroot -p -e "SET GLOBAL log_bin_trust_function_creators = 0;"
```

Configurer le mot de passe de la base dans /etc/zabbix/zabbix_server.conf :

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

```
# Décommenter et renseigner :
```

```
DBPassword=MotDePasseZbx
```

Démarrer et activer les services au boot :

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

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

Vérifier que le serveur Zabbix est bien actif :

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

Une fois les services lancés, l'assistant web est accessible à l'adresse **http://192.168.1.1/zabbix** depuis le poste d'administration. Le navigateur affiche l'écran de bienvenue de l'installateur Zabbix.

ZABBIX Welcome to Zabbix 7.0

1. Welcome
2. Check of pre-requisites
3. **Configure DB connection**
4. Settings
5. Pre-installation summary
6. Install

Configure DB connection

Please create database manually,
and set the configuration parameters for connection to this database.
Press "Next step" button when done.

Database type: MySQL
Database host: localhost
Database port: 0 (0 - use default port)
Database name: zabbix
Store credentials in: Plain text HashiCorp Vault
User: zabbix
Password: *****
Database TLS encryption: Connection will not be encrypted because it uses a socket file

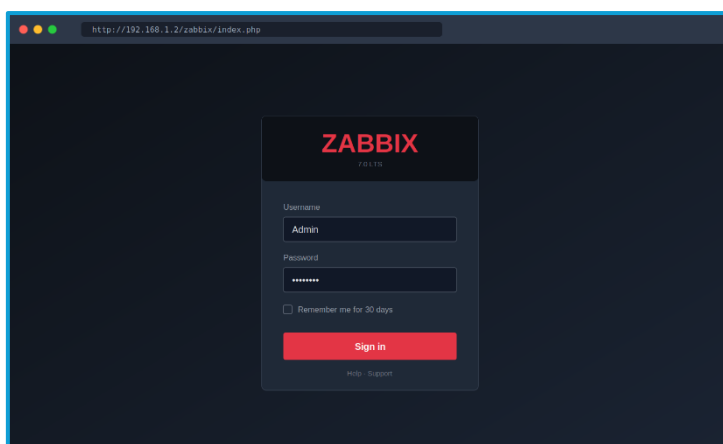
Back Next step

Mise en place d'un serveur Zabbix

Renseignez les informations de connexion à la base de données précédemment créée (type MySQL, hôte localhost, nom de la base zabbix, utilisateur zabbix et mot de passe correspondant), puis cliquez sur « Next step ». L'assistant procédera ensuite aux derniers réglages (nom du serveur Zabbix, fuseau horaire, thème par défaut) avant d'écrire le fichier de configuration final.

Une fois l'installation terminée, on accède à la page de connexion. Les identifiants par défaut sont :

- Identifiant : Admin
- Mot de passe : zabbix



Après connexion, on accède au tableau de bord global, qui présente l'état de santé du serveur Zabbix, le nombre d'hôtes et de problèmes en cours.

Pour vérifier la configuration du serveur :

```
sudo cat /etc/zabbix/zabbix_server.conf | grep -v '^#' | grep -v '^$'
```

| Installation et enregistrement de l'agent Zabbix sur le client Linux |

Sur la machine SRV-DEB-01, ajouter le dépôt Zabbix puis installer l'agent :

```
wget https://repo.zabbix.com/zabbix/7.0/debian/pool/main/z/zabbix-release/zabbix-  
release_latest+debian13_all.deb  
sudo dpkg -i zabbix-release_latest+debian13_all.deb  
sudo apt update  
sudo apt install zabbix-agent -y
```

Configurer l'agent pour pointer vers le serveur Zabbix :

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

```
Server=192.168.1.1  
ServerActive=192.168.1.1  
Hostname=SRV-DEB-01
```

Démarrer et activer l'agent :

```
sudo systemctl restart zabbix-agent  
sudo systemctl enable zabbix-agent  
sudo systemctl status zabbix-agent
```

L'agent doit afficher le statut active (running). Si le pare-feu est actif, autoriser le port 10050 pour la connexion entrante du serveur.

```
sudo ufw allow from 192.168.1.1 to any port 10050
```

| Installation et enregistrement de l'agent Zabbix sur le client Windows |

Sur la machine SRV-WIN-01, télécharger l'agent Zabbix MSI depuis le site officiel :
https://www.zabbix.com/download_agents.

Choisir la version Windows correspondant à l'architecture (64-bit), MSI, version 7.0 LTS. Lancer l'installateur en tant qu'administrateur et renseigner les paramètres suivants lorsqu'ils sont demandés :

- Hostname : SRV-WIN-01
- Zabbix server IP/DNS : 192.168.1.1
- Server or Proxy for active checks : 192.168.1.1
- Listen port : 10050 (port par défaut)
- Cocher « Add agent location to the PATH »

Une fois l'installation terminée, vérifier que le service « Zabbix Agent » est bien démarré dans services.msc, puis ouvrir le port 10050 entrant dans le pare-feu Windows :

