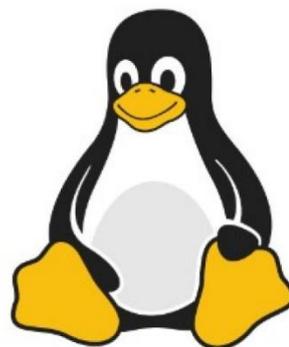
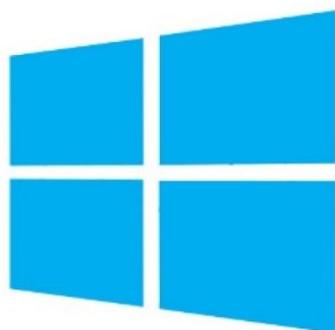


PR06 : SIO SISR - Mise en place d'un outil de supervision réseau sous Linux

- Zabbix -



Sommaires

Contexte :.....	3
Objectifs :.....	3
Cahier des charges :	3
Solution :.....	3
Schéma ASI :.....	4
Prérequis :	4
Installation et Configuration du serveur Zabbix :.....	4
Installation du serveur :	4
Configuration depuis l'interface Web :	7
Configuration de l'agent Zabbix sur un Serveurs Windows	11
Configuration de l'agent Zabbix sur le serveur Web (Debian11)	15
Configuration des interfaces réseaux :.....	17
Création des Hôtes :	19
Création de l'hôte Windows :	20
Création de l'hôte Debian :	20
Vérification des alertes :.....	21
Vérification et configuration des alertes :	21
Test de remonter des alertes :.....	24
Configuration du Dashboard :.....	26
Conclusion :.....	27

Contexte :

Une entreprise nous a sollicités afin de mettre en place un système de supervision pour ses serveurs Windows et Linux.

Objectifs :

L'objectif est de surveiller les performances des ressources critiques, notamment l'utilisation du processeur (CPU) et de la mémoire vive (RAM), afin d'assurer la disponibilité et la stabilité de ses services et applications hébergés.

Cahier des charges :

Mise en Place d'un Système de Supervision :

- Procéder à l'installation et à la configuration d'un logiciel de supervision, en détaillant chaque étape de la procédure selon l'environnement choisi (Windows ou Linux).
- Effectuer l'installation des agents nécessaires sur les postes client et vérifier que ces derniers sont correctement détectés et répertoriés dans l'interface du logiciel de supervision.

Gestion des Règles et des Alertes :

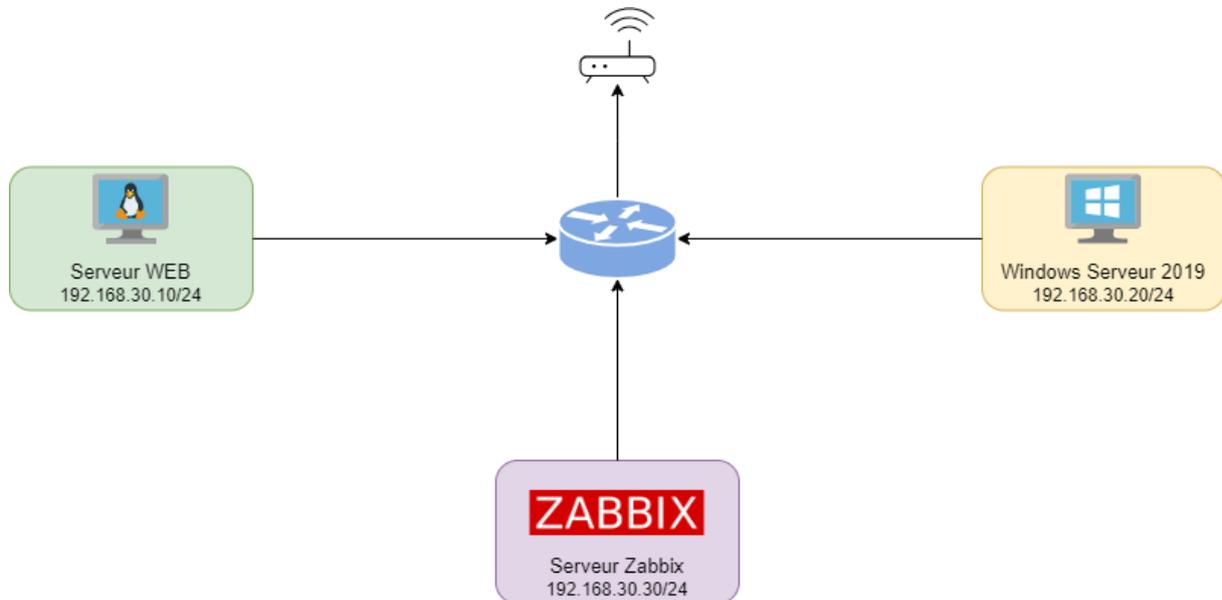
- Configurer des alertes (triggers) afin de tester la bonne réception des données et de valider le bon fonctionnement de la supervision des services.
- Définir des règles de supervision spécifiques aux ressources matérielles, telles que la charge du processeur (CPU) et l'état de la mémoire vive (RAM).
- Centraliser et visualiser les alertes sur un tableau de bord dédié pour un suivi optimal.

Solution :

Pour répondre au cahier des charges, je vais mettre en place une machine virtuelle (VM) sous Debian11 afin d'héberger mon serveur Zabbix. De plus, j'installerai un agent Zabbix sur un serveur web sous Debian 11, ainsi qu'un agent Zabbix sur un serveur Windows 2019.

Pour le serveur Windows 2019 nous utiliseront OCCT afin d'effectuer des tests.

Schéma ASI :



Prérequis :

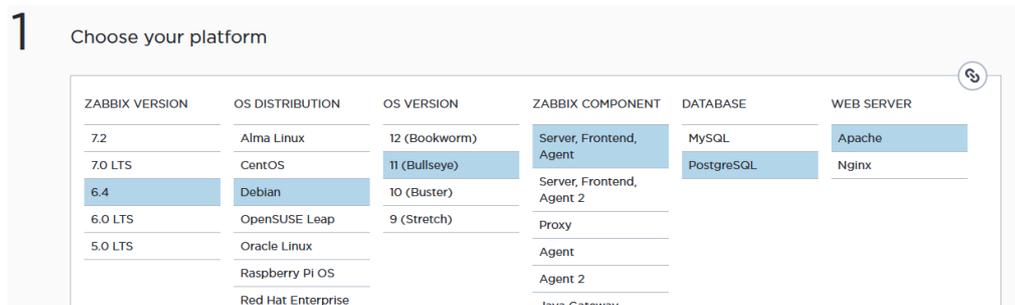
Tout d'abord, nous allons installer et paramétrer une machine virtuelle sous Debian, qui servira de serveur Zabbix. Sur cette machine virtuelle sera présent un serveur LAMP (Linux, Apache, MariaDB et PHP) afin de pouvoir faire fonctionner Zabbix.

Ensuite, nous configurerons les interfaces réseau du serveur web sous Debian 11 et du serveur Windows 2019 afin qu'ils soient sur le même réseau que le serveur Zabbix. Enfin, nous utiliserons un routeur pour permettre à toutes les machines d'accéder à Internet.

Installation et Configuration du serveur Zabbix :

Installation du serveur :

Nous allons nous rendre sur le site officiel de Zabbix afin de choisir le package correspondant à notre configuration :



Nous allons ensuite suivre l'installation préconisée :

Nous allons dans un premier temps récupérer le repository de Zabbix et l'installer :

- `wget https://repo.zabbix.com/zabbix/6.4/debian/pool/main/z/zabbix-release/zabbix-release_latest_6.4+debian11_all.deb`
- `dpkg -i zabbix-release_latest_6.4+debian11_all.deb`
- `apt update`

Nous allons ensuite installer le serveur, le frontend et l'agent Zabbix :

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

Nous devons d'abord créer et configurer la base de données sur notre serveur MariaDB :

