



ZABBIX

DE LA SUPERVISION RESEAUX

BTS S.I.O. OPTION S.I.S.R. -- 2025 - 2026

MR MICHELE MASTROGIACOMO

Michele M.

Contexte et objectifs	PAG. 2
<ul style="list-style-type: none">• Présentation de la supervision• Pourquoi utiliser Zabbix• Objectif : surveiller des services critiques	
Architecture de la solution	PAG. 3
<ul style="list-style-type: none">• Description des machines• Choix techniques (Debian, IP fixes...)	
Installation des systèmes	PAG. 4
<ul style="list-style-type: none">• Installation de **Debian 13• Configuration réseau	
Installation et configuration de Zabbix	PAG. 6
<ul style="list-style-type: none">• Serveur + base de données• Interface web	
Déploiement des agents	PAG. 9
<ul style="list-style-type: none">• Installation agent• Vérification communication	
Mise en supervision	PAG. 11
<ul style="list-style-type: none">• Ajout des hôtes• Templates• Surveillance ressources + services	
Mise en place d'alertes	PAG. 13
<ul style="list-style-type: none">• Triggers• Notifications	
Tests et validation	PAG. 15
<ul style="list-style-type: none">• Simulation panne• Analyse des résultats	
Sécurisation et limites	PAG. 17
<ul style="list-style-type: none">• Firewall• Limites du TP	
Conclusion	PAG. 18

Contexte et objectifs

La supervision des systèmes et des services est essentielle afin de garantir la disponibilité, la performance et la sécurité du système d'information.

Les entreprises doivent être capables de détecter rapidement les anomalies, les pannes ou les dégradations de service afin d'intervenir efficacement. Pour répondre à ce besoin, des outils de supervision sont mis en place.

Dans ce contexte, j'ai choisi d'utiliser **Zabbix**, une solution open-source de supervision largement utilisée en entreprise. Cet outil permet de surveiller en temps réel l'état des serveurs, des services réseau et des équipements informatiques.

Objectifs du TP

Ce travail pratique a pour objectif de :

- Mettre en place une solution de supervision complète
- Installer et configurer un serveur de supervision
- Déployer des agents sur des machines clientes
- Surveiller les ressources système (CPU, RAM, disque)
- Superviser des services (SSH, etc.)
- Mettre en place des alertes en cas de problème
- Tester le bon fonctionnement du système de supervision

L'ensemble de ce TP est réalisé dans un environnement virtualisé à l'aide de **VMware Workstation Pro**, permettant de simuler une infrastructure réseau réaliste.

Architecture de la solution

L'infrastructure mise en place repose sur un environnement virtualisé sous **VMware Workstation Pro**, permettant de simuler un réseau d'entreprise.

Machines utilisées

Nom machine	Système	Rôle
srv-zabbix	Debian 13	Serveur de supervision
client-linux	Lubuntu (Linux)	Machine supervisée
client-windows	Windows 10/11	Machine cliente supervisée

Choix techniques

- Utilisation de distributions Linux pour la légèreté et la stabilité
- Utilisation de **Zabbix** comme solution de supervision
- Réseau isolé (Host-Only) pour éviter les conflits avec d'autre Machine
- Attribution d'adresses IP statiques pour garantir la communication entre les machines

Plan d'adressage

Machine	Adresse IP
srv-zabbix	192.168.56.10
client-linux	192.168.56.20
client-windows	192.168.56.30

