

# I. Installation serveur Zabbix 7.4 sur Debian 13 trixie

Guide tiré de la documentation officielle :

[Download and install Zabbix 7.4 for Debian 13 Trixie](#) (amd64, arm64)

Cette dernière est très bien faite car en fonction de ce qu'on sélectionne, elle nous donne les commandes adaptées.

## I. Choose your platform / Setup

```
apt update && apt upgrade
```

```
Apt install sudo
```

```
Apt install mariadb-server
```

```
wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release\_latest\_7.4+debian13\_all.deb
```

```
dpkg -i zabbix-release_latest_7.4+debian13_all.deb
```

```
apt update
```

## II. Install and configure Zabbix for your platform

### a. Install Zabbix repository

Documentation

```
wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release\_latest\_7.4+debian13\_all.deb
```

```
dpkg -i zabbix-release_latest_7.4+debian13_all.deb
```

```
apt update
```

### b. Install Zabbix server, frontend, agent

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

### c. Create initial database

Documentation

Make sure you have database server up and running.

Run the following on your database host.

```
mysql -uroot -p
```

➤ **Entrer le mot de passe désiré**

```
mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
```

**attention, avant d'entrer la prochaine commande, changer le « password » !!!**

```
mysql> create user zabbix@localhost identified by 'password';
```

```
mysql> grant all privileges on zabbix.* to zabbix@localhost;
```

```
mysql> set global log_bin_trust_function_creators = 1;
```

```
mysql> quit;
```

**On Zabbix server host import initial schema and data. You will be prompted to enter your newly created password.**

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

**Disable log\_bin\_trust\_function\_creators option after importing database schema.**

```
mysql -uroot -p
```

```
password
```

```
mysql> set global log_bin_trust_function_creators = 0;
```

```
mysql> quit;
```

#### **d. Configure the database for Zabbix server**

Edit file /etc/zabbix/zabbix\_server.conf

```
DBPassword=password
```

#### **e. Start Zabbix server and agent processes**

Start Zabbix server and agent processes and make it start at system boot.

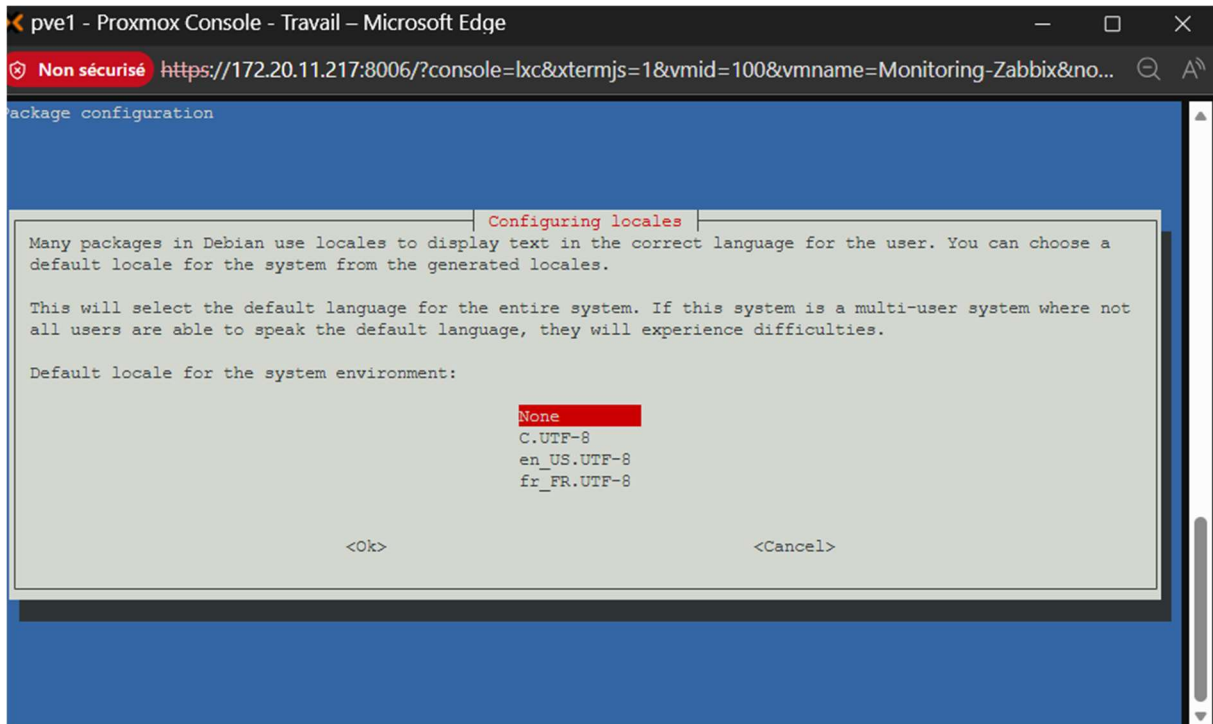
```
systemctl restart zabbix-server zabbix-agent apache2
```

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

#### **f. Install locales**

Installer les langues nécessaires grâce à « sudo dpkg-reconfigure locales »

Ici, nous choisirons « **en\_US.UTF-8** » :



## g. web-configuration

Puis, se rendre sur l'interface web de zabbix :

172.20.11.210 :8081/zabbix/setup.php

ID par défaut : Admin (avec le A majuscule)

Mot de passe par défaut : zabbix

Choisir US :

Welcome to

# Zabbix 7.4

Default language  

Next step > Next step

**ZABBIX**

Welcome

Check of pre-requisites

Configure DB connection

Settings

Pre-installation summary

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:

Database host:

Database port:  0 - use default port

Database name:

Store credentials in:  Plain text  HashiCorp Vault  CyberArk Vault

User:

Password:

Database TLS encryption: *Connection will not be encrypted because it uses a socket file (on Unix) or shared memory (Windows).*

Laisser MySQL, localhost, 0, zabbix sur le nom de la db et l'utilisateur, et en password mettre celui désiré.

