

**SUPPORT DE FORMATION INSTALLATION ET
CONFIGURATION DE ZABBIX 6**



ZABBIX

Sommaire

INTRODUCTION.....	3
I. Présentation des Composants	3
I.1 Modes de collecte des données.....	4
I.2 Modes de Découverte	5
I.3 Analyse Comparative	5
II. Concepts & Définitions.....	6
III. Déploiement de notre infrastructure de monitoring.....	7
III.1 Architecture	7
III.2 Installation et configuration de la base de donnée PostgreSQL.....	7
III.3 Installation et configuration du server zabbix.....	10
III.4 Installation des agents	15
III.4.1 Installation agent Linux.....	15
III.4.2 Installation agent Windows	22
III.5 Création d'une métrique personnalisée	25
Conclusion	26

Franck TCHINKOU

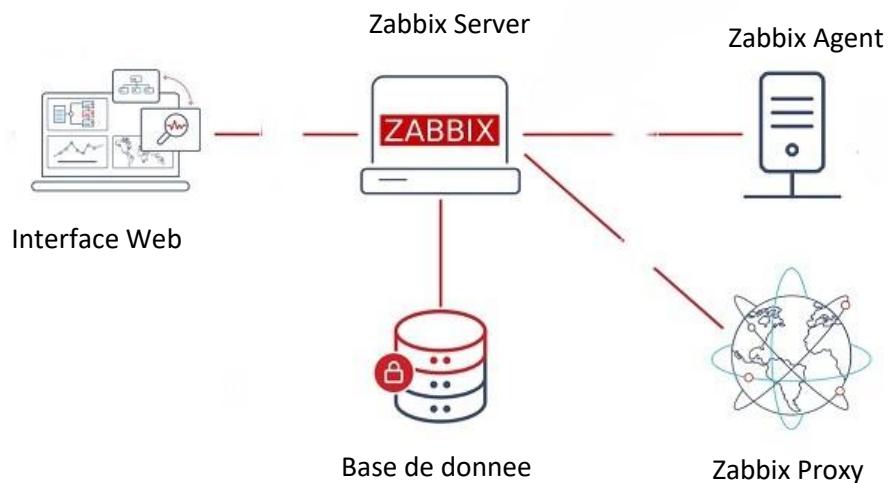
Version 1.0

INTRODUCTION

Zabbix est une solution de supervision / monitoring de niveau entreprise, open-source et distribuée. Son rôle est de surveiller l'état des services réseau, du matériel serveur et des applications en temps réel. Pour une organisation, c'est l'outil qui garantit la disponibilité du système d'information.

I. Présentation des Composants

L'architecture de zabbix repose sur plusieurs piliers interconnectés :



Zabbix Server : Le cerveau du système. Il exécute les vérifications, gère les alertes et stocke les données.

Interface Web (FrontEnd) : L'interface graphique (souvent sous Nginx ou Apache) qui permet de visualiser les données et de configurer l'outil.

Base de données : Le lieu de stockage de toutes les configurations et données historiques (dans notre cas **PostgreSQL**).

Zabbix Agent : Le logiciel installé sur les machines surveillées pour remonter des données précises au serveur.

Zabbix Proxy : Un composant qui collecte les données sur un site distant ou sur les systèmes de faible capacité sur lesquels on n'a pas installé d'agents et les renvoie au serveur central, idéal pour décharger le serveur principal.

I.1 Modes de collecte des données

Il existe plusieurs façons de récupérer des informations :

- **Agent Zabbix (Actif/Passif)** :

Agent Passif : Le serveur va récupérer les données.

Firewall (autoriser l'entre des flux vers les différents serveurs exposition trop grande) : Le serveur zabbix doit avoir accès à tous les agents pour pouvoir collecter les données.

Si l'infrastructure est très grande le serveur zabbix va beaucoup travailler car il doit récupérer les métriques sur l'ensemble du parc à la fois.

Agent Actif : L'agent envoie ses données de lui-même

Firewall (autoriser la sortie des flux des machines vers le serveur zabbix idéal pour franchiser les pare-feu).

Charge moindre sur le serveur zabbix.

- **Scripts personnalisés (UserParameters)** : Permettent d'exécuter des commandes Linux spécifiques (ex : script de comptage de processus).
- **SNMP (Simple Network Management Protocol)** : Utilisé pour surveiller les équipements réseaux (switches, routeurs, imprimantes) qui ne peuvent pas recevoir d'agent.
- **Vérifications Basiques (Simple Checks)** : Surveillance sans agent, comme le test de disponibilité d'un port (HTTP, SSH) ou le ping ICMP.

I.2 Modes de Découverte

Pour éviter d'ajouter chaque machine à la main, Zabbix propose :

Découverte réseau (Network Discovery) : Scanne une **plage IP** définie pour détecter la présence d'agents.

Auto-enregistrement (Autoregistration) : L'agent actif se présente de lui-même au serveur dès qu'il est installé.

Agent SNMP.

I.3 Analyse Comparative

Solution	Points forts	Points faibles
Nagios	Historique, très stable.	Configuration complexe (fichiers textes), interface datée.
Centreon	Basé sur Nagios, interface moderne, très bon reporting.	Moins flexible que Zabbix pour le monitoring actif.
Prométhée	Excellent pour le Cloud et Kubernetes (Séries temporelles).	Difficile à configurer pour du matériel "standard" (serveurs physiques).
Zabbix	Tout-en-un, ultra flexible, gestion native des agents actifs.	Courbe d'apprentissage un peu raide au début.

II. Concepts & Définitions

Groupes : Ensemble d'éléments similaires permettant une configuration commune.

Proxy zabbix : agent intermédiaire permettant de regrouper la collecte d'agent dédié.

Sonde zabbix : composant déployé pour surveiller.

Evènement : seuil ou déclencheur activant une action.

Trigger : déclenchement ou franchissement d'un seuil d'alerte.

Webhook : transmission d'informations ou d'action à d'autres applications.

Macro : variation générique définie pour être réutilisée.

Low level discovery (LLD) : découverte automatique des éléments à surveiller.

Host Template : modèle réutilisable pour configurer le monitoring des hôtes spécifiques et similaires.

Intégrations : plugin permettant la connexion à d'autres outils (docker, Cloud).

Notification : messages envoyés via des systèmes de communication (sms, email).