Installation des systèmes

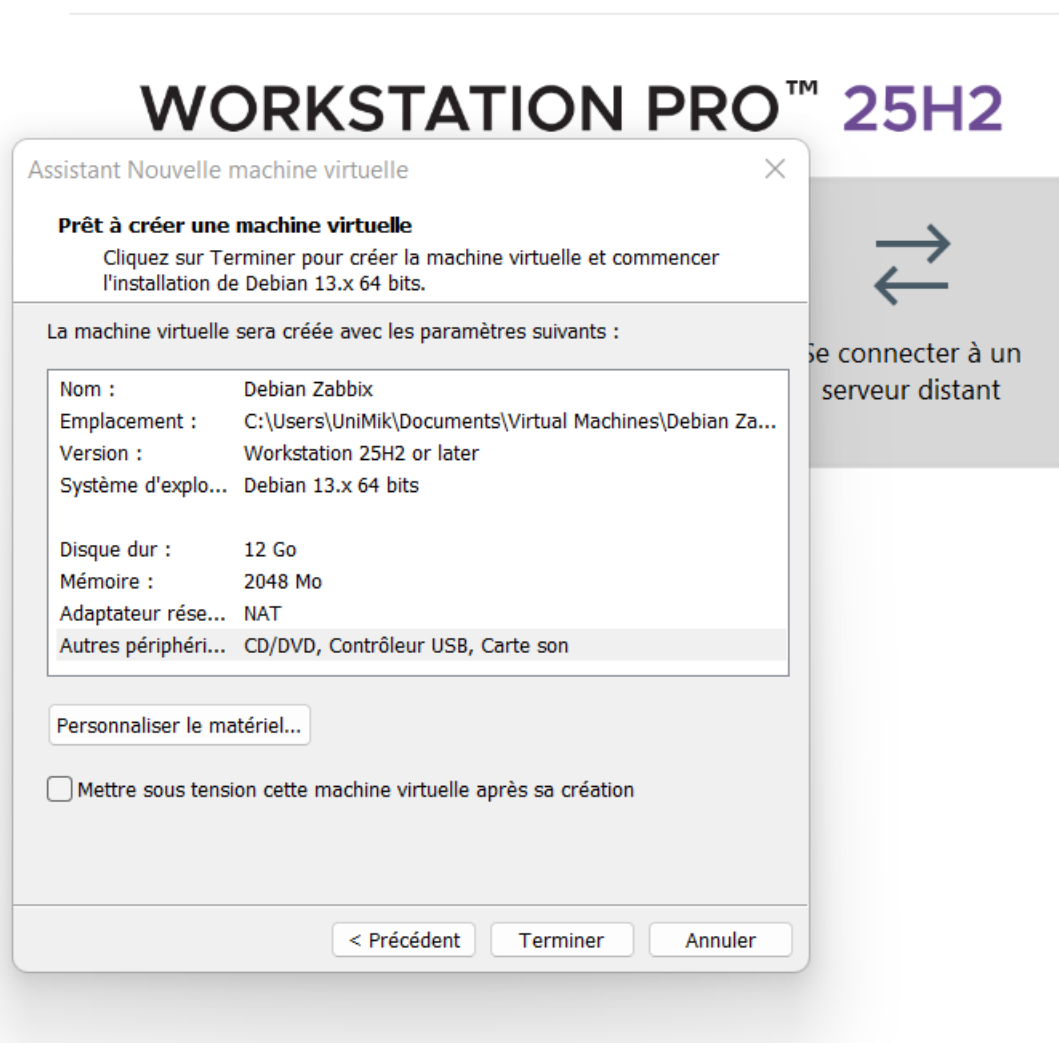
Plusieurs machines virtuelles ont été déployées afin de mettre en place une infrastructure de supervision basée sur Zabbix.

Les machines ont été créées à l'aide de **VMware Workstation Pro** avec des configurations adaptées afin de limiter la consommation de ressources.

Trois machines virtuelles ont été mises en place :

- Un serveur Debian hébergeant le serveur de supervision
- Un client Linux (Lubuntu)
- Un client Windows

Image 1 : Création de la machine virtuelle Debian Server Zabbix sous VMware



Installation du Serveur Debian 13

Une installation de Debian a été réalisée avec des options choisies lors de l'installation tel que serveur SSH.

Image 2: Sélection de paquet à installer lors de l'installation Debian



Configuration réseau

Un adressage IP statique a été mis en place afin d'assurer une communication fiable entre les machines. Chaque machine a été renommée afin de faciliter son identification dans le réseau.