1. Créer la base de données Zabbix :

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

2. Créer un utilisateur Zabbix :

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

3. Accorder des privilèges à l'utilisateur Zabbix :

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

4. Activer l'option pour autoriser la création de fonctions et de déclencheurs :

- `mysql> set global log_bin_trust_function_creators = 1;`

5. Quitter MySQL :

- `mysql> quit;`

6.Importation du schéma et des données initiales de Zabbix

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

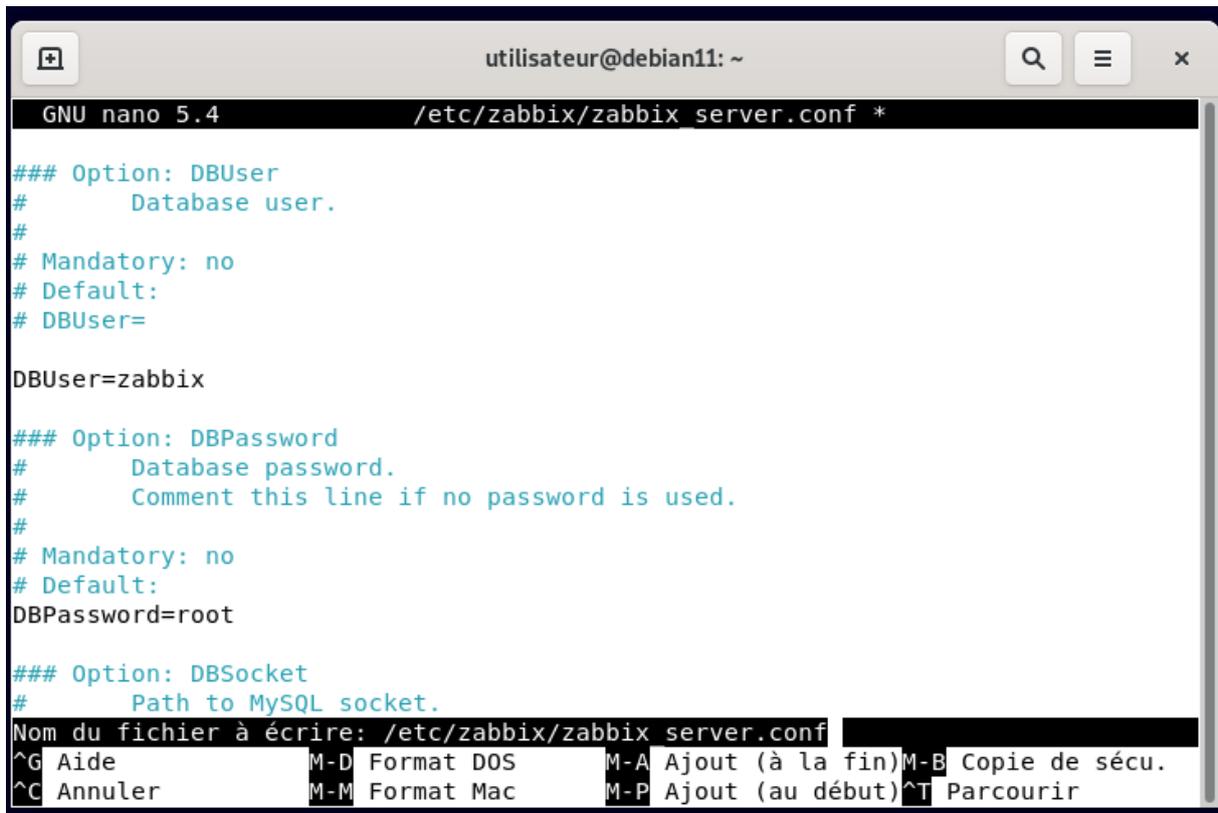
Cette commande décompresse le fichier SQL et l'importe dans la base de données Zabbix.

7.Désactivation de l'option `log_bin_trust_function_creators`

- Connection à MySQL avec l'utilisateur root :
 - `mysql -uroot -p`
- On doit désactiver l'option :
 - `mysql> set global log_bin_trust_function_creators = 0;`
- Quitter MySQL :
 - `mysql> quit;`

Pour finir nous configurons la base de données pour le serveur Zabbix en modifiant le fichier :

- `/etc/zabbix/zabbix_server.conf`



```
utilisateur@debian11: ~
GNU nano 5.4 /etc/zabbix/zabbix_server.conf *
### Option: DBUser
# Database user.
#
# Mandatory: no
# Default:
# DBUser=
DBUser=zabbix

### Option: DBPassword
# Database password.
# Comment this line if no password is used.
#
# Mandatory: no
# Default:
DBPassword=root

### Option: DBSocket
# Path to MySQL socket.
Nom du fichier à écrire: /etc/zabbix/zabbix_server.conf
^G Aide M-D Format DOS M-A Ajout (à la fin) M-B Copie de sécu.
^C Annuler M-M Format Mac M-P Ajout (au début) ^T Parcourir
```

Nous allons maintenant redémarrer et activer notre configuration ainsi que l'agent :

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

Nous allons maintenant pouvoir nous connecter à l'interface web.

Configuration depuis l'interface Web :

Nous pouvons nous connecter en tapant l'adresse ip de notre serveur Zabbix suivi de /zabbix pour avoir accès à l'interface web :

- <http://192.168.30.30/zabbix>

Pour commencer nous allons choisir la langue.



Nous allons ensuite vérifier les prérequis :



Vérification des prérequis

- Bienvenue
- Vérification des prérequis
- Configurer la connexion à la base de données
- Paramètres
- Résumé pré-installation
- Installer

	Valeur actuelle	Requis	
Version de PHP	7.4.33	7.4.0	OK
Option PHP "memory_limit"	128M	128M	OK
Option PHP "post_max_size"	16M	16M	OK
Option PHP "upload_max_filesize"	2M	2M	OK
Option PHP "max_execution_time"	300	300	OK
Option PHP "max_input_time"	300	300	OK
support de bases de données par PHP	MySQL		OK
bcmath pour PHP	sur		OK
mbstring pour PHP	sur		OK
Option PHP "mbstring.func_overload"	inactif	inactif	OK

Retour
Prochaine étape

Nous configureront ensuite la connexion à la base de données :

Configurer la connexion à la base de données

Veuillez créer la base de données manuellement et configurer les paramètres de connexion. Appuyez sur le bouton "Prochaine étape" quand c'est fait.

- Bienvenue
- Vérification des prérequis
- Configurer la connexion à la base de données
- Paramètres
- Résumé pré-installation
- Installer

Type de base de données

Hôte base de données

Port de la base de données 0 - utiliser le port par défaut

Nom de la base de données

Stocker les informations d'identification dans Texte brut Coffre HashiCorp Coffre CyberArk

Utilisateur

Mot de passe

Chiffrement TLS de la base de données La connexion ne sera pas chiffrée car elle utilise un fichier socket (sous Unix) ou de la mémoire partagée (Windows).

Retour
Prochaine étape

Licencié sous [GPL v2](#)

Nous allons définir le nom du serveur, le fuseau horaire et le thème par défaut :



Bienvenue
Vérification des prérequis
Configurer la connexion à la base de données
Paramètres
Résumé pré-installation
Installer

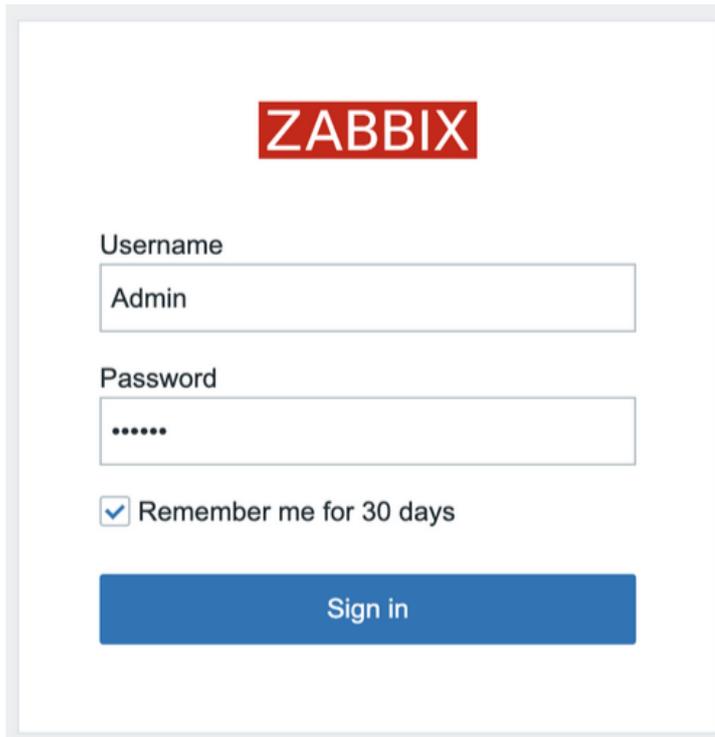
Paramètres

Nom du serveur Zabbix
Fuseau horaire par défaut
Thème par défaut

[Retour](#)[Prochaine étape](#)

L'installation est maintenant terminée !

Par défaut le compte administrateur a pour identifiants :



Username : Admin

Password : zabbix

Nous arrivons donc sur le dashboard :

Paramètre	Valeur	Détails
Le serveur Zabbix est en cours d'exécution	Oui	localhost:10051
Nombre d'hôtes (activé/désactivé)	1	1 / 0
Nombre de modèles	313	
Nombre d'éléments (activés/désactivés/non supportés)	111	100 / 0 / 11
Nombre de déclencheurs (activés/désactivés (problème/ok))	64	64 / 0 / 0 / 64
Nombre d'utilisateurs (en ligne)	2	1
Performance serveur requise, nouvelles valeurs par seconde	1.54	

Statut	Nombre
1 Disponible	1
0 Non disponible	0
0 Inconnu	0
1 Total	1

Sevérité	Nombre
0 Désastre	0
0 Haut	0
0 Moyen	0
0 Avertissement	0
0 Information	0
0 Non classé	0

Le tableau de bord de Zabbix permet de surveiller en temps réel les performances du système. Il offre des fonctionnalités pour :

- Suivre l'utilisation d'un système comme l'utilisation du CPU, de la RAM, et les processus en cours.
- Gérer les alertes (triggers) pour détecter rapidement les incidents critiques.
- Visualiser les hôtes enregistrés selon leur importance.
- Consulter les logs des utilisateurs et les infos du serveur Zabbix.