Remédiation : lancer des actions pour résoudre des problèmes remontés par la surveillance.

Ex : restart de services.

Authentification : zabbix permet de coupler son authentification à LDAP et d'autres SSO.

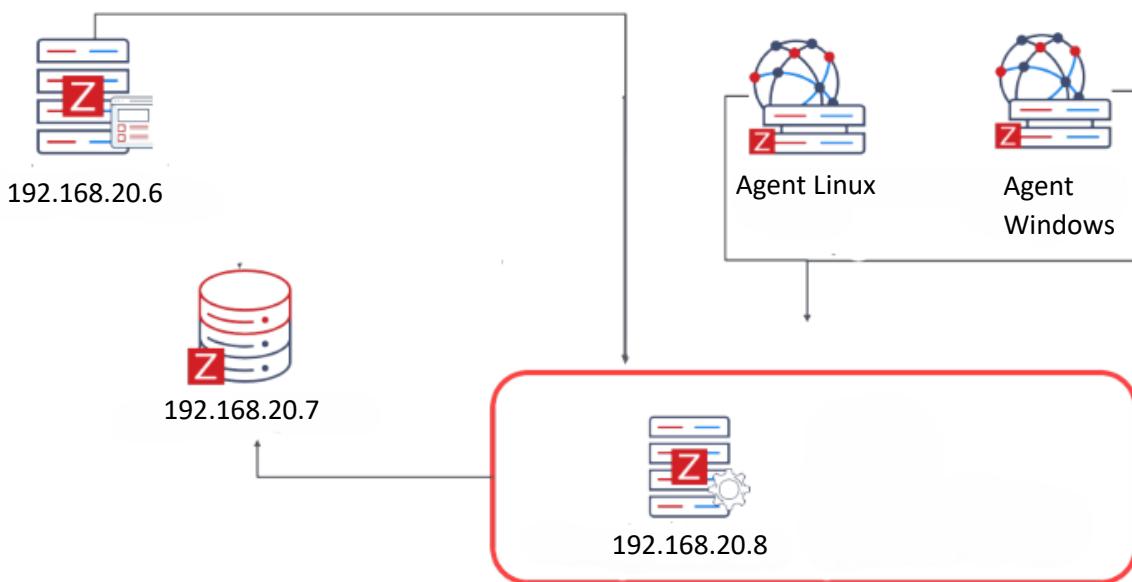
Agent actif : l'agent initie et réalise la collecte des données (sans intervention du serveur).

Agent passif : l'agent collecte les données à la demande de serveur zabbix.

III. Déploiement de notre infrastructure de monitoring

Cette partie détaille le déploiement technique de la solution au sein de notre environnement de test.

III.1 Architecture



Server zabbix : Ubuntu 22.04.4 LTS

Server PostgreSQL : Ubuntu 22.04.4 LTS

III.2 Installation et configuration de la base de donnée PostgreSQL

Le choix de PostgreSQL s'est porté sur sa robustesse et sa gestion efficace des gros volumes de données historiques.

Nous avons utilisé la commande **apt install -y postgresql** pour l'installation.

Puis nous avons créé le cluster main en utilisant la commande **pg_ctlcluster 14 main start**. Cette commande lance les processus en arrière-plan pour que la base de données soit prête à recevoir des connexions. Sans cette commande il serait impossible de se connecter à la base de données.

Création de la database zabbix (Nous avons nomé zabbix car par défaut dans l'installation du serveur zabbix la base de données se nomme déjà zabbix) ça nous permet de gagner en temps.

```
root@srv-postgresql:~# sudo -u postgres psql -c "create database zabbix;"  
could not change directory to "/root": Permission denied  
CREATE DATABASE
```

On obtient l'erreur permission denied car on exécute la commande en tant qu'utilisateur postgres (sudo –u postgres) mais nous sommes connectés dans le /root or l'utilisateur postgres n'a pas les droits pour accéder à /root. Donc postgres SQL essaie d'aller dans le dossier courant /root mais ça refuse néanmoins la base de donnée est bien créée.

Vérifions la création de la database zabbix.

```
root@srv-postgresql:~# su - postgres  
postgres@srv-postgresql:~$ psql  
psql (14.20 (Ubuntu 14.20-0ubuntu0.22.04.1))  
Type "help" for help.  
  
postgres=# \l  
          List of databases  
   Name    |  Owner   | Encoding | Collate  |  Ctype   | Access privileges  
-----+-----+-----+-----+-----+-----+  
postgres | postgres | UTF8    | fr_FR.UTF-8 | fr_FR.UTF-8 | =c/postgres      +  
template0 | postgres | UTF8    | fr_FR.UTF-8 | fr_FR.UTF-8 | postgres=CTc/postgres  
template1 | postgres | UTF8    | fr_FR.UTF-8 | fr_FR.UTF-8 | =c/postgres      +  
zabbix   | postgres | UTF8    | fr_FR.UTF-8 | fr_FR.UTF-8 | postgres=CTc/postgres  
(4 rows)  
  
postgres=# _
```

Une fois la database créée nous allons créer l'utilisateur et le mot de passe :

```
postgres=# create user zabbix with password 'admin';  
CREATE ROLE  
postgres=#
```

Ensuite attribuons tous les priviléges sur database zabbix à notre utilisateur zabbix :

```
postgres=# grant ALL privileges on database zabbix to zabbix;
GRANT
postgres=#
```

Nous allons maintenant modifier le fichier postgreSQL.conf pour qu'on puisse travailler sur la base de donnée depuis un autre hôte. Pour ce faire nous allons modifier la ligne listen_addresses qui est en localhost par défaut en *. Le 14 dans le chemin d'accès représente la version de notre système (le modifier selon le système utilisé).

```
postgres@srv-postgresql:~$ echo "listen_addresses='*'" >> /etc/postgresql/14/main/postgresql.conf
```

Nous devons ensuite changer les droits du fichier hba.conf (fichier qui permet d'autoriser ou refuser ce qu'un utilisateur a le droit de faire et à partir d'où.

```
postgres@srv-postgresql:~$ echo "host      zabbix      all      192.168.20.0/24      md5" >> /etc/postgresql/14/main/pg_hba.conf
postgres@srv-postgresql:~$
```

Ici nous avons donné tous les droits à tous les utilisateurs qui se trouve dans le sous réseau 192.168.20.0/24 de se connecter à la base de donnée zabbix (on pouvait filtrer et ne mettre que l'utilisateur zabbix) et celui-ci doit se connecter en entrant un mot de passe.

```
# Allow replication connections from localhost, by a user with the
# replication privilege.
local  replication  all                                     peer
host   replication  all          127.0.0.1/32            scram-sha-256
host   replication  all          ::1/128                scram-sha-256
host   zabbix       all          192.168.20.0/24        md5
postgres@srv-postgresql:~$
```

Après avoir fait tout ça on redémarre le service postgres : systemctl restart postgresql.