Configuration du client Linux

Le client Linux (Ubuntu / Lubuntu) a été configuré avec :

- Une IP statique (192.168.56.20)
- Un accès réseau fonctionnel
- Le service SSH actif

Configuration du client Windows

Le client Windows a été configuré avec :

- Une adresse IP statique (192.168.56.30)
- Une connectivité réseau avec le serveur Zabbix

Installation et configuration de Zabbix

Dans cette partie, nous allons mettre en place le serveur de supervision **Zabbix** sur la machine Debian. L'installation de la solution de supervision **Zabbix** a été réalisée en s'appuyant sur la documentation officielle fournie par l'éditeur.

Cette documentation garantit une installation conforme aux bonnes pratiques et adaptée au système d'exploitation utilisé (Debian 13 avec base de données MySQL et serveur web Apache).

Des ajustements ont toutefois été réalisés afin d'adapter la configuration à l'environnement virtualisé mis en place dans ce TP.

Documentation officielle Zabbix :

<https://www.zabbix.com/download>

Zabbix repose sur plusieurs composants :

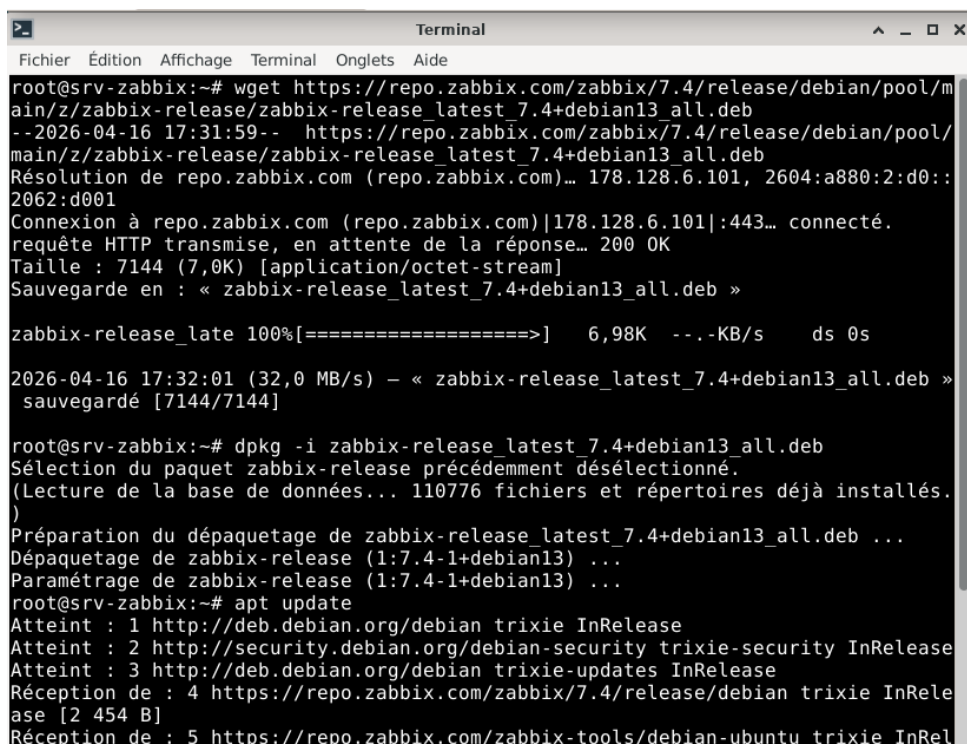
- Un serveur de supervision
- Une base de données
- Une interface web accessible via navigateur

Avant toute installation, le système est mis à jour afin d'assurer la compatibilité des paquets avec la commande bien connu : **apt update && apt upgrade**

Installation des prérequis

J'ai téléchargé le paquet d'installation Zabbix depuis son site officiel, installer le paquet téléchargé. Elle ajoute le dépôt Zabbix au système, et mets à jour la liste des paquets disponibles.

Image 3: Ajout dépôts paquet Zabbix et mis à jour



```
root@srv-zabbix:~# wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_latest_7.4+debian13_all.deb
--2026-04-16 17:31:59-- https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_latest_7.4+debian13_all.deb
Résolution de repo.zabbix.com (