A la prochaine étape, mettre un nom de serveur quelconque (par ex « zabbix-groupe2 »

#### h. Fichier de conf zabbix

Il faut désormais indiquer le mot de passe choisi à l'étape c pour l'utilisateur **zabbix@localhost** dans le fichier de conf de zabbix, pour qu'il puisse accéder à la base de données.

Pour cela :

**sudo nano /etc/zabbix/zabbix\_server.conf**

Rechercher via **Ctrl+F « DBPassword= »**

enlever le « # » puis indiquer le mot de passe.

**Enregistrer et quitter.**

Bravo, z'avez terminé, vous pouvez passer aux agents !

## II. Installation des agents (debian 13)

L'agent sera installé sur les machines souhaitées, et peut même l'être sur le serveur lui-même.

### a. Installer le repository

wget [https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release\\_latest\\_7.4+debian13\\_all.deb](https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_latest_7.4+debian13_all.deb)

```
dpkg -i zabbix-release_latest_7.4+debian13_all.deb
```

```
apt update
```

#### **b. Installer l'agent**

```
apt install zabbix-agent2
```

#### **NB : il existe 2 agents, agent 1 et agent 2.**

Agent 1 : codé en C, consomme moins de RAM, plus ancien.

Agent 2 : codé en Go, plus moderne et efficace.

#### **c. Install Zabbix agent 2 plugins**

```
apt install zabbix-agent2-plugin-mongodb zabbix-agent2-plugin-mssql zabbix-agent2-plugin-postgresql
```

#### **d. Start Zabbix agent 2 process**

```
systemctl restart zabbix-agent2
```

```
systemctl enable zabbix-agent2
```

```
systemctl status zabbix-agent2
```

## III. Configuration des agents (debian 13)

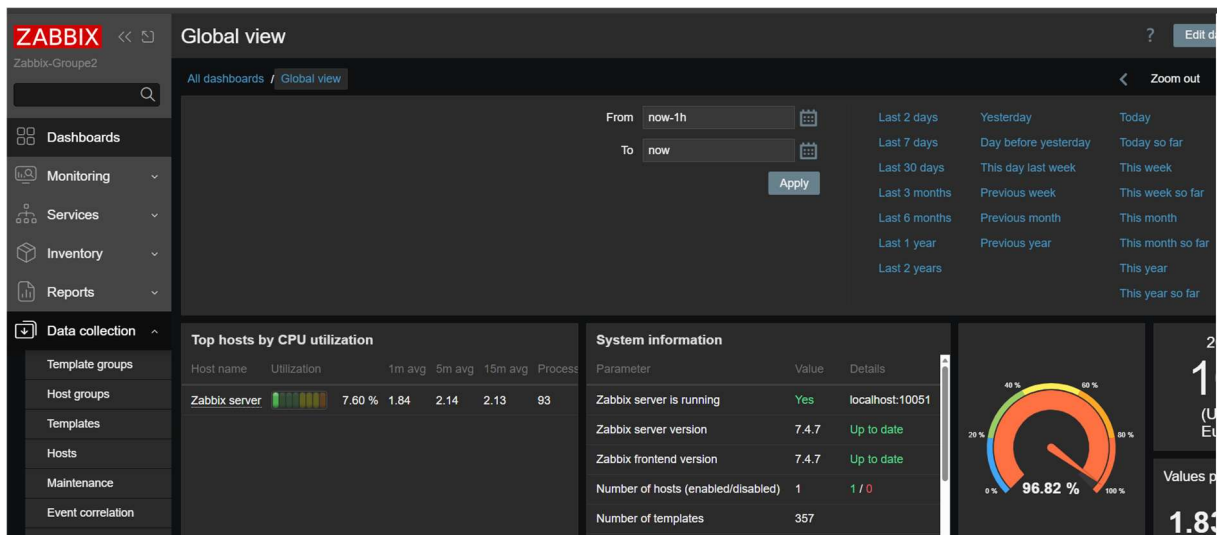
#### **a. Configuration du fichier de conf de l'agent**

```
sudo nano /etc/zabbix/zabbix_agent2.conf
```

Rechercher via **Ctrl+F** : « **SERVER=** »

enlever le « # » puis indiquer l'ip du serveur zabbix.