On peut faire un test en utilisant netstat -ntaup.

III.3 Installation et configuration du server zabbix

Une fois le socle de donnée stabilisé, le déploiement du serveur zabbix constitue l'étape pivot de l'infrastructure, agissant comme l'ordonnanceur central responsable de collecter, d'analyser et de distribuer les flux de surveillance.

Nous allons installer la version 6.0 de zabbix, pour ce faire nous commençons par télécharger le fichier .deb qui nous permettra ensuite d'installer zabbix.

```
root@srv-zabbix:~# wget https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.0-4+ubuntu22.04_all.deb
--2026-02-04 12:51:53--  https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.0-4+ubuntu22.04_all.deb
Resolving repo.zabbix.com (repo.zabbix.com)... 178.128.6.101, 2604:a880:2:d0::2062:d001
Connecting to repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3676 (3,6K) [application/octet-stream]
Saving to: ‘zabbix-release_6.0-4+ubuntu22.04_all.deb’

zabbix-release_6.0-4+ubuntu22.0 100%[=====] 3,59K --.-KB/s in 0s
2026-02-04 12:51:54 (1,52 GB/s) - ‘zabbix-release_6.0-4+ubuntu22.04_all.deb’ saved [3676/3676]

root@srv-zabbix:~# _
```

Nous installons ensuite les paquets de configuration des dépôts. La commande **dpkg -i zabbix-release_6.0-4+ubuntu22.04_all.deb** n'installe pas le logiciel de supervision lui-même, mais configure l'accès aux sources officielles.

Une fois le dépôt déclaré, la commande **apt update** est exécutée pour synchroniser le serveur avec les dépôts distants.

```
root@srv-zabbix:~# dpkg -i zabbix-release_6.0-4+ubuntu22.04_all.deb
Sélection du paquet zabbix-release précédemment désélectionné.
(Lecture de la base de données... 75400 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de zabbix-release_6.0-4+ubuntu22.04_all.deb ...
Dépaquetage de zabbix-release (1:6.0-4+ubuntu22.04) ...
Paramétrage de zabbix-release (1:6.0-4+ubuntu22.04) ...
root@srv-zabbix:~# apt update
Récception de :1 https://repo.zabbix.com/zabbix-agent2-plugins/1/ubuntu jammy InRelease [4 952 B]
Récception de :2 https://repo.zabbix.com/zabbix/6.0/ubuntu jammy InRelease [2 886 B]
Récception de :3 https://repo.zabbix.com/zabbix-agent2-plugins/1/ubuntu jammy/main Sources [1 002 B]
Récception de :4 https://repo.zabbix.com/zabbix-agent2-plugins/1/ubuntu jammy/main amd64 Packages [624 B]
Récception de :5 https://repo.zabbix.com/zabbix/6.0/ubuntu jammy/main Sources [41,8 kB]
Récception de :6 https://repo.zabbix.com/zabbix/6.0/ubuntu jammy/main amd64 Packages [116 kB]
Récception de :7 https://repo.zabbix.com/zabbix/6.0/ubuntu jammy/main all Packages [22,0 kB]
Récception de :8 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Atteint :9 http://archive.ubuntu.com/ubuntu jammy InRelease
Récception de :10 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Récception de :11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Récception de :12 http://archive.ubuntu.com/ubuntu jammy/main Translation-fr [486 kB]
Récception de :13 http://archive.ubuntu.com/ubuntu jammy/restricted Translation-fr [4 760 B]
Récception de :14 http://archive.ubuntu.com/ubuntu jammy/universe Translation-fr [3 564 kB]
Récception de :15 http://archive.ubuntu.com/ubuntu jammy/multiverse Translation-fr [93,5 kB]
4 722 ko réceptionnés en 9s (521 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
65 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
root@srv-zabbix:~# _
```

Nous pouvons maintenant faire l'installation de zabbix : **apt install -y -qq zabbix-server-pgsql zabbix-frontend-php php8.1-pgsql zabbix-nginx-conf zabbix-sql-scripts zabbix-agent postgresql-client**.

Cette commande constitue le cœur de l'installation. Elle demande au gestionnaire de paquets d'installer simultanément tous les composants nécessaires au fonctionnement du serveur de supervision.

zabbix-server-pgsql : Le moteur principal de Zabbix, configuré pour communiquer avec une base de données **PostgreSQL**. C'est lui qui calcule les alertes et gère les hôtes.

zabbix-frontend-php : Les fichiers de l'interface Web (écrits en PHP) qui permettent de visualiser vos données sur votre navigateur.

zabbix-nginx-conf : Le fichier de configuration pré-rempli pour le serveur Web **Nginx**, permettant de faire le lien entre le navigateur et l'interface PHP.

zabbix-sql-scripts : Contient les scripts SQL indispensables pour créer les tables de la base de données après l'installation.

zabbix-agent : Installe l'agent sur le serveur Zabbix lui-même pour qu'il puisse s'auto-surveiller (CPU, RAM, état des services).

php8.1-pgsql : Le connecteur qui permet au langage PHP (le front-end) de discuter avec votre base de données PostgreSQL.

postgresql-client : Les outils en ligne de commande pour manipuler la base de données depuis votre terminal (très utile pour importer le schéma SQL).