Zabbix facilite l'analyse des performances, la détection des problèmes de sécurité et la gestion des alarmes, le tout via cette interface personnalisable.

Nous allons maintenant procéder à l'installation des différents agents .

Configuration de l'agent Zabbix sur un Serveurs Windows

Nous allons choisir notre agent sur le site officiel de Zabbix :

The screenshot shows the Zabbix website's 'Get Zabbix' page. It features several installation options: Zabbix Packages, Zabbix Cloud (with a 'Free trial' badge), Third-Party cloud vendors, Zabbix Containers, Zabbix Appliance, Zabbix Sources, and Zabbix Agents. Below these is a section for 'Download pre-compiled Zabbix agent binaries' with a table of options.

OS DISTRIBUTION	OS VERSION	HARDWARE	ZABBIX VERSION	ENCRYPTION	PACKAGING
Windows	Any	amd64	7.2	OpenSSL	MSI
Linux		i386	7.0 LTS	No encryption	Archive
macOS			6.4		
AIX			6.2		
FreeBSD			6.0 LTS		
OpenBSD			5.4		
Solaris			5.2		
			5.0 LTS		
			4.4		
			4.2		
			4.0 LTS		
			3.0 LTS		

Zabbix Release: 6.4.20

Zabbix agent v6.4.20 [Read manual](#)

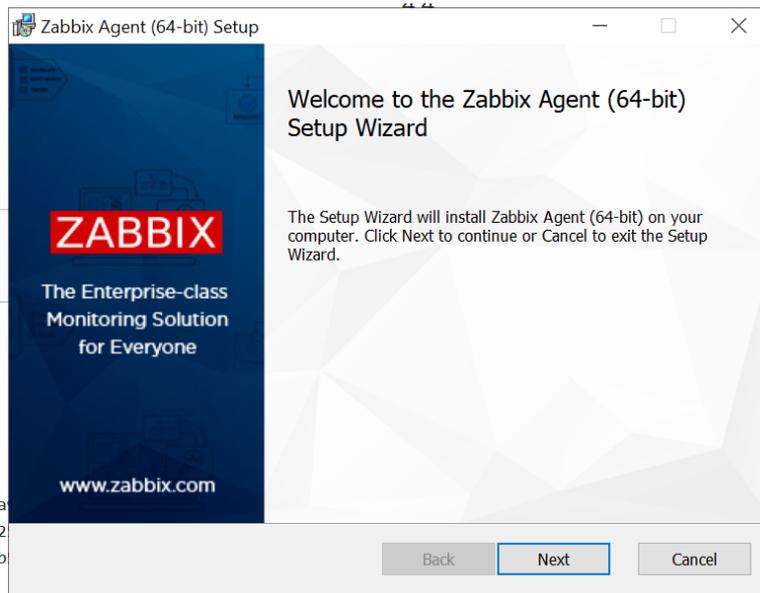
Package: MSI
 Encryption: OpenSSL
 Linkage: Dynamic
 Checksum: sha256: 1f787e4e803a1215e412325e744b0c0222e4f46c09675f6f6c13a39074d
 sha1: 84971f189e7267a1a24e244445c142707c41
 md5: f67e337f4992342b3e16a229f927a8

[DOWNLOAD](https://cdn.zabbix.com/zabbix/binaries/stable/6.4/6.4.20/zabbix_agent-6.4.20-windows-amd64-openssl.msi) https://cdn.zabbix.com/zabbix/binaries/stable/6.4/6.4.20/zabbix_agent-6.4.20-windows-amd64-openssl.msi

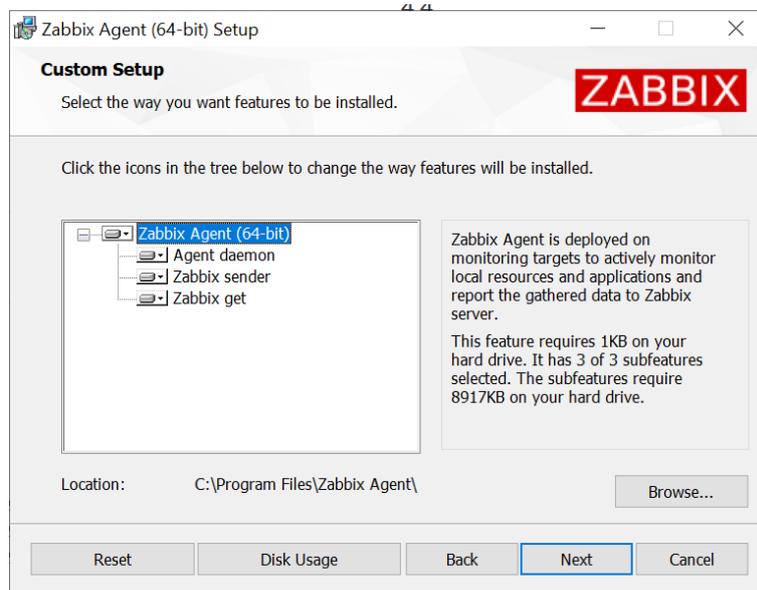
Téléchargement de l'agent :



Début de l'installation :

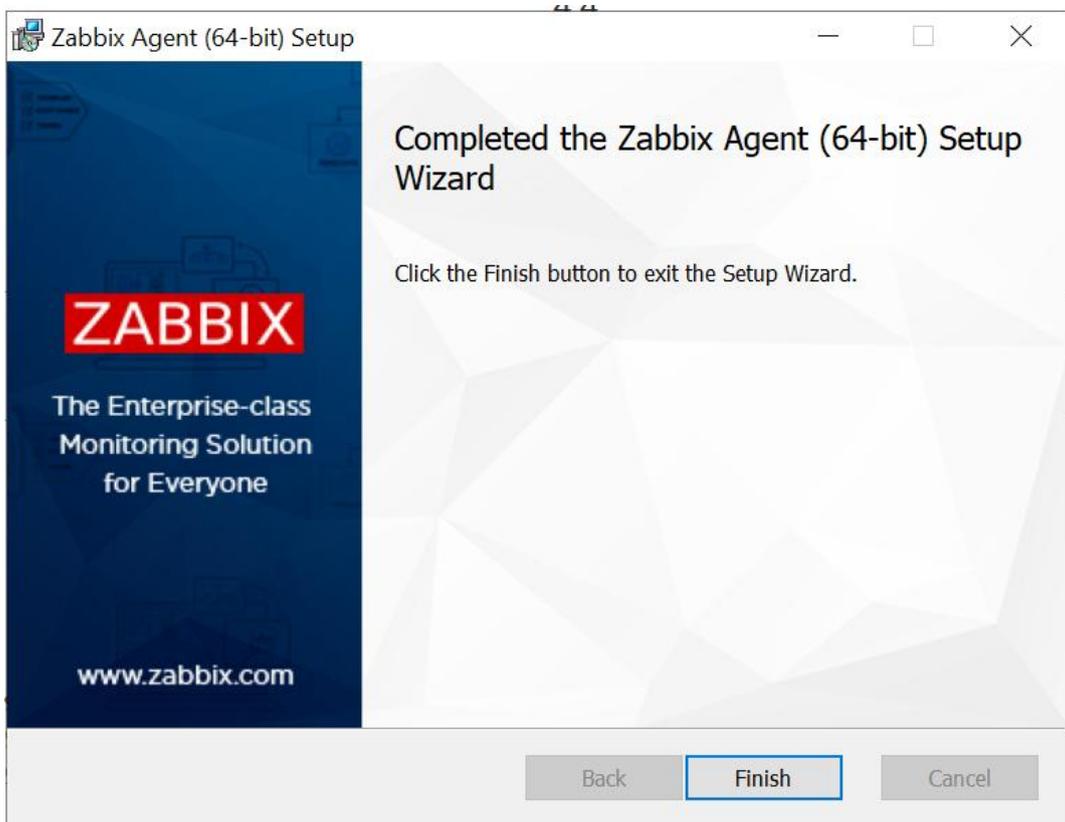
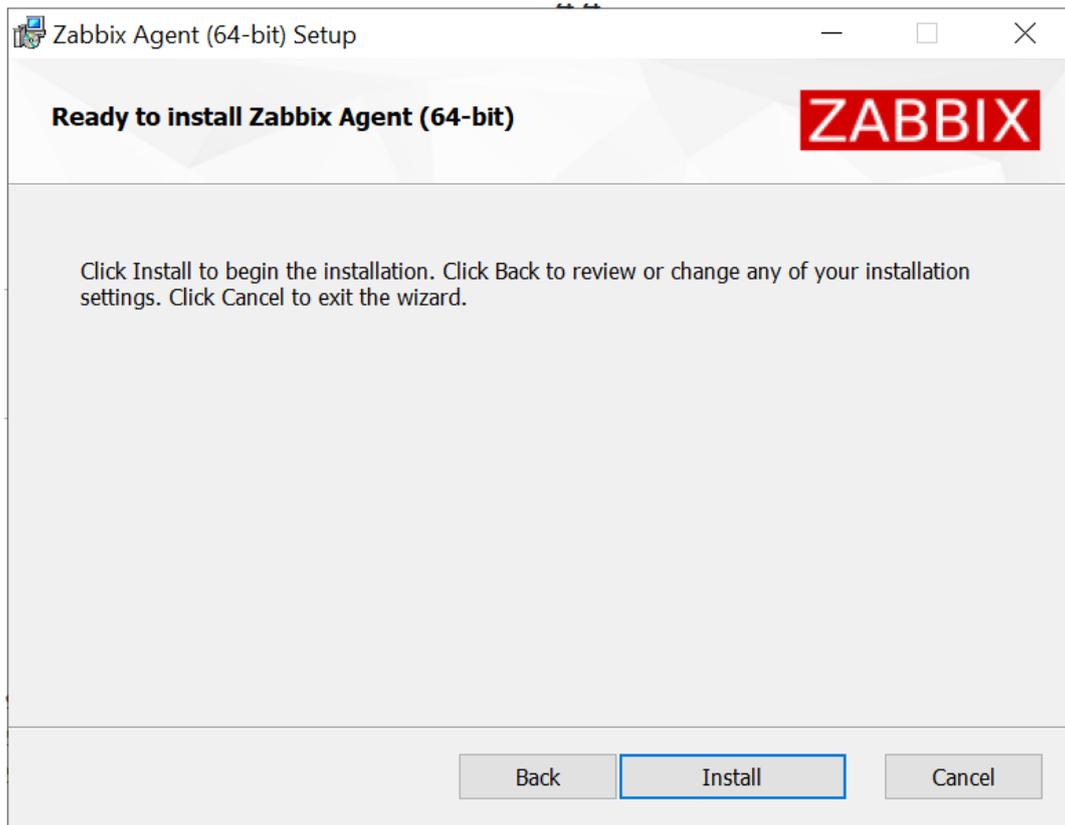