**Enregistrer et quitter.**

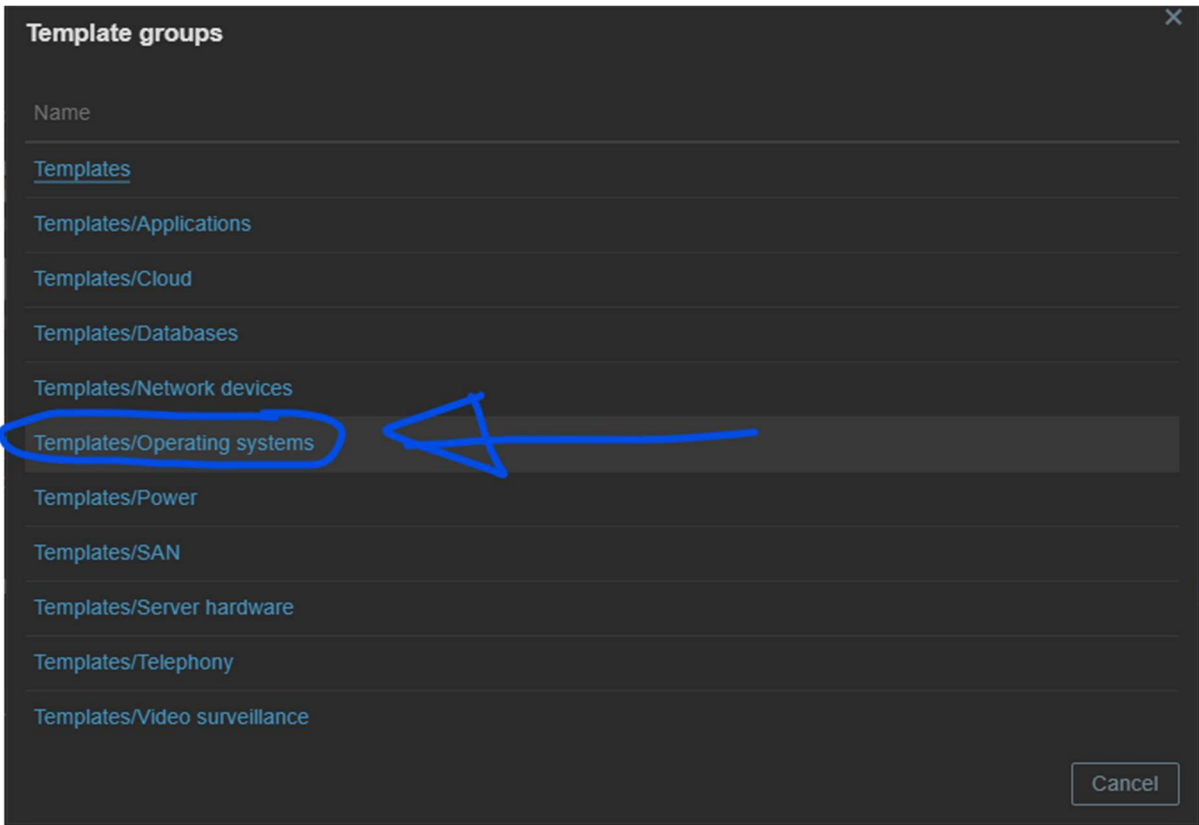


## b. Déclaration de(s) l'hôte(s) sur la plateforme web du serveur zabbix

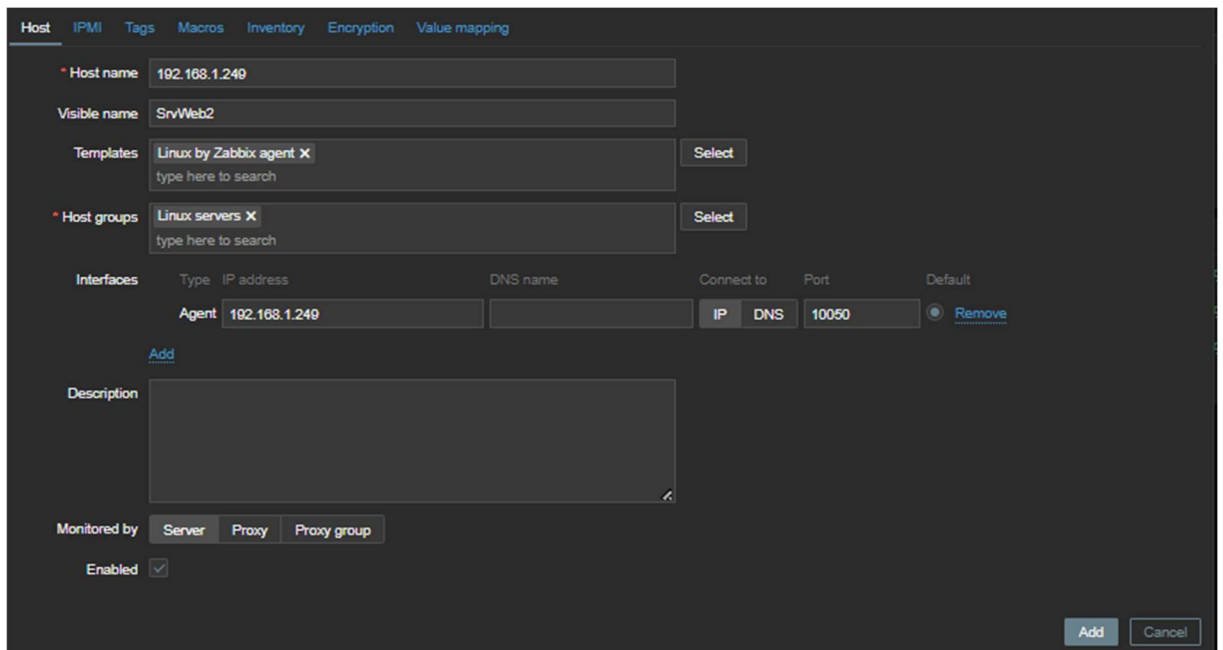
Une fois sur la plateforme web de Zabbix, aller dans « **Data Collection** » puis « **Hosts** » et enfin « **Create host** » qui va nous permettre d'indiquer les hôtes qu'on souhaite surveiller grâce à notre serveur Zabbix.