Une fois l'installation terminée on va s'intéresser à la configuration de notre zabbix

```
root@srv-zabbix:~# zcat /usr/share/zabbix-sql-scripts/postgresql/server.sql.gz | PGPASSWORD="admin" psql -h 192.168.20.7 -U zabbix zabbix
```

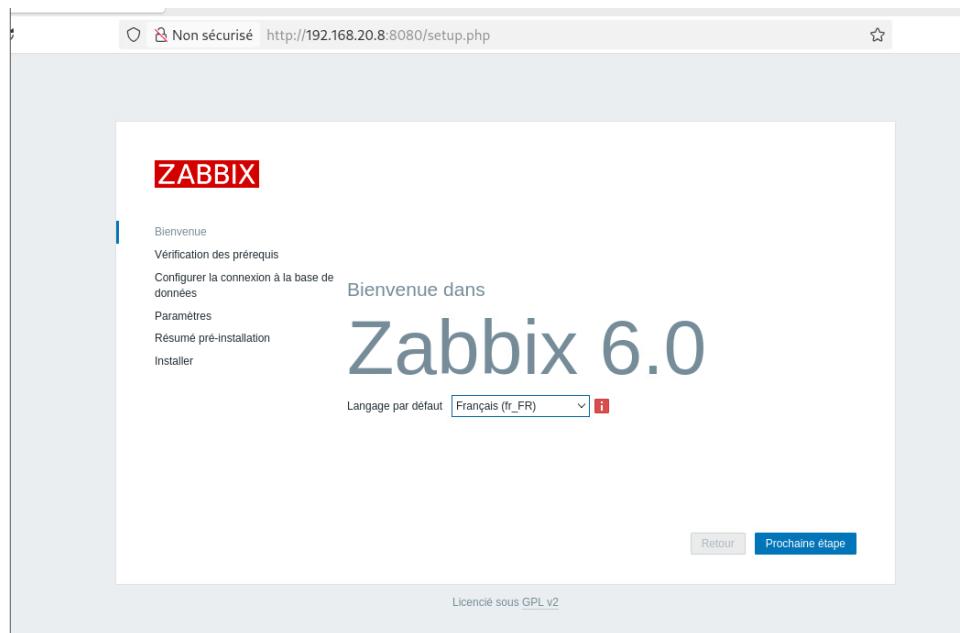
```
root@srv-zabbix:~# sed -i s/"#.*/server_name.*"/"server_name zabbix.training; "/g /etc/zabbix/nginx.conf
```

```
echo "DBPassword=admin" >> /etc/zabbix/zabbix_server.conf  
echo "DBHost=192.168.20.7" >> /etc/zabbix/zabbix_server.conf  
systemctl restart zabbix-server zabbix-agent nginx php8.1-fpm  
systemctl enable zabbix-server zabbix-agent nginx php8.1-fpm
```

Nous avons terminé avec l'installation cote serveur, Nous allons maintenant ouvrir le navigateur pour configurer frontend.

On entre l'adresse de notre serveur zabbix avec le port 8080.

Là on choisit la langue puis prochaine étape.



Là le système d'installation check l'ensemble des éléments et nous fait un récapitulatif.

Vérification des prérequis			
	Valeur actuelle	Requis	
Version de PHP	8.1.2-1ubuntu2.23	7.2.5	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	PostgreSQL		OK
bcmath pour PHP	sur		OK
mbstring pour PHP	sur		OK
Option PHP "mbstring.func_overload"	inatif	inatif	OK

C'est ici qu'on configure la connexion avec la base de données.

ZABBIX

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 PostgreSQL

Hôte base de données 192.168.20.7

Port de la base de données 5432 0 - utiliser le port par défaut

Nom de la base de données zabbix

Schéma de la base de données

Stocker les informations d'identification dans Texte brut Coffre HashiCorp

Utilisateur zabbix

Mot de passe *****

Chiffrement TLS de la base de données

[Retour](#)

[Prochaine étape](#)

Licencié sous GPL v2

Installation a été effectué avec succès.

ZABBIX

Installer

Bienvenue

Vérification des prérequis

Configurer la connexion à la base de données

Paramètres

Résumé pré-installation

Installer

Félicitations ! Vous avez installé l'interface Zabbix avec succès.

Fichier de configuration "conf/zabbix.conf.php" créé.

[Retour](#)

[Terminé](#)

Licencié sous GPL v2

III.4 Installation des agents

L'agent Zabbix est le composant léger installé sur les hôtes surveillés pour collecter localement les données système et applicatives. Cette section détaille les différentes méthodes de déploiement adaptées à la diversité de notre parc informatique.

III.4.1 Installation agent Linux

Le parc de serveurs Linux représentant le cœur de notre infrastructure de services, le déploiement de l'agent sur ces systèmes a été réalisé via les dépôts officiels afin de garantir une intégration parfaite avec le noyau Debian.

III.4.1.1 Installation agent passif

Dans cette configuration traditionnelle, le serveur Zabbix prend l'initiative de la connexion. Il interroge l'agent à intervalles réguliers pour récupérer les métriques demandées.

On va se rendre sur la machine cible puis installer et configurer l'agent zabbix 6 :

Installation des dependences.

```
root@srv-zabbix:~# wget -q https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.0-4+ubuntu22.04_all.deb
root@srv-zabbix:~# ls
snap/
zabbix-release_6.0-4+ubuntu22.04_all.deb  zabbix-release_6.0-4+ubuntu22.04_all.deb.1
root@srv-zabbix:~# _
```

```
root@srv-zabbix:~# dpkg -i zabbix-release_6.0-4+ubuntu22.04_all.deb
(Lecture de la base de données... 78160 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de zabbix-release_6.0-4+ubuntu22.04_all.deb ...
Dépaquetage de zabbix-release (1:6.0-4+ubuntu22.04) sur (1:6.0-4+ubuntu22.04) ...
Paramétrage de zabbix-release (1:6.0-4+ubuntu22.04) ...
root@srv-zabbix:~# _
```

```
root@srv-zabbix:~# apt update
Atteint :1 http://archive.ubuntu.com/ubuntu jammy InRelease
Atteint :2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Atteint :3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Atteint :4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Atteint :5 https://repo.zabbix.com/zabbix-agent2-plugins/1/ubuntu jammy InRelease
Atteint :6 https://repo.zabbix.com/zabbix/6.0/ubuntu jammy InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
65 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
root@srv-zabbix:~# _
```

Une fois les dépendances installées ut mise à jour on peut passer à l'installation de l'agent en utilisant la commande : **apt install zabbix-agent**

Une fois l'agent installer on doit le configurer pour qu'il puisse communiquer avec le server zabbix.

```
root@srv-zabbix:~# nano /etc/zabbix/zabbix_agentd.conf
```