Sélection des éléments à installer :



Configuration de l'agent Zabbix :

Paramètre	Description
<i>Host name</i>	Spécifiez le nom de l'hôte. Ici ServeurWindows
<i>Zabbix server IP/DNS</i>	Spécifiez l'adresse IP/DNS du serveur Zabbix : 192.168.30.30
<i>Agent listen port</i>	Spécifiez le port d'écoute de l'agent (10050 par défaut).
<i>Server or Proxy for active checks</i>	Spécifiez l'adresse IP/DNS du serveur/proxy Zabbix pour les vérifications d'agent actif : 127.0.0.1
<i>Enable PSK</i>	Cochez la case pour activer le support TLS via des clés pré-partagées.
<i>Add agent location to the PATH</i>	Ajouter l'emplacement de l'agent à la variable PATH .



L'agent Zabbix est donc bien installé !

Configuration de l'agent Zabbix sur le serveur Web (Debian11)

Comme pour l'agent Windows nous allons choisir notre agent sur le site officiel de Zabbix :

Choose your platform

ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	ZABBIX COMPONENT	DATABASE	WEB SERVER
7.2	Alma Linux	12 (Bookworm)	Server, Frontend, Agent	---	---
7.0 LTS	CentOS	11 (Bullseye)	Server, Frontend, Agent 2	---	---
6.4	Debian	10 (Buster)	Proxy	---	---
6.0 LTS	OpenSUSE Leap	9 (Stretch)	Agent	---	---
5.0 LTS	Oracle Linux		Agent 2	---	---
	Raspberry Pi OS		Java Gateway	---	---
	Red Hat Enterprise Linux		Web Service	---	---
	Rocky Linux			---	---
	SUSE Linux Enterprise Server			---	---
	Ubuntu			---	---
	Ubuntu (arm64)			---	---

Et suivre l'installation :

Nous allons dans un premier temps récupérer le repository de Zabbix et l'installer :

- `wget https://repo.zabbix.com/zabbix/6.4/debian/pool/main/z/zabbix-release/zabbix-release_latest_6.4+debian11_all.deb`
- `dpkg -i zabbix-release_latest_6.4+debian11_all.deb`
- `apt update`

Puis installer l'agent Zabbix :

- `apt install zabbix-agent`

Nous allons maintenant redémarrer et activer notre agent :

- `systemctl restart zabbix-agent`
- `systemctl enable zabbix-agent`

Et on modifie le fichier de configuration, dans la même idée que pour Windows, afin que Zabbix accède aux données de notre serveur Debian.

On modifie le fichier : `/etc/zabbix/zabbix_agentd.conf`

```

GNU nano 5.4
Server=192.168.30.30

### Option: ListenPort
#   Agent will listen on
#
# Mandatory: no
# Range: 1024-32767
# Default:
ListenPort=10050

### Option: ListenIP
#   List of comma delimi
#   First IP address is
#
# Mandatory: no
# Default:
ListenIP=192.168.30.10

### Option: StartAgents
#   Number of pre-forked
#   If set to 0, disable
#
# Mandatory: no
# Range: 0-100
# Default:
# StartAgents=3

##### Active checks related

### Option: ServerActive
#   Zabbix server/proxy
#   Server/proxy address
#   Cluster configuratio
#   Multiple Zabbix serv
#   More than one Zabbix
#   If Zabbix proxy is s
#   Multiple comma-delim
#   If port is not speci
#   IPv6 addresses must
#   If port is not speci
#   If this parameter is
#   Example for Zabbix p
#   ServerActive
#   Example for multiple
#   ServerActive
#   Example for high ava
#   ServerActive
#   Example for high ava
#   ServerActive
#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=192.168.30.30

### Option: Hostname
#   List of comma delimi
#   Required for active
#   Value is acquired fr
#
# Mandatory: no
# Default:
# Hostname=

Hostname=ServeurDebian

```

Server=192.168.30.30 Les connexions entrantes seront acceptées uniquement à partir des hôtes listés ici.
ListenPort=10050 L'agent écoutera sur ce port les connexions provenant du serveur.
ListenIP=192.168.30.10 Liste des adresses IP sur lesquelles l'agent doit écouter.
ServerActive=192.168.30.30 Liste des adresses IP du serveurs que doit contacter l'agent Zabbix pour qu'il effectue des vérifications actives
Hostname=ServeurDebian Nom d'hôte du serveur.

 Aide  Écrire

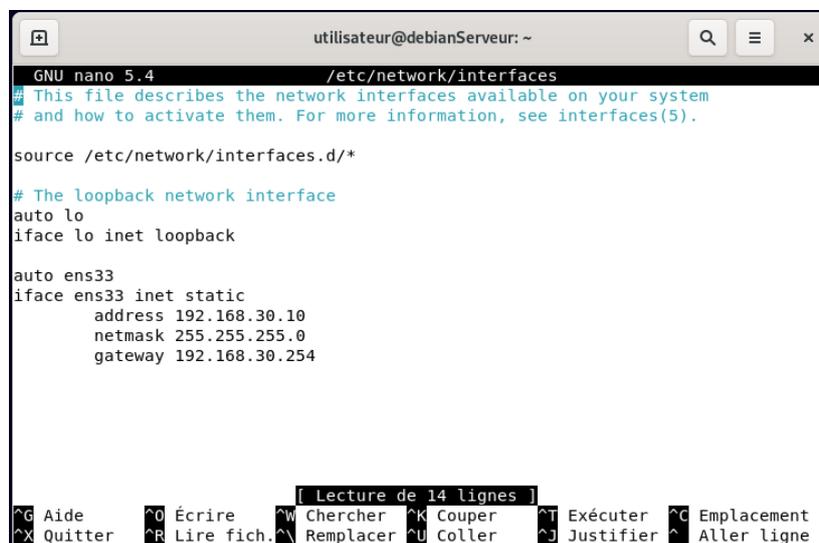
Configuration des interfaces réseaux :

On va d'abord aller dans le fichier `/etc/network/interfaces` afin de paramétrer notre carte réseaux avec la commande `sudo nano /etc/network/interfaces`.

On redémarrera les interfaces avec la commande `sudo systemctl restart networking.service` pour que les changements prennent effet.

Il ne faut pas oublier d'activer l'accès par pont.