1. **Host name** : indiquer l'IP de la machine à surveiller
2. **Visible Name** : Lui donner un p'tit nom friendly
3. **Templates** :  
Cliquez sur **Select** :

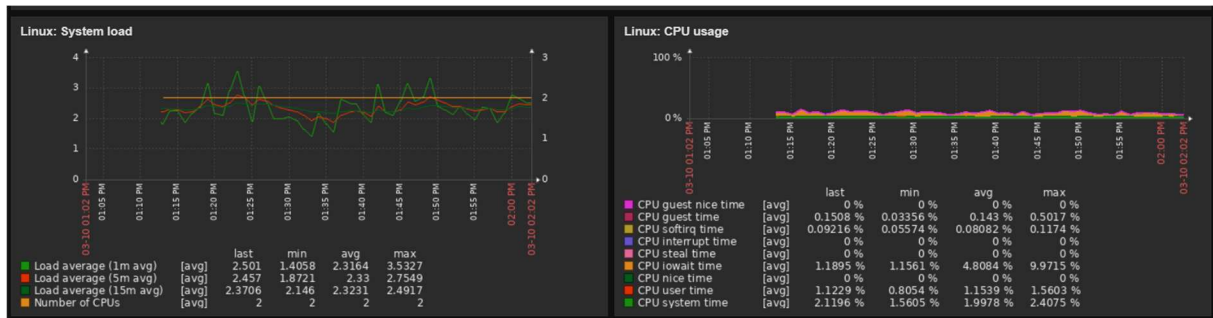
4. **Select** à nouveau
5. Cliquez sur **Templates-Operating Systems**



6. Puis cocher « **Linux by Zabbix Agent** » et enfin **select**
7. **Host Groups** : mettre « Linux Servers »
8. **Host groups** : à vous de voir, pour un SW j'ai choisi **Linux Servers**.
9. **Interfaces** : add > **Agent**. Indiquer l'IP de la machine à surveiller, laisser cocher « IP » et le port d'écoute de zabbix « **10050** »
10. **Monitored by** : Server.



Bravo ! Vous devriez pouvoir accéder au **Dashboard** de votre machine



## IV. Install et config des agents (si debian 12)

### I. Installation des agents

L'agent sera installé sur les machines souhaitées, et peut même l'être sur le serveur lui-même.

#### a. Installer le repository

```
wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_latest_7.4+debian12_all.deb
```

```
dpkg -i zabbix-release_latest_7.4+debian12_all.deb
```

```
apt update
```

#### b. Installer l'agent

```
apt install zabbix-agent2
```

**NB** : il existe 2 agents, agent 1 et agent 2. Agent 1 : codé en C, consomme moins de RAM, plus ancien. Agent 2 : codé en Go, plus moderne et efficace.

#### c. Install Zabbix agent 2 plugins

```
apt install zabbix-agent2-plugin-mongodb zabbix-agent2-plugin-mssql zabbix-agent2-plugin-postgresql
```

#### d. Start Zabbix agent 2 process

```
systemctl restart zabbix-agent2
```

```
systemctl enable zabbix-agent2
```

```
systemctl status zabbix-agent2
```

## II. Configuration des agents

### a. Configuration du fichier de conf de l'agent

```
sudo nano /etc/zabbix/zabbix_agent2.conf
```

Rechercher via Ctrl+W : Server= , enlever le # puis indiquer l'IP du serveur Zabbix.  
Enregistrer et quitter.

### **b. Déclaration de(s) l'hôte(s) sur la plateforme web du serveur Zabbix**

Une fois sur la plateforme web de Zabbix, aller dans Data Collection puis Hosts et enfin Create host.

**Host name** : indiquer l'IP de la machine à surveiller

**Visible Name** : lui donner un p'tit nom friendly

**Templates** : cliquer sur Select

Select à nouveau

Cliquer sur **Templates-Operating Systems**

Puis cocher **Linux by Zabbix Agent** et enfin Select

**Host Groups** : mettre **Linux Servers**

**Interfaces** : Add > Agent. Indiquer l'**IP** de la machine à surveiller, laisser cocher IP et le port d'écoute de Zabbix **10050**

**Monitored by** : Server

Bravo ! Vous devriez pouvoir accéder au Dashboard de votre machine.

## **V. Installation des agents (Mikrotik)**

Depuis le terminal Mikrotik de Winbox :