L'adresse de server ici doit correspondre a l'IP de notre server zabbix ; on doit egalement se rassurer que la partie server Active soit vide.

```
# Server=
Server=192.168.20.8

### Option: ListenPort
#       Agent will listen on this port for connections from the server.
#
# Mandatory: no
# Range: 1024-32767

#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=

### Option: Hostname
#       List of comma delimited unique, case sensitive
#       Required for active checks and must match host
#       Value is acquired from HostnameItem if undefined
#
```

On confirme aussi le Hostname qui correspond au nom qu'on veut attribuer a notre agent.

```
# Hostname=
Hostname=Zabbix_server

### Option: HostnameItem
#       Item used for generating Hostname
#       Does not support UserParameters or
#
```

Une fois l'agent configurer on le redémarre pour prendre en compte les modifications.

```
root@srv-zabbix:~# systemctl restart zabbix-agent
root@srv-zabbix:~# _
```

Après avoir terminé la configuration de l'agent sur la machine cible, on se rend sur l'interface graphique pour intégrer cet agent afin d'obtenir nos premières métriques. Suivre le chemin suivant : Configuration > Hôtes > créer un hôte.

- On remplit le nom de l'hôte qui doit correspondre à celui indiqué lors de la configuration de l'agent.
- Ensuite on choisit un modèle généralement Linux by zabbix agent pour les agents passif linux.
- On sélectionne un groupe.
- On clique sur ajouter et choisit agent qui correspond à notre mode de découverte et on entre l'IP de la machine cible (celle qu'on souhaite surveiller).
- Cocher la case Activer puis Ajouter.

Nouvel hôte

Hôte IPMI Tags Macros Inventaire Chiffrement Table de correspondance

* Nom de l'hôte: zabbix_sever
Nom visible: zabbix_sever
Modèles: Linux by Zabbix agent
Groupes: Linux servers
Interfaces: Type: Adresse IP, Nom DNS: , Connexion à: IP, Port: 10050, Default: IP
Description:
Ajouter

Ajouter Annuler

Hôtes

Nom	Éléments	Déclencheurs	Graphiques	Découverte	Web	Interface	Proxy	Modèles	Etat	Disponibilité	Chiffrage
zabbix_server	61	10	9	3	192.168.20.8:10050	1	Linux by Zabbix agent	1	Actif	100%	256

0 sélectionné Activer Désactiver Exporter Modification collective Supprimer

Bien que simple à mettre en œuvre, ce mode passif peut être limité par les politiques de pare-feu restrictives, ce qui nous amène à considérer l'utilisation de l'agent en mode actif.

III.4.1.2 Installation agent actif

Contrairement au mode passif, l'agent actif initie lui-même la communication avec le serveur Zabbix pour demander sa liste de tâches à accomplir.

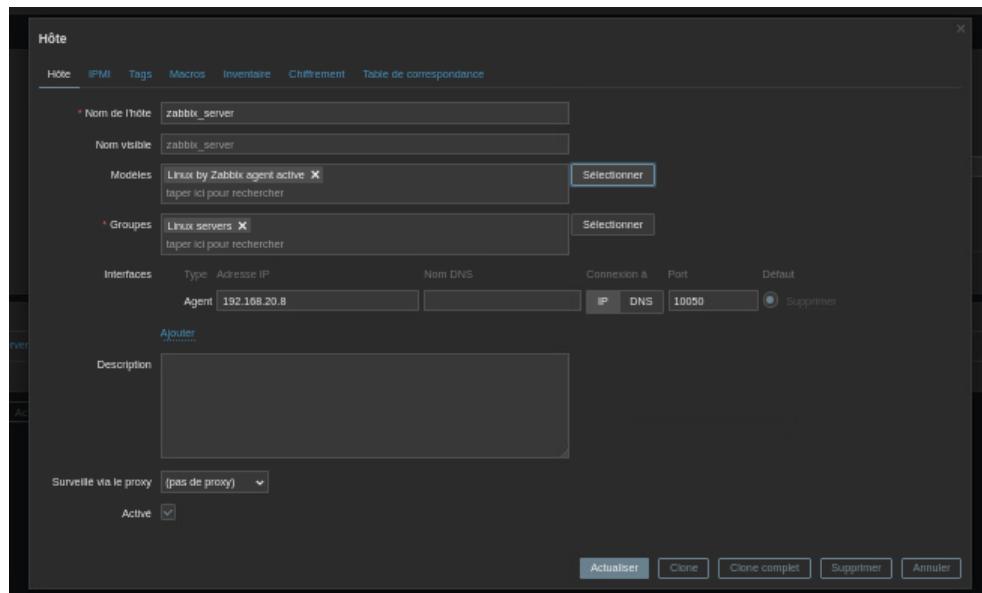
Une fois l'agent zabbix installé, la configuration et l'intégration de l'agent se font comme suit :

- Configurer cette machine cible est exactement comme celle d'un agent passif à la seule différence qu'on configure aussi la partie ServerActive :

```
### Option: ServerActive
#   Zabbix server/proxy address or cluster configuration to get active checks from.
#   Server/proxy address is IP address or DNS name and optional port separated by colon.
#   Cluster configuration is one or more server addresses separated by semicolon.
#   Multiple Zabbix servers/clusters and Zabbix proxies can be specified, separated by comma.
#   More than one Zabbix proxy should not be specified from each Zabbix server/cluster.
#   If Zabbix proxy is specified then Zabbix server/cluster for that proxy should not be specified.
#   Multiple comma-delimited addresses can be provided to use several independent Zabbix servers in parallel. Spaces are allowed between addresses.
#   If port is not specified, default port is used.
#   IPv6 addresses must be enclosed in square brackets if port for that host is specified.
#   If port is not specified, square brackets for IPv6 addresses are optional.
#   If this parameter is not specified, active checks are disabled.
# Example for Zabbix proxy:
#   ServerActive=127.0.0.1:10051
# Example for multiple servers:
#   ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
# Example for high availability:
#   ServerActive=zabbix.cluster.node1:zabbix.cluster.node2:20051;zabbix.cluster.node3
# Example for high availability with two clusters and one server:
#   ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster2.node2,zabbix.cluster2.node3
# Mandatory: no
# Default:
# ServerActive=
ServerActive=192.168.20.8
```

Une fois cette modification effectuée on redémarre le service zabbix-agent puis on se rend sur l'interface graphique.