```
netsh advfirewall firewall add rule name="Zabbix Agent" dir=in action=allow
```

| Création des hôtes dans l'interface Zabbix |

Depuis l'interface web Zabbix, se rendre dans **Configuration** → **Hosts**, puis cliquer sur **Create host**. Renseigner les informations suivantes pour chaque client :

Type	IP address	DNS name	Connect to	Port
Agent	192.168.1.11		IP DNS	10050

Pour SRV-DEB-01 :

- Host name : SRV-DEB-01
- Templates : Linux by Zabbix agent
- Host groups : Linux servers
- Interfaces : Agent → IP 192.168.1.10 → Port 10050

Pour SRV-WIN-01 :

- Host name : SRV-WIN-01
- Templates : Windows by Zabbix agent
- Host groups : Windows servers
- Interfaces : Agent → IP 192.168.1.11 → Port 10050

Cliquer sur « Add » pour valider. Après quelques secondes, l'icône d'état dans la colonne Availability passe au vert, ce qui confirme que la communication agent-serveur fonctionne correctement.

The screenshot shows the Zabbix web interface in the Configuration section, specifically the Hosts page. At the top, there are navigation tabs: Dashboards, Monitoring, Services, Inventory, Reports, Configuration (selected), and Administration. Below these are sub-tabs: Host groups, Hosts (selected), Templates, Maintenance, Actions, and Discovery. A 'Create host' button is visible in the top right corner.

A filter section allows searching by Name, Host groups, and Templates, with 'Apply' and 'Reset' buttons. Below the filter is a table listing hosts:

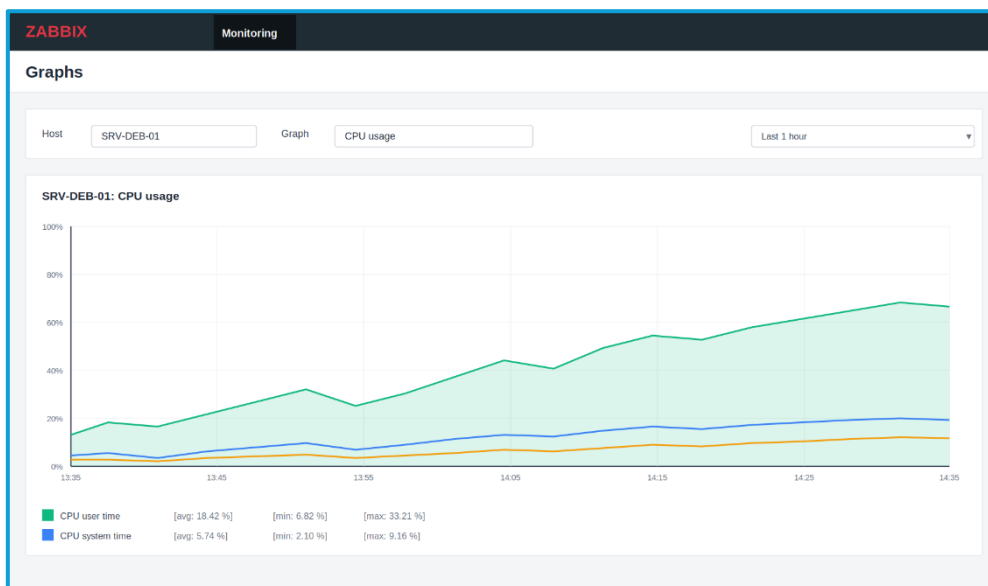
Name	Interface	Host group	Templates	Status	Availability	Info
Zabbix server	127.0.0.1:10050	Zabbix servers	Linux by Zabbix agent	Enabled	ZBX	
SRV-DEB-01	192.168.1.10:10050	Linux servers	Linux by Zabbix agent	Enabled	ZBX	
SRV-WIN-01	192.168.1.11:10050	Windows servers	Windows by Zabbix agent	Enabled	ZBX	

At the bottom left, it says 'Displaying 3 of 3 found'.

| Vérification de la collecte des données |

Pour vérifier que les données sont bien collectées, aller dans **Monitoring** → **Hosts** → **Latest data**. On doit voir apparaître les valeurs en temps réel des items remontés par l'agent (charge CPU, utilisation mémoire, espace disque, trafic réseau, etc.).

Les graphes sont accessibles via **Monitoring** → **Hosts** → **Graphs**. On peut sélectionner le client souhaité, le graphe à afficher et la plage temporelle (1h, 24h, 7 jours, etc.).



| Test du déclenchement d'une alerte |

Pour valider la chaîne complète de supervision, on simule une saturation CPU sur le client Linux à l'aide de la commande stress :

```
sudo apt install stress -y  
stress --cpu 2 --timeout 600
```

Au bout de quelques minutes, un trigger de niveau Average ou High se déclenche dans l'interface Zabbix sous Monitoring → Problems. Si une action a été configurée, un mail d'alerte est envoyé. Cela confirme que les agents, le serveur, les triggers et les actions fonctionnent correctement de bout en bout.

Conclusion

La mise en place du serveur Zabbix, accompagnée de l'installation et de la configuration de ses différents services, m'a permis de mettre en œuvre une solution complète et efficace de supervision de parc informatique. Les tests d'enregistrement d'hôtes Linux et Windows, la collecte des indicateurs en temps réel et la validation du déclenchement d'alertes ont confirmé la fiabilité et la performance de l'infrastructure. Grâce à Zabbix, il devient possible d'avoir une vision centralisée et instantanée de l'état du système d'information, d'anticiper les incidents et de réagir rapidement en cas de problème, tout en assurant une gestion homogène des environnements Linux et Windows.

Auto-évaluation

Le temps imparti de seize heures a été respecté, grâce à une bonne préparation en amont, notamment le téléchargement anticipé des ISO et des paquets, ainsi que la connaissance préalable de la procédure d'installation. Cette anticipation m'a permis d'optimiser le déroulement du TP et d'atteindre les objectifs fixés dans les délais. Une amélioration possible serait d'aller plus loin sur la partie alerting (configuration des notifications mail et des escalades), ainsi que sur la création de dashboards personnalisés adaptés à un contexte d'entreprise réel.