### **1. Activer le SNMP (Simple Network Mail Protocol)**

```
/snmp set enabled=yes
```

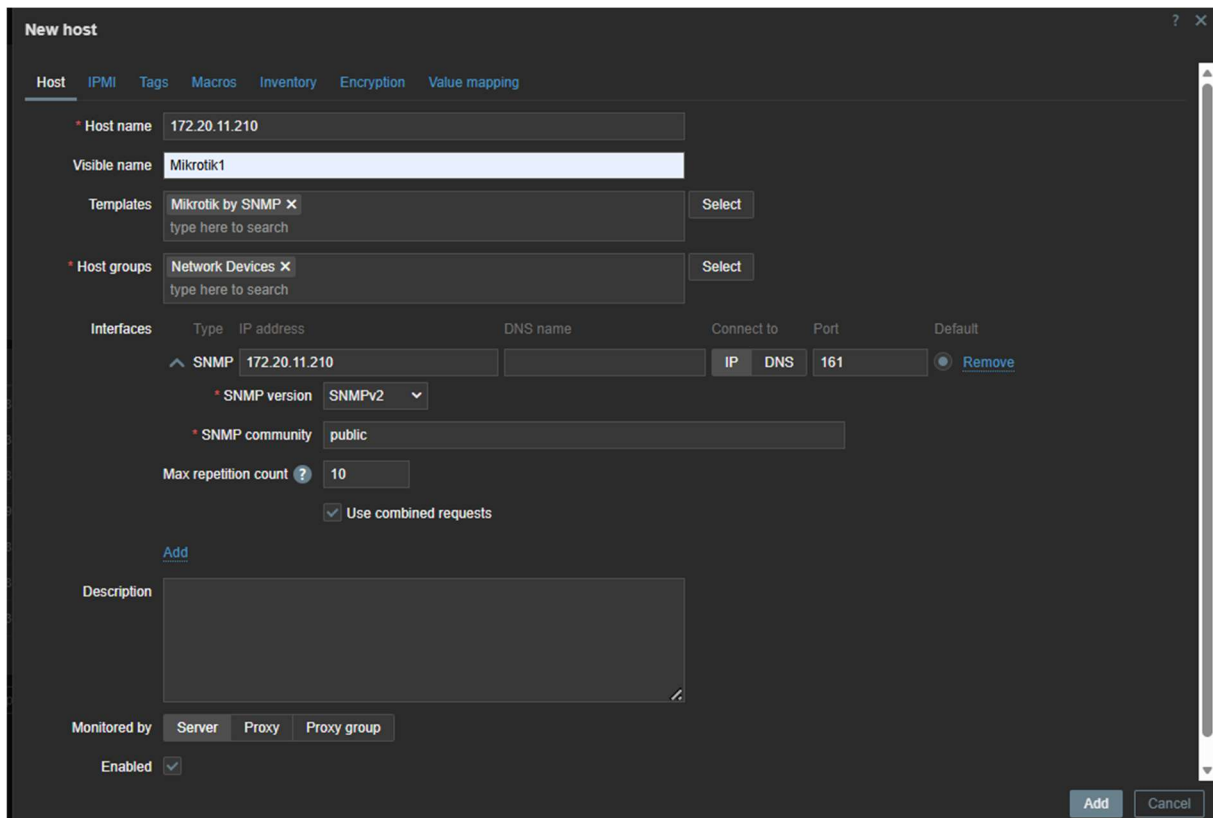
### **2. Configurer la community**

```
/snmp community set 0 name=public
```

### **3. Vérifier qu'il est actif**

```
/snmp print
```

## Puis créer l'host sur Zabbix



The screenshot shows the 'New host' configuration window in Zabbix. The form is filled with the following information:

- Host name:** 172.20.11.210
- Visible name:** Mikrotik1
- Templates:** Mikrotik by SNMP
- Host groups:** Network Devices
- Interfaces:** A table with one entry: Type: SNMP, IP address: 172.20.11.210, DNS name: (empty), Connect to: IP, Port: DNS, 161, Default: Remove.
- SNMP version:** SNMPv2
- SNMP community:** public
- Max repetition count:** 10
- Use combined requests:** checked
- Description:** (empty text area)
- Monitored by:** Server, Proxy, Proxy group
- Enabled:** checked

**Host Name :** l'IP du routeur

**Template :** Templates/Network Devices

**Host Groups :** Choisir un de ceux existant ou alors un personnel précédemment créé comme Network Devices

**Interfaces :** SNMP : l'IP du routeur

**SNMP community :** indiquer le nom créé sur le routeur

## VI. ERREURS POSSIBLES



Interface	Status	Error
192.168.2.250:10	Not available	Get value from agent failed: cannot read response: cannot read from socket: [104] 050 Connection reset by peer

**Vérifier si agent démarré :** `systemctl status zabbix-agent2`

Si pas actif : `systemctl restart zabbix-agent2`

Vérifier si **l'IP du serveur dans fichier de conf** est la bonne :

`nano /etc/zabbix/zabbix_agent2.conf`

Sinon, cela veut dire que le **port 10050 n'est pas ouvert** sur le parefeu de la VM :

```
apt install ufw
```

```
ufw allow 10050/tcp
```

Zabbix devrait maintenant remonter.

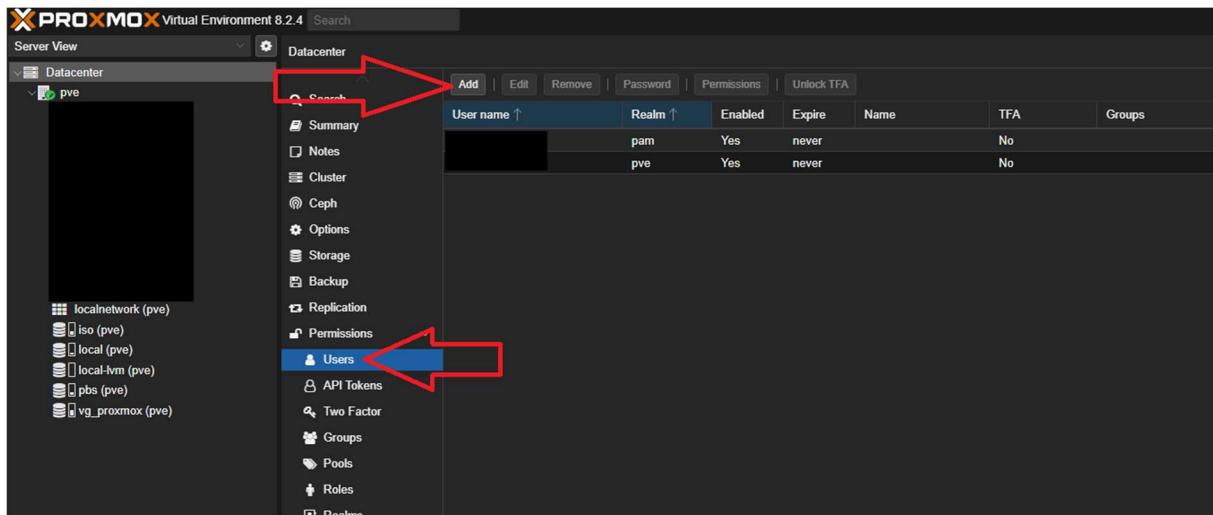
## Bonus : Installer le modèle Zabbix « Proxmox VE by http »

En plus des agents sur les containers et VMs, ceci permet de surveiller de façon plus globale le cluster.