- La seule modification à faire ici est le choix du modèle. Dans le cas d'un agent actif le modèle à choisir est « Linux by zabbix agent active »



NB : On remarque que le mode d'affichage de la disponibilité pour un agent actif est différent que pour un agent actif.

Une fois ces modes de collecte maîtrisés, le défi réside dans le déploiement à grande échelle, une problématique résolue par les mécanismes d'auto-découverte.

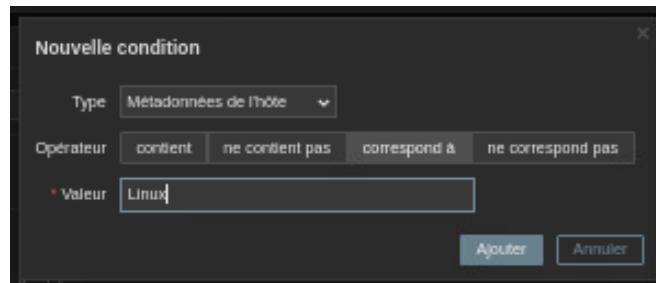
III.4.1.3 Installation agent Autoenregistrement

Pour éviter une gestion manuelle fastidieuse et sujette aux erreurs, Zabbix propose des fonctions d'automatisation permettant de détecter et d'intégrer de nouveaux hôtes dès leur apparition sur le réseau.

- On va commencer par faire la configuration cote interface graphique, suit le chemin Configuration > Actions > Action d'enregistrement automatique > Créer une action.
- On va nommer notre action puis on ajoute la condition suivante :

On a la possibilité de faire des check de trois types différents soit sur les host Name, les hosts Meta datas ou les proxys. Dans notre cas on choisit host métadonnées

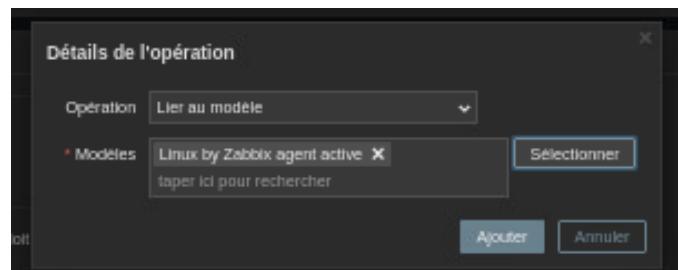
Lorsque notre agent va contacter le serveur il va fournir des métadonnées, le serveur va ensuite faire une liste d'action que nous avons configurer. On va ajouter la condition l'opérateur correspond à la valeur « Linux » (comme pour dire si le système d'exploitation correspond a linux)



Une fois L'action configurer on configurer l'opération :

Nous allons créer deux operation :

- Ajouter un hote.
- Lier au modele « Linux by zabbix agent active »
- Puis on ajoute et on Add notre action.



Une fois la configuration terminée cote interface graphique on doit créer un agent Autoregistration cote machine cible, on maintient la configuration de l'agent actif et on y ajoute les lignes suivantes :

Définir un hostnameItem et le HostMetadataItem.

```
### Option: HostnameItem
#      Item used for generating Hostname if it is undefined. Ignored if Hostname is defined.
#      Does not support UserParameters or aliases.
#
# Mandatory: no
# Default:
HostnameItem=system.hostname
```

```

### Option: HostMetadataItem
#      Optional parameter that defines an item used for getting host metadata.
#      Host metadata is used at host auto-registration process.
#      During an auto-registration request an agent will log a warning message if
#      the value returned by specified item is over limit of 255 characters.
#      This option is only used when HostMetadata is not defined.
#
# Mandatory: no
# Default:
HostMetadataItem=system.uname_

```

Une fois l'ensemble de ces opérations effectuées on retourne sur l'interface graphique.

The screenshot shows the Zabbix 'Hôtes' (Hosts) management screen. At the top, there are search and filter fields for 'Groupes d'hôtes', 'Modèles', and 'Nom'. Below these are filters for 'Surveillé par' (All, Server, Proxy), 'Proxy', and 'Tags'. The main table lists a single host:

Nom	Éléments	Déclencheurs	Graphiques	Découverte	Web	Interface	Proxy	Modèles	État	Disponibilité	Chiffrement	Info	Tags
zabbix_server	Éléments 07	Déclencheurs 26	Graphiques 14	Découverte 3	Web	192.168.20.8:10050		Linux by Zabbix agent active	Active	ZBX		Aucun	

At the bottom of the table, there are buttons for 'Activer' (Enable), 'Désactiver' (Disable), 'Exporter' (Export), 'Modification collective' (Bulk modification), and 'Supprimer' (Delete). A note at the bottom right says 'Affichage de 1 sur 1 trouvés'.

L'enregistrement s'est fait automatiquement.

NB : Pour pouvoir configurer l'Autoenregistrement, l'agent doit être configuré en actif.

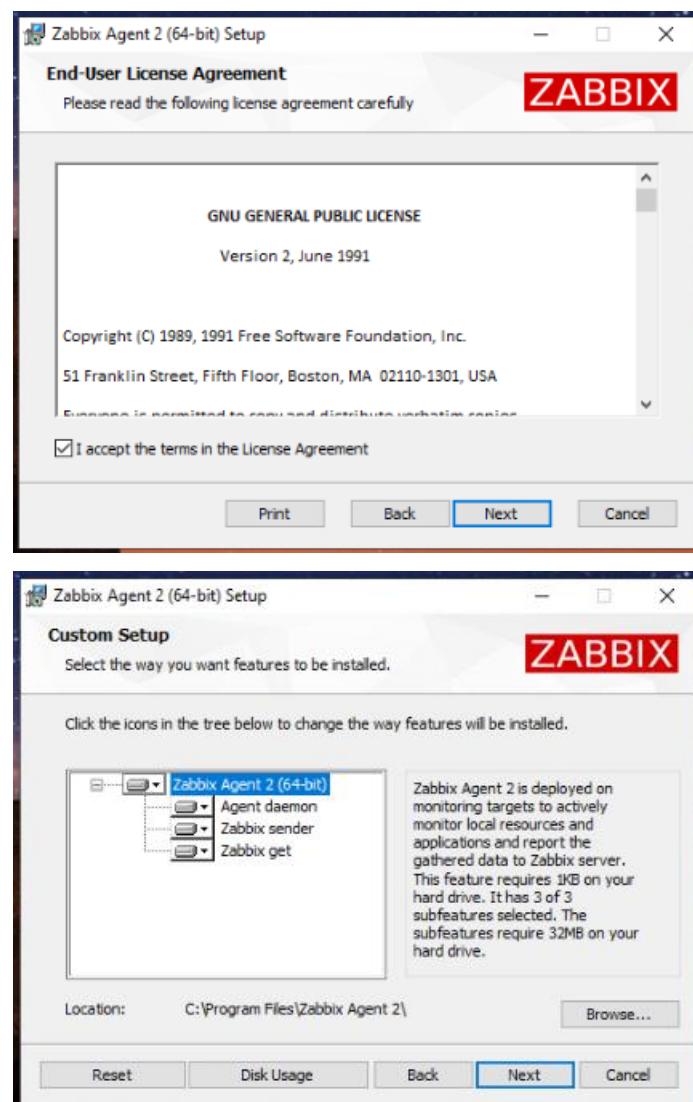
Si l'environnement Linux constitue une partie majeure de l'infrastructure, la supervision du SI ne serait pas complète sans l'intégration des systèmes Windows.