Pour notre serveur Web on lui définit comme adresse IP : [192.168.30.10](#)



```
utilisateur@debianServeur: ~
GNU nano 5.4 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

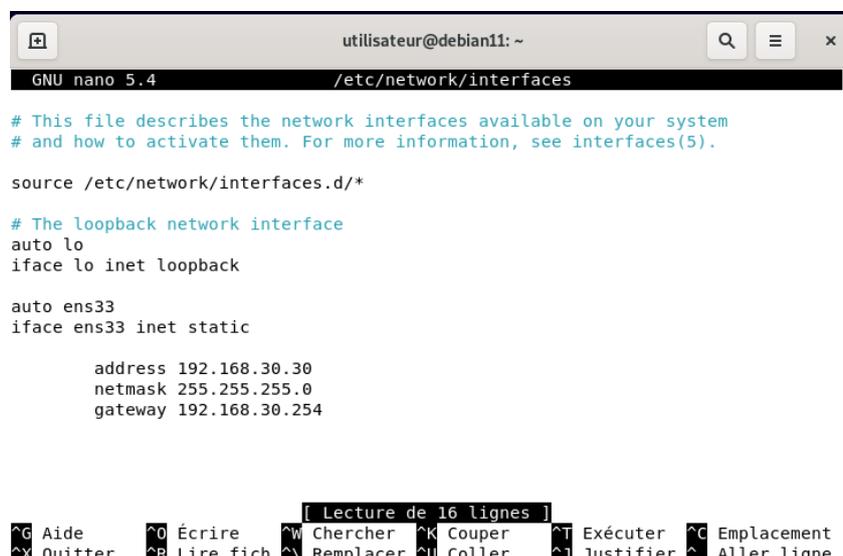
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet static
    address 192.168.30.10
    netmask 255.255.255.0
    gateway 192.168.30.254

[ Lecture de 14 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
```

Notre serveur Zabbix a pour adresse IP : [192.168.30.30](#)



```
utilisateur@debian11: ~
GNU nano 5.4 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

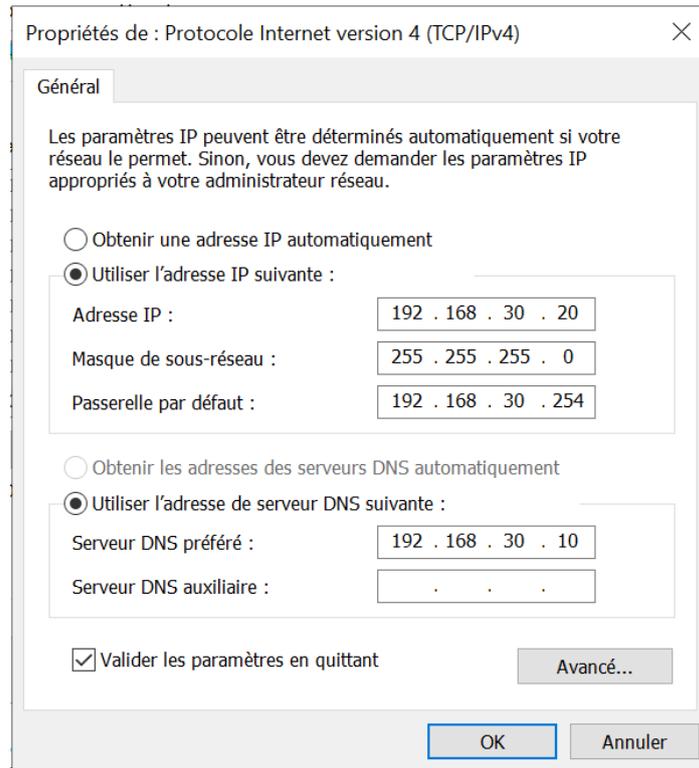
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet static
    address 192.168.30.30
    netmask 255.255.255.0
    gateway 192.168.30.254

[ Lecture de 16 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
```

Notre serveur Windows 2019 aura pour adresse IP : [192.168.30.20](#)



Et pour notre routeur :

```
utilisateur@debianRouter: ~  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
auto ens33  
iface ens33 inet dhcp  
post-up iptables-restore < /etc/iptables_rules.save  
  
auto ens34  
iface ens34 inet static  
    address 192.168.10.254  
    netmask 255.255.255.0  
  
auto ens35  
iface ens35 inet static  
    address 192.168.20.254  
    netmask 255.255.255.0  
  
auto ens36  
iface ens36 inet static  
    address 192.168.30.254  
    netmask 255.255.255.0  
root@debianRouter:~#
```

Nous allons maintenant passer à la création des hôtes.

Création des Hôtes :

Pour superviser notre parc, l'installation d'un agent ne suffit pas. Il est indispensable de créer un hôte (host) sur le serveur Zabbix.

Pour cela Il faut aller dans : [Collecte de données](#) → [Hôtes](#) → [Créer un hôte](#).

Une fois sur la page de création, on pourra configurer les hôtes correspondant aux machines de notre parc.

The screenshot shows the 'Nouvel hôte' (New Host) configuration page in Zabbix. The page has a navigation bar with tabs: 'Hôte', 'IPMI', 'Tags', 'Macros', 'Inventaire', 'Chiffrement', and 'Table de correspondance'. The 'Hôte' tab is active. The form contains the following fields and options:

- Nom de l'hôte**: A required text input field.
- Nom visible**: A text input field.
- Modèles**: A search input field with the placeholder 'taper ici pour rechercher' and a 'Sélectionner' button.
- Groupes d'hôtes**: A search input field with the placeholder 'taper ici pour rechercher' and a 'Sélectionner' button.
- Interfaces**: A table with columns: Type, adresse IP, Nom DNS, Connexion à, Port, and Défaut.
 - Row 1: Type 'Agent', adresse IP '127.0.0.1', Nom DNS (empty), Connexion à 'IP' (selected), Port '10050', and Défaut 'Supprimer' (radio button).
- Description**: A large text area with an 'Ajouter' link above it.
- Surveillé via le proxy**: A dropdown menu currently set to '(pas de proxy)'.
- Activé**: A checked checkbox.

At the bottom right, there are two buttons: 'Ajouter' (Add) and 'Annuler' (Cancel).

Nous devons configurer les champs suivants :

- **Nom de l'hôte (« Host name ») :**
Indiquez le nom exact de la machine à superviser. Ce nom doit correspondre à celui utilisé lors de la configuration de l'agent Zabbix.
- **Modeles (« Templates ») :**
Zabbix propose des modèles préconfigurés appelés *Templates*, qui incluent des règles prédéfinies telles que le suivi de la charge CPU, l'utilisation de la RAM, le trafic réseau, etc.
Il est essentiel de choisir le Template adapté au type de machine. Par exemple, pour un environnement Windows, le Template « [Windows by Zabbix agent](#) » est recommandé afin de collecter un maximum d'informations pertinentes.
- **Interfaces :**
Cette section définit le point de connexion entre l'agent et le serveur. Sélectionnez «

Agent Zabbix », puis saisissez l'adresse IP de la machine ainsi que le port de communication, généralement 10050.

- **Groupes d'hôtes (« Host groups ») :**

Cette rubrique permet de regrouper les machines par catégories, facilitant ainsi leur gestion, notamment dans des environnements comptant plusieurs centaines de dispositifs.

- **Description :**

Il est conseillé d'ajouter une brève description pour identifier facilement à quelle machine correspond l'hôte.

- **Activation de l'hôte :**

Pour finaliser la configuration, assurez-vous de cocher la case « **Activé** » en bas du formulaire afin d'activer l'hôte.

Création de l'hôte Windows :

Voici le formulaire pour la création de l'hôte Windows :

Hôte

[Hôte](#) [IPMI](#) [Tags](#) [Macros](#) [Inventaire](#) [Chiffrement](#) [Table de correspondance](#)

* Nom de l'hôte

Nom visible

Modèles

Nom	Action
Windows by Zabbix agent	Supprimer lien Supprimer lien et nettoyer

* Groupes d'hôtes

Interfaces

Type	adresse IP	Nom DNS	Connexion à	Port	Défaut
Agent	<input type="text" value="192.168.30.20"/>	<input type="text"/>	<input type="button" value="IP"/> <input type="button" value="DNS"/>	<input type="text" value="10050"/>	<input checked="" type="radio"/> Supprimer

[Ajouter](#)

Description

Surveillé via le proxy

Activé

Création de l'hôte Debian :

Voici le formulaire pour la création de l'hôte Debian :

Hôte

Hôte IPMI Tags Macros Inventaire Chiffrement Table de correspondance

* Nom de l'hôte

Nom visible

Modèles

Nom	Action
Linux by Zabbix agent	Supprimer lien Supprimer lien et nettoyer

* Groupes d'hôtes

Interfaces

Type	adresse IP	Nom DNS	Connexion à	Port	Défaut
Agent	<input type="text" value="192.168.30.10"/>	<input type="text"/>	<input checked="" type="radio"/> IP <input type="radio"/> DNS	<input type="text" value="10050"/>	<input checked="" type="radio"/> Supprimer

[Ajouter](#)

Description

Surveillé via le proxy (pas de proxy)

Activé

Nous pouvons aller voir si les agents sont bien remontés :

Nom ▲	Interface	Disponibilité	Tags	État	Dernières données	Problèmes	Graphiques	Tableaux de bord	Web
ServeurDebian	192.168.30.10:10050	ZBX	class: os target: linux	Activé	Dernières données 68	1	Graphiques 14	Tableaux de bord 3	Web
ServeurWindows	192.168.30.20:10050	ZBX	class: os target: windows	Activé	Dernières données 110	1	Graphiques 12	Tableaux de bord 3	Web
Zabbix server	127.0.0.1:10050	ZBX	class: os class: software target: linux ...	Activé	Dernières données 129	1	Graphiques 25	Tableaux de bord 5	Web

Affichage de 3 sur 3 trouvés

Vérification des alertes :

Vérification et configuration des alertes :

Lors de la création de l'hôte, nous avons sélectionné un modèle. Par exemple, pour un environnement Windows, nous avons choisi le [Template « Windows by Zabbix agent »](#). Cela nous a permis de disposer d'une liste d'éléments déjà configurés, facilitant la collecte d'informations système telles que la charge CPU, l'utilisation de la mémoire, etc.