### Configuration de Proxmox

Pour sécuriser la connexion, nous allons créer un jeton API pour un utilisateur dédié avec des droits limités au datacenter.

Nous allons créer un utilisateur dédié en accédant à Datacenter > Permissions > Users > Add.



Pour la démonstration, je nomme cet utilisateur « sup\_zabbix ».

**Add: User** ✕

User name:  First Name:

Realm:  Last Name:

Group:  E-Mail:

Expire:

Enabled:

Comment:

Advanced  **Add**

Une fois l'utilisateur créé, cliquez sur Permissions > Add > User Permission.

**Datacenter**

Search

Summary

Notes

Cluster

Ceph

Options

Storage

Backup

Replication

**Permissions**

- User
- API Tokens
- Two factor
- Groups
- Pools
- Roles
- Realms

HA

SDN

Zones

VNets

**Add** **Remove**

Group Permission	User/Group/API Token	Role
<input checked="" type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		

Ajoutez l'utilisateur au groupe PVEAuditor.

**Add: User Permission** ✕

Path:

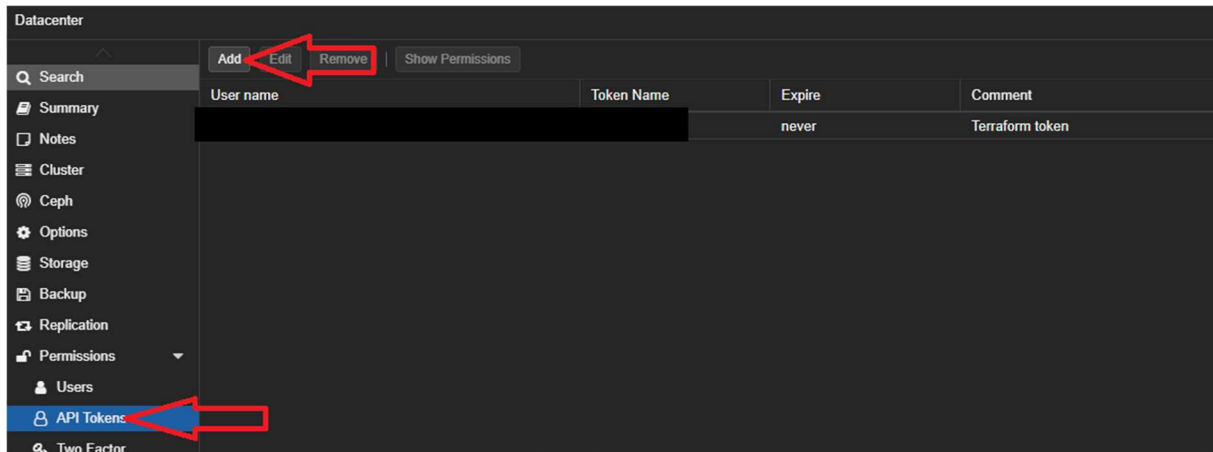
User:

Role:

Propagate:

**Help** **Add**

Créez une clé API pour cet utilisateur.



Saisissez votre utilisateur et attribuez un nom personnalisé au jeton.

**Add: Token**

User:  Privilege Separation:

Token ID:  Expire:

Comment:

Ensuite, le jeton apparaîtra une seule fois. Récupérez-le et stockez-le dans un endroit sécurisé. **Pour notre AP4 : 1ab9436b-92c0-4dc6-9abc-9e9de3bf3d48**

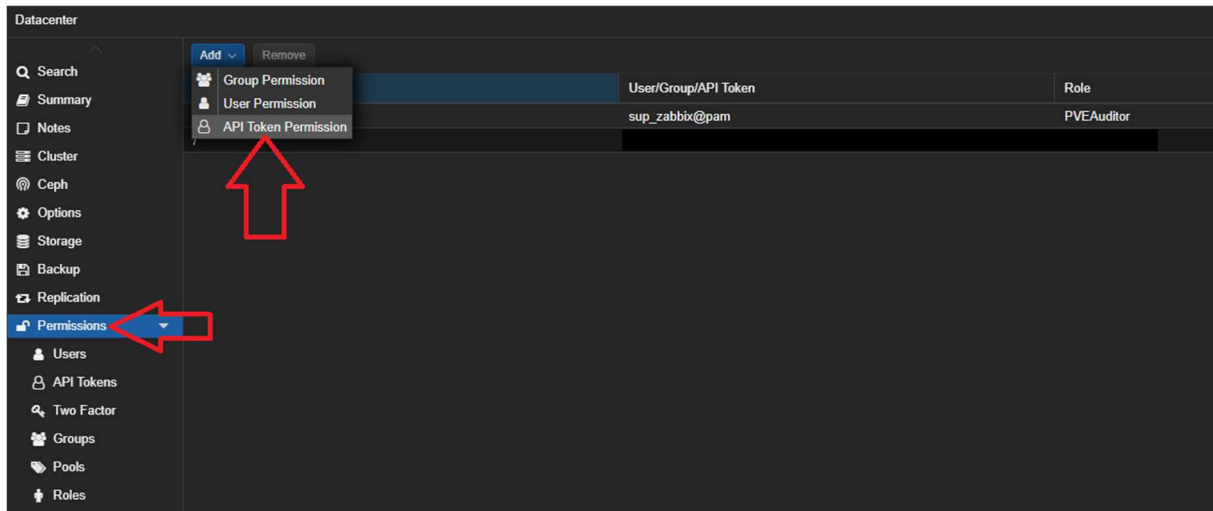
**Token Secret**

Token ID:

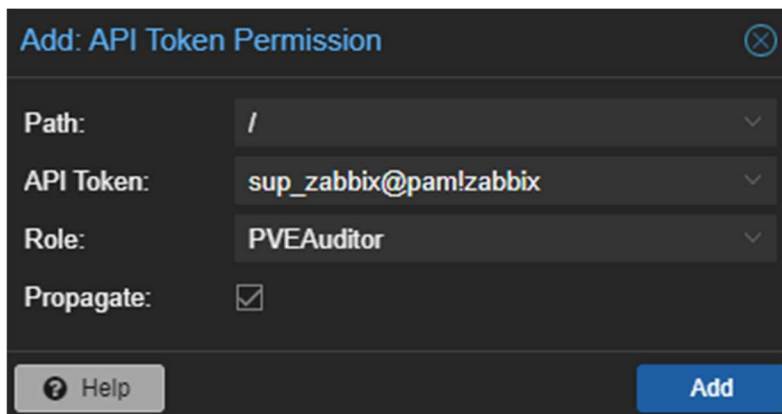
Secret:

Please record the API token secret - it will only be displayed now

Ajoutez maintenant les autorisations au jeton.



Le rôle sera le même que l'utilisateur, c'est-à-dire « PVEAuditor ».



Tout est prêt du côté de Proxmox.

### Configuration sur Zabbix

Il existe déjà un template « Proxmox VE by HTTP » pour Proxmox sur Zabbix, et c'est celui que nous allons utiliser.

Nous allons éditer les macros pour qu'elles se connectent à notre hôte Proxmox.

Pour cela, recherchez le template Proxmox dans Collecte de données > Modèles > Proxmox VE by HTTP.

The screenshot shows the Zabbix interface for managing templates. The left sidebar contains a navigation menu with 'Modèles' selected. The main content area displays a form for creating a new template. The 'Nom' field is filled with 'proxmox'. Below the form, a table lists existing templates, with 'Proxmox VE by HTTP' highlighted in red. The table columns include 'Nom', 'Hôtes', 'Éléments', 'Déclencheurs', 'Graphiques', 'Tableaux de bord', 'Découverte', 'Web', 'Fabricant', and 'Version'.

Dans l'onglet Macros, modifiez les trois valeurs nécessaires.

The screenshot shows the 'Macros' tab for a Zabbix template. The table lists macros with their values and descriptions. Three macros are highlighted in red:

Macro	Valeur	Description
{PVE.CPU.PUSE.MAX.WARN}	90	Maximum used CPU in percentage.
{PVE.LXC.CPU.PUSE.MAX.WARN}	90	Maximum used CPU in percentage.
{PVE.LXC.MEMORY.PUSE.MAX.WARN}	90	Maximum used memory in percentage.
{PVE.MEMORY.PUSE.MAX.WARN}	90	Maximum used memory in percentage.
{PVE.ROOT.PUSE.MAX.WARN}	90	Maximum used root space in percentage.
{PVE.STORAGE.PUSE.MAX.WARN}	90	Maximum used storage space in percentage.
{PVE.SWAP.PUSE.MAX.WARN}	90	Maximum used swap space in percentage.
{PVE.TOKEN.ID}	sup_zabbix@pam!zabbix	API tokens allow stateless access to most parts of the REST API by another system, software or API client.
{PVE.TOKEN.SECRET}	[redacted]	Secret key.
{PVE.URL.HOST}	192.168.[redacted]	The hostname or IP address of the Proxmox VE API host.
{PVE.URL.PORT}	8006	The API uses the HTTPS protocol and the server listens to port 8006 by default.
{PVE.VM.CPU.PUSE.MAX.WARN}	90	Maximum used CPU in percentage.
{PVE.VM.MEMORY.PUSE.MAX.WARN}	90	Maximum used memory in percentage.

Buttons at the bottom include: Actualiser, Clone, Supprimer, Supprimer et effacer, Annuler.

**Ici, VIGILANCE : c'est sup\_zabbix@pam!zabbix, avec un point d'exclamation !!!**

Ajout de l'hôte dans Zabbix dans Collecte de données > Hôtes > Créer un hôte

The screenshot shows the Zabbix interface for managing hosts. The left sidebar contains a navigation menu with 'Hôtes' selected. The main content area has a form for creating a new host with fields for name, IP, and various discovery triggers. A red arrow points to the 'Créer un hôte' button in the top right corner.

Ajoutez l'hôte dans Zabbix en accédant à Collecte de données > Hôtes > Créer un hôte.

The screenshot shows the 'Nouvel hôte' form. The 'Nom de l'hôte' field contains 'Proxmox'. The 'Nom visible' field also contains 'Proxmox'. The 'Modèles' dropdown is set to 'Proxmox VE by HTTP'. The 'Groupes d'hôtes' dropdown is set to 'Proxmox'. The 'Interfaces' section shows 'Aucune interface n'est définie'. The 'Description' field is empty. The 'Surveillé par' section has 'Serveur' selected. The 'Activé' checkbox is checked. 'Ajouter' and 'Annuler' buttons are at the bottom right.