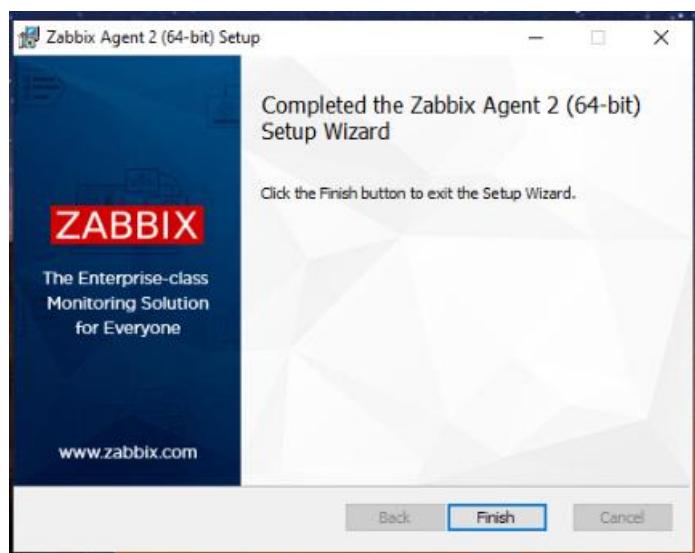
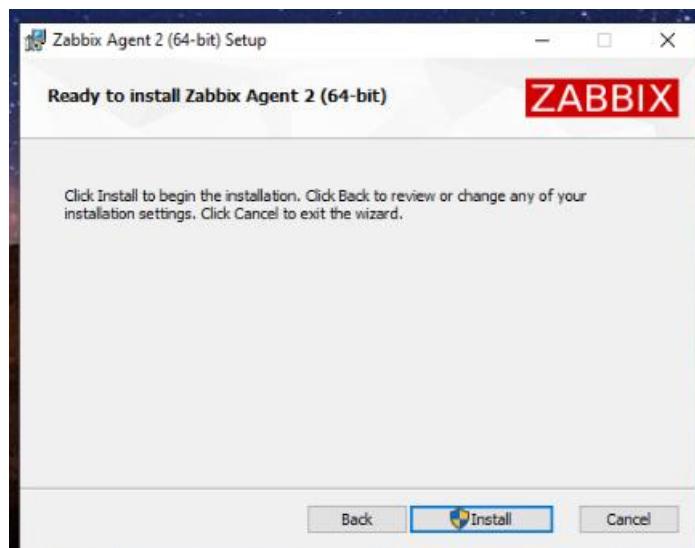
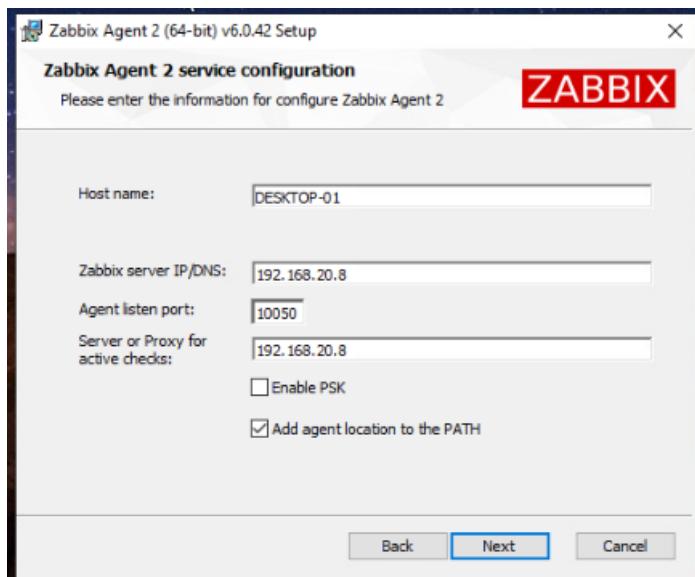
III.4.2 Installation agent Windows

La surveillance des environnements Windows nécessite l'installation l'agent sous forme de service système, permettant de remonter les compteurs de performance spécifiques à l'écosystème Microsoft (WMI, journaux d'événements).

Il faut tout d'abord télécharger la version de l'agent correspondant sur le site :
https://www.zabbix.com/download_agents

Ensuite suivre les étapes d'installation et d'intégration suivantes :





Hôtes

Nouvel hôte

Hôte IPMI Tags Macros Inventaire Chiffrement Table de correspondance

* Nom de l'hôte: DESKTOP-01
Nom visible: DESKTOP-01
Modèles: Windows by Zabbix agent X
taper ID pour rechercher
* Groupes: Virtual machines X
taper ID pour rechercher

Interfaces Type: Adresse IP Nom DNS Connexion à Port Default
Agent: 192.168.10.2 IP DNS 10050 Supprimer

Ajouter Description:

Surveille via le proxy: (pas de proxy) Active: ✓

Ajouter Annuler

ZABBIX Hôtes

Créer un hôte

Surveillance Hôtes Dernières données Services Inventaire Rapports Configuration Administration Support Intégrations Aide Paramètres utilisateur Déconnexion

Nom: Groupes d'hôtes: IP: DNS: Port: Sévérité: Non classé Avertissement Haut Information Moyen Desastre