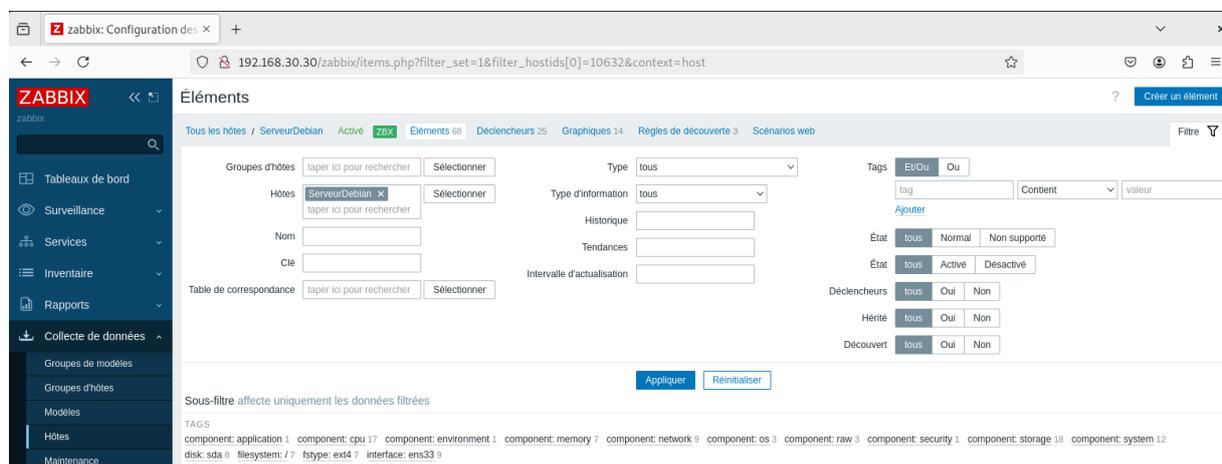
Ce Template comprend également des [triggers](#) que l'on peut configurer selon nos besoins. Ils permettent de générer des [alertes](#), par exemple lorsque qu'un serveur ne répond plus ou qu'une charge CPU est trop élevée.

Dans la section de collecte de données, il est possible de sélectionner un hôte afin de visualiser les différents éléments de surveillance et les déclencheurs (triggers) qui lui sont associés.

Par exemple, pour notre serveur web sous Debian, on observe qu'il dispose de [68 éléments de surveillance](#) et de [25 déclencheurs configurés](#).

Parmi les éléments surveillés par ce template, on retrouve des métriques essentielles telles que :

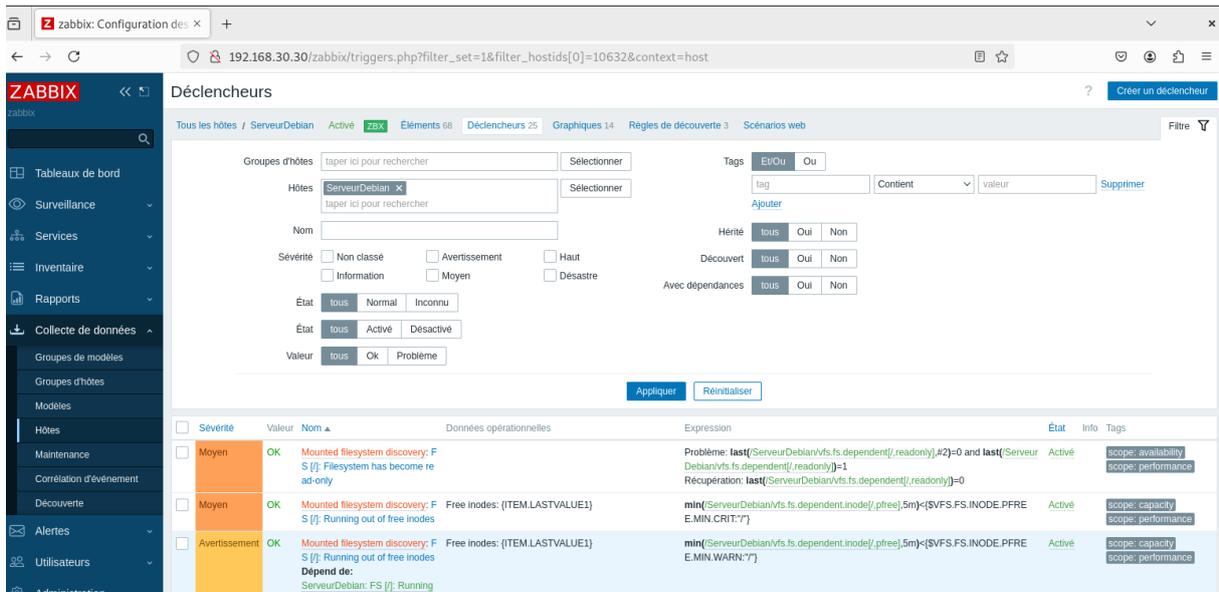
- La charge CPU pour évaluer l'utilisation des ressources processeur,
- La gestion de la mémoire vive (RAM) pour détecter d'éventuelles saturations,
- La réponse au ping, qui permet de vérifier la disponibilité du serveur,
- Ainsi que d'autres indicateurs liés aux performances réseau, à l'espace disque, et aux processus en cours.



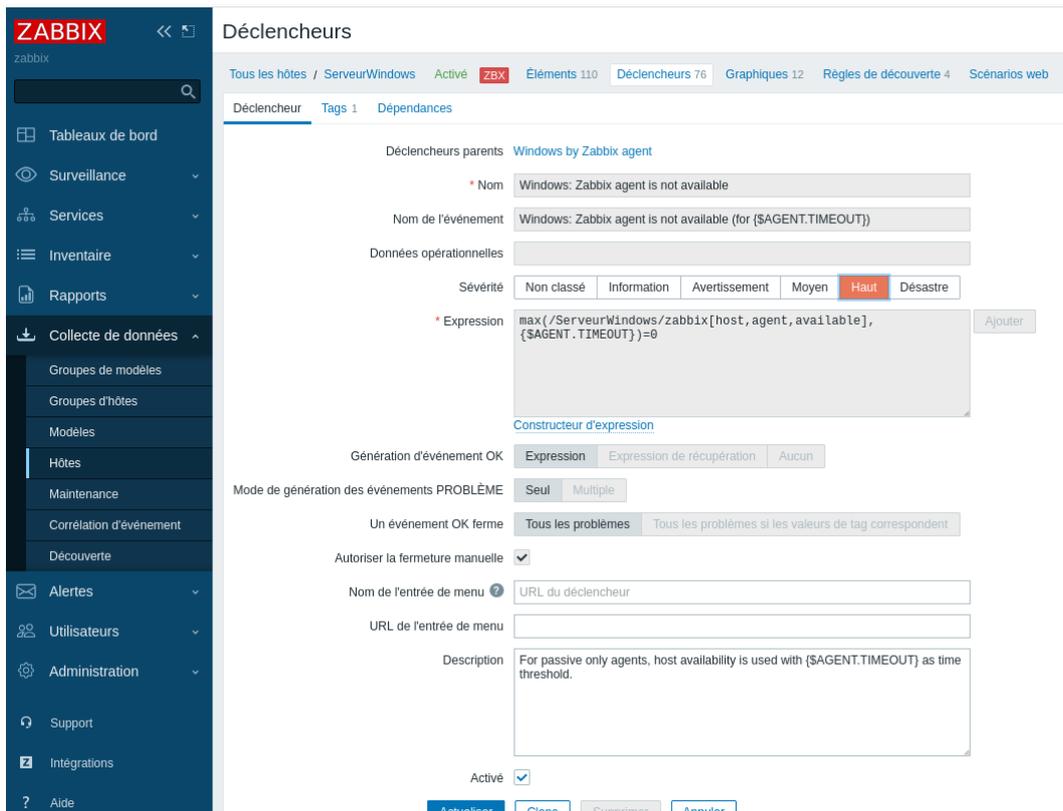
The screenshot shows the Zabbix web interface for configuring monitoring items. The browser address bar indicates the URL: 192.168.30.30/zabbix/items.php?filter_set=1&filter_hostids[0]=10632&context=host. The page title is 'Éléments'. The left sidebar contains navigation links: 'Tableaux de bord', 'Surveillance', 'Services', 'Inventaire', 'Rapports', and 'Collecte de données'. The main content area is divided into several sections: 'Groupes d'hôtes' (with a search field and 'Sélectionner' button), 'Hôtes' (with a dropdown menu showing 'ServeurDebian' and a 'Sélectionner' button), 'Nom' and 'Clé' (input fields), 'Table de correspondance' (with a search field and 'Sélectionner' button), 'Type' (dropdown menu set to 'tous'), 'Type d'information' (dropdown menu set to 'tous'), 'Historique' (input field), 'Tendances' (input field), 'Intervalle d'actualisation' (input field), 'Tags' (with 'Ei/Ou' and 'Ou' buttons, a search field, and a 'Contient' dropdown), 'État' (with 'tous', 'Normal', and 'Non supporté' buttons), 'État' (with 'tous', 'Activé', and 'Désactivé' buttons), 'Déclencheurs' (with 'tous', 'Oui', and 'Non' buttons), 'Hérité' (with 'tous', 'Oui', and 'Non' buttons), and 'Découvert' (with 'tous', 'Oui', and 'Non' buttons). At the bottom, there are 'Appliquer' and 'Réinitialiser' buttons. A note states: 'Sous-filtre affecte uniquement les données filtrées'. Below this, a list of tags is shown: 'TAGS component: application 1 component: cpu 17 component: environment 1 component: memory 7 component: network 9 component: os 3 component: raw 3 component: security 1 component: storage 18 component: system 12 disk: sda 6 filesystem: /7 fstype: ext4 7 interface: ens33 3'.