Une fois notre hôte créé, nous pouvons vérifier que les valeurs de celui-ci ne présentent pas d'erreurs.

Item	Unit	Step	Interval	Time	Script	Status	Component		
Proxmox VE by HTTP: API service status	Déclencheurs	1	proxmox.api.available	5m	7d	365d	Script	Activé	component: system
Proxmox VE by HTTP: Get cluster resources			proxmox.cluster.resources	5m	0		Agent HTTP	Activé	component: raw
Proxmox VE by HTTP: Get cluster status			proxmox.cluster.status	5m	0		Agent HTTP	Activé	component: raw
Node discovery: Node [pve]: RRD statistics: Node [pve]: CPU, iowait			proxmox.node.iowait[pve]	7d		365d	Élément dépendant	Activé	component: cpu node: pve
Node discovery: Node [pve]: RRD statistics: Node [pve]: CPU, loadavg			proxmox.node.loadavg[pve]	7d		365d	Élément dépendant	Activé	component: cpu node: pve
Node discovery: Node [pve]: RRD statistics: Node [pve]: CPU, usage	Déclencheurs	1	proxmox.node.cpu[pve]	7d		365d	Élément dépendant	Activé	component: cpu node: pve
Node discovery: Node [pve]: RRD statistics: Node [pve]: Incoming data, rate			proxmox.node.netin[pve]	7d		365d	Élément dépendant	Activé	component: network node: pve
Node discovery: Node [pve]: Status: Node [pve]: Kernel version	Déclencheurs	1	proxmox.node.kernelversion[pve]	7d			Élément dépendant	Activé	component: system node: pve
Node discovery: Node [pve]: Time: Node [pve]: Localtime			proxmox.node.localtime[pve]	7d		365d	Élément dépendant	Activé	component: system node: pve
Node discovery: Node [pve]: RRD statistics: Node [pve]: Memory, total	Déclencheurs	1	proxmox.node.memtotal[pve]	7d		365d	Élément dépendant	Activé	component: memory node: pve
Node discovery: Node [pve]: RRD statistics: Node [pve]: Memory, used	Déclencheurs	1	proxmox.node.memused[pve]	7d		365d	Élément dépendant	Activé	component: memory node: pve
Node discovery: Node [pve]: RRD statistics: Node [pve]: Outgoing data, rate			proxmox.node.netout[pve]	7d		365d	Élément dépendant	Activé	component: network node: pve
Node discovery: Node [pve]: Status: Node [pve]: PVE version	Déclencheurs	1	proxmox.node.pveversion[pve]	7d			Élément dépendant	Activé	component: system node: pve
Node discovery: Node [pve]: RRD statistics: Node [pve]: Root filesystem, total	Déclencheurs	1	proxmox.node.roottotal[pve]	7d		365d	Élément dépendant	Activé	component: storage node: pve
Node discovery: Node [pve]: RRD statistics: Node [pve]: Root filesystem, used	Déclencheurs	1	proxmox.node.rootused[pve]	7d		365d	Élément dépendant	Activé	component: storage node: pve
Node discovery: Node [pve]: RRD statistics			proxmox.node.rrd[pve]	1m	0		Agent HTTP	Activé	component: raw node: pve

Et dans l'onglet « Dernières données », on constate la remontée correcte des métriques.

<input type="checkbox"/> Proxmox	API service status <a href="#">🔗</a>	9m 22s	OK (200)	component: system	Graphique
<input type="checkbox"/> Proxmox	Get cluster resources <a href="#">🔗</a>			component: raw	
<input type="checkbox"/> Proxmox	Get cluster status <a href="#">🔗</a>			component: raw	
<input type="checkbox"/> Proxmox	Node [pve]: CPU_iowait <a href="#">🔗</a>	27s	0.08384 %	-0.007397 %	component: cpu node: pve Graphique
<input type="checkbox"/> Proxmox	Node [pve]: CPU_loadavg <a href="#">🔗</a>	27s	0.2377	+0.0003333	component: cpu node: pve Graphique
<input type="checkbox"/> Proxmox	Node [pve]: CPU_usage <a href="#">🔗</a>	27s	6.9835 %	+4.0512 %	component: cpu node: pve Graphique
<input type="checkbox"/> Proxmox	Node [pve]: Incoming_data_rate <a href="#">🔗</a>	27s	146.22 Kbps	+101.57 Kbps	component: network node: pve Graphique
<input type="checkbox"/> Proxmox	Node [pve]: Kernel version <a href="#">🔗</a>	1m 15s	Linux 6.8.8-2-pve #1 SM...		component: system node: pve Historique
<input type="checkbox"/> Proxmox	Node [pve]: Localtime <a href="#">🔗</a>	11s	19/07/2024 00:41:52	+00:01:00	component: system node: pve Graphique
<input type="checkbox"/> Proxmox	Node [pve]: Memory_total <a href="#">🔗</a>	1m 27s	31.01 GB		component: memory node: pve Graphique
<input type="checkbox"/> Proxmox	Node [pve]: Memory_used <a href="#">🔗</a>	27s	26.97 GB	-84.52 MB	component: memory node: pve Graphique
<input type="checkbox"/> Proxmox	Node [pve]: Outgoing_data_rate <a href="#">🔗</a>	27s	1.54 Mbps	+1.42 Mbps	component: network node: pve Graphique

## Conclusion

Grâce à l'intégration de Zabbix, nous sommes désormais en mesure de surveiller Proxmox de manière proactive, d'anticiper les problèmes, de réagir rapidement aux alertes et de maintenir une infrastructure performante et sécurisée.