État: Tous Actif Désactivé Tags: ET OU tag: Content: valeur: Supprimer Ajouter... Afficher les hôtes en maintenance: ✓ Afficher les problèmes supprimés: □

Enregistrer sous: Appliquer Réinitialiser

Nom	Interface	Disponibilité	Tags	Etat	Dernières données	Problèmes	Graphiques	Tableaux de bord	Web
DESKTOP-01	192.168.10.2:10050	ZBX	class:os target:windows	Actif	Dernières données 41	Problèmes 0	Graphiques 5	Tableaux de bord 2	Web
zabbix_server	192.168.20.8:10050	ZBX	class:os target:linux	Actif	Dernières données 68	Problèmes 0	Graphiques 14	Tableaux de bord 2	Web

Affichage de 2 sur 2 trouvés

III.5 Cr éation d'une m étrique personnalisée

Bien que Zabbix propose des milliers de m étriques par d éfaut, la r éalit é du terrain exige parfois de surveiller des éléments sp écifiques à une organisation. Pour r épondre à ce besoin, nous avons mis en œuvre l'utilisation des **UserParameters**, une fonctionnalité permettant d'étendre les capacités de l'agent en lui injectant des commandes syst ème personnalisées.

Nous avons cr éé une m étrique qui permet de surveiller l'état du serveur zabbix. Elle compte le nombre de processus trappers en actifs sur le serveur zabbix.

Cette m étrique a été d éclarée dans un fichier d édié sous /etc/zabbix/zabbix_agentd.d/

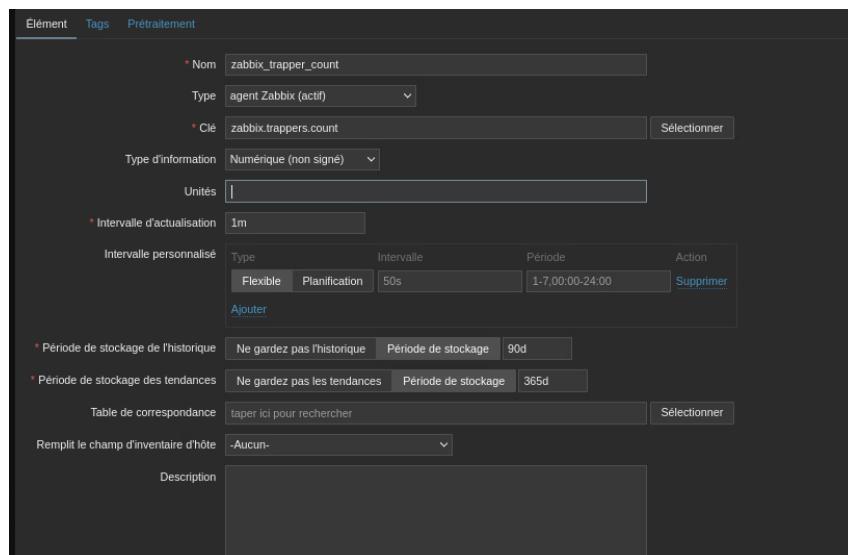
```
root@srv-zabbix:~# nano /etc/zabbix/zabbix_agentd.d/zabbix_health.conf
GNU nano 6.2                               /etc/zabbix/zabbix_agentd.d/zabbix_health.conf
UserParameter=zabbix.trappers.count,ps aux | grep -c "zabbix_server:trapper"
```

Grâce à la commande zabbix_agentd -R userparameter_reload, la nouvelle m étrique a été prise en compte instantanément par l'agent sans interruption de service.

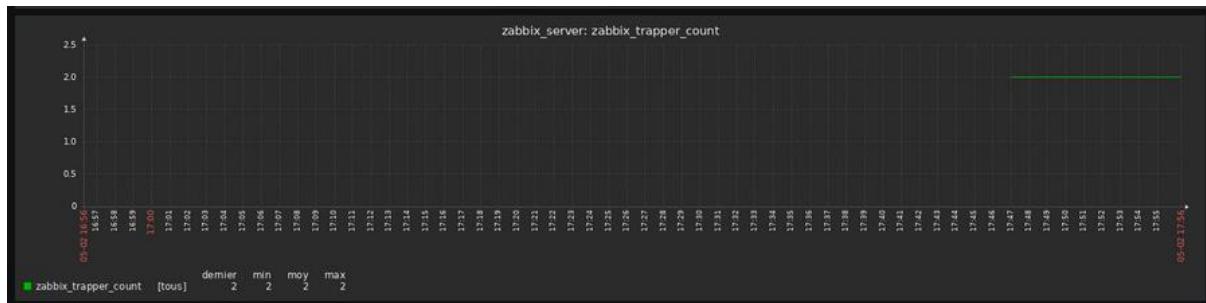
```
root@srv-zabbix:~# zabbix_agentd -R userparameter_reload
zabbix_agentd [8061]: Warning: both Hostname and HostnameItem defined, using [zabbix_server]
zabbix_agentd [8061]: command sent successfully
root@srv-zabbix:~#
```

Une fois la m étrique cr ée su le serveur cible on doit l'ajouter sur l'interface graphique en suivant le chemin suivant : Configuration > Hôte > Elément > cr éer un élément.

On configure notre élément, la clé doit être identique au nom du UserParameter



Notre métrique a bien été créé et est fonctionnelle.



Conclusion

Ce projet de mise en œuvre d'une infrastructure de monitoring sous Zabbix 6.0 a permis de simuler un environnement de production réel, de l'installation de la base de données PostgreSQL jusqu'à la création de sondes personnalisées.

Compétences acquises

La réalisation de ce travail m'a permis de consolider des expertises clés en administration système :

- **Administration Système Linux** : Maîtrise de la gestion des paquets (dpkg, apt), configuration de services réseau (Nginx) et gestion des droits d'exécution.
- **Gestion de Bases de Données** : Installation, sécurisation et maintenance d'une instance PostgreSQL dédiée à la supervision.
- **Architecture de Supervision** : Compréhension profonde des flux de données entre agents (actifs/passifs), proxies et serveur central.
- **Automatisation et Agilité** : Utilisation de l'auto-découverte pour optimiser le déploiement d'un parc informatique à grande échelle.

En conclusion, ce projet m'offre une base solide pour atteindre mon objectif professionnel : devenir un expert capable de bâtir des infrastructures résilientes et visibles, tout en développant une expertise technique.