Ces données sont essentielles pour garantir la disponibilité des services et anticiper d'éventuels problèmes de performance.

Les déclencheurs peuvent être personnalisés afin de générer des alertes en cas d'anomalies, comme une surcharge CPU, une perte de connectivité réseau, ou une utilisation excessive de la mémoire



Par exemple, nous pouvons ajuster le niveau de **sévérité** en cas de perte de connexion de notre serveur Windows :



Maintenant que nous avons vérifié que les alertes nécessaires sont bien présentes, nous pouvons tester leur remontée.

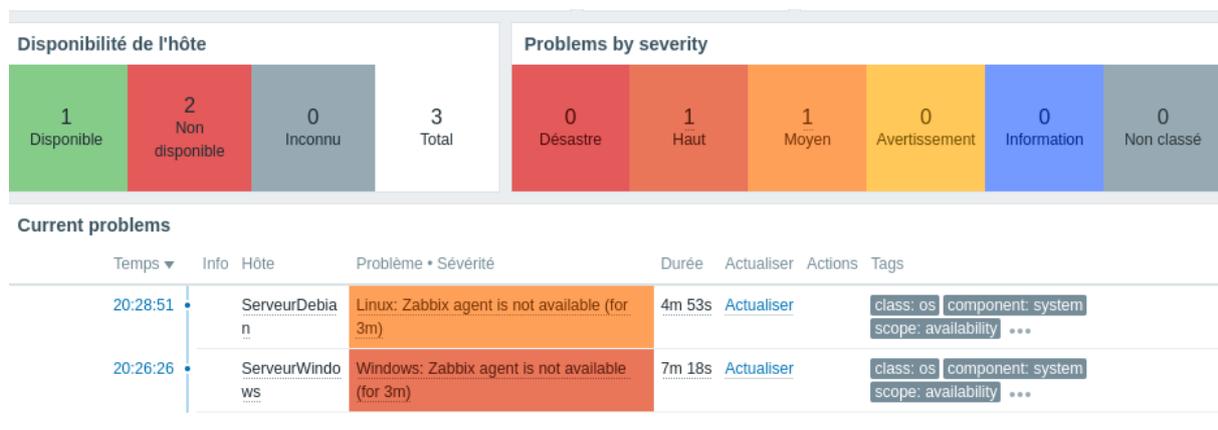
Test de remonter des alertes :

Nous allons procéder à la vérification et aux tests des alertes de notre superviseur Zabbix.

Pour cela, nous commencerons par ne pas démarrer notre serveur Windows ainsi que notre serveur web Linux.

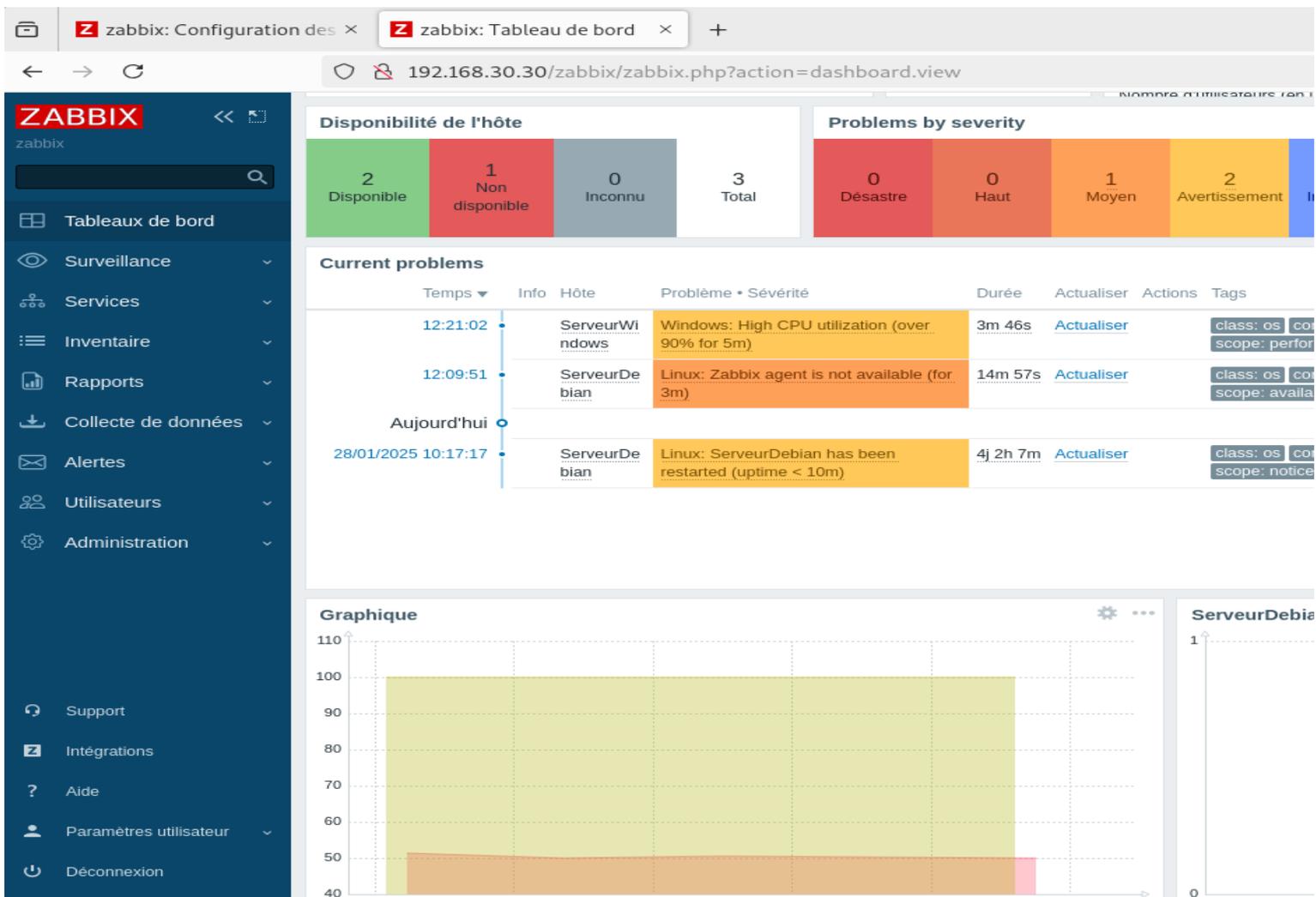
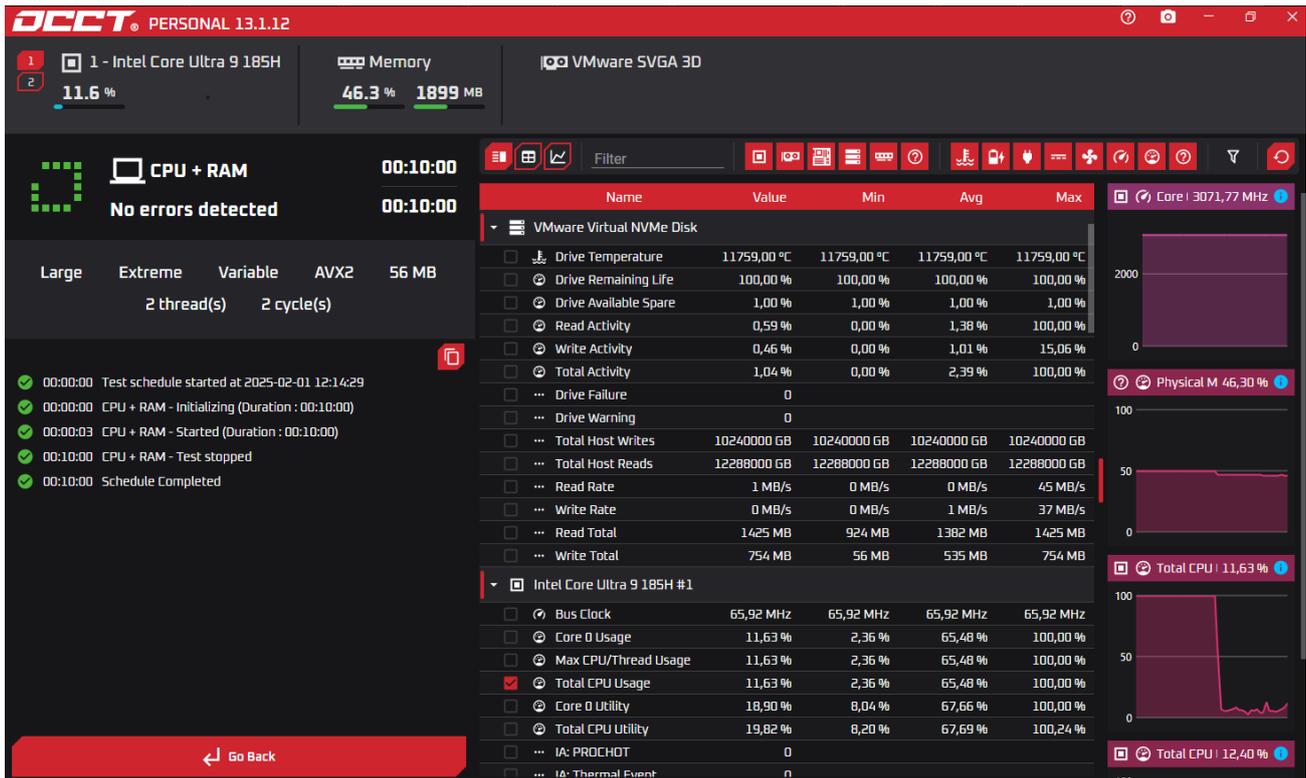
Nous pourrons ainsi constater que les alertes remontent correctement et observer la différence entre deux types d'alertes : celle que nous avons modifiée et celle utilisant la configuration par défaut.

Pour le serveur Windows, l'alerte remonte correctement avec une gravité **élevée**, tandis que sur le serveur web utilisant la configuration par défaut, elle apparaît avec une gravité **moyenne**.



Pour poursuivre nos tests, nous allons utiliser **OCCT**, un outil permettant de solliciter et de mettre à l'épreuve nos différents composants matériels, tels que le **CPU**, la **RAM**, etc. Ces tests seront effectués sur le poste Windows.

La RAM sera sollicitée à **50%**, ce qui ne devrait normalement pas générer d'alerte, tandis que le CPU sera poussé à **100%**, et nous devrions donc recevoir une alerte en conséquence.



Nous pouvons constater qu'un avertissement est apparu, nous alertant que l'utilisation du CPU a dépassé les 90 % depuis plus de 5 minutes.

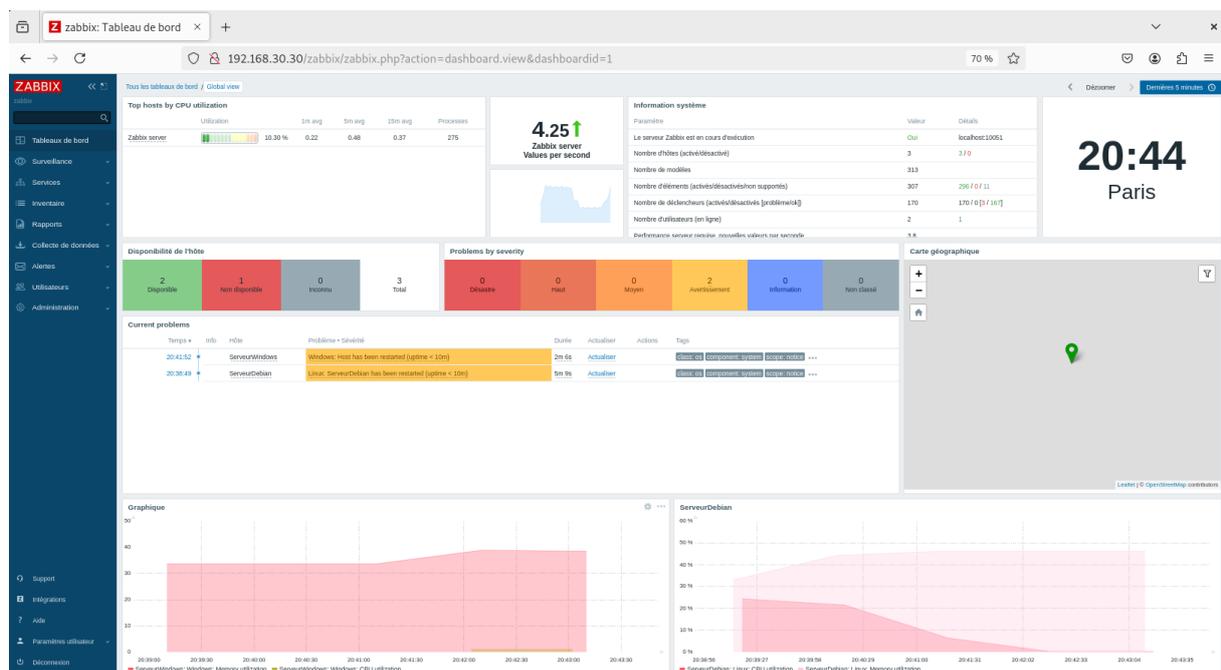
Toutes ces alertes peuvent être configurées manuellement. De plus, il est possible de surveiller des services spécifiques tels que HTTP, FTP, etc.

Configuration du Dashboard :

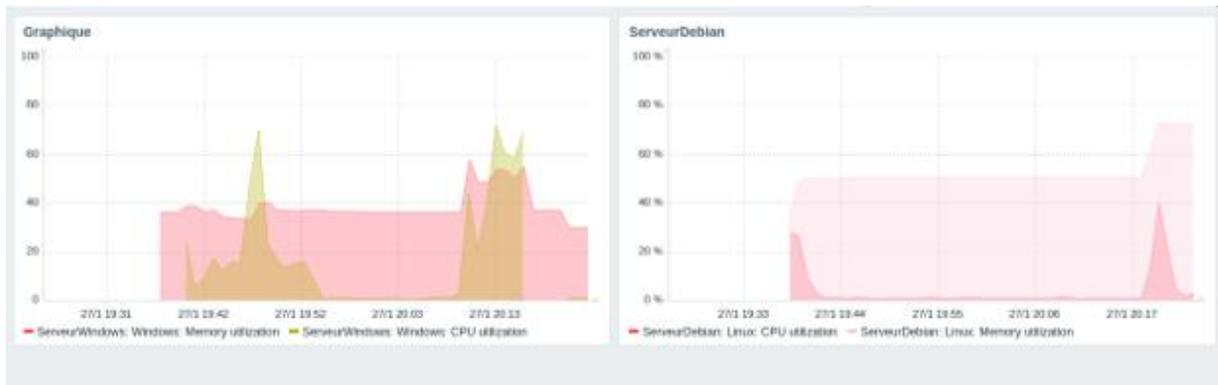
Le tableau de bord de surveillance de Zabbix offre une vue en temps réel de l'utilisation du CPU, de la RAM, des processus en cours, de la disponibilité des hôtes et des éventuels problèmes rencontrés sur le système.

Il permet également de suivre les performances réseau, de surveiller les services essentiels (HTTP, FTP, etc.), et de visualiser les alertes en cas de dépassement de seuils critiques.

Grâce à des graphiques interactifs et des rapports détaillés, il facilite l'analyse des tendances et pour une gestion proactive de l'infrastructure informatique.



Nous avons opté pour l'affichage de graphiques représentant l'utilisation du CPU et de la RAM pour le serveur web, chacun sous forme de courbes distinctes. La même configuration a été appliquée pour le serveur web sous Debian :



Conclusion :

L'utilisation de Zabbix pour la mise en place de notre outil de supervision réseau nous a permis d'atteindre tous les objectifs définis dans le cahier des charges, en assurant une surveillance efficace et une gestion proactive de notre environnement informatique.

Mise en place de la solution de supervision :

Nous avons installé et configuré Zabbix sur notre infrastructure, en détaillant la procédure d'installation du logiciel dans un environnement Linux. Nous avons également installé les agents nécessaires sur les postes clients et vérifié leur bonne intégration dans Zabbix.

Configuration des hôtes et des alertes :

Trois hôtes ont été créés et configurés : un serveur Windows, un serveur Zabbix et un serveur web sous Debian. Nous avons assigné des items à chaque hôte, ce qui permet aux agents de collecter et d'extraire les données pertinentes ainsi que les événements associés.

Surveillance et alertes :

Nous avons mis en place des alertes spécifiques pour chaque hôte, permettant de notifier l'administrateur en cas de problèmes sur les machines clientes ou serveurs. Cela nous permet de réagir rapidement en cas d'incidents, améliorant ainsi la disponibilité et la performance des systèmes.

En résumé, Zabbix nous a permis de surveiller efficacement notre infrastructure informatique en temps réel, d'optimiser les opérations de maintenance et de garantir une meilleure disponibilité des services. Grâce à cette solution de supervision, nous avons automatisé la gestion des incidents, renforçant la performance globale de notre environnement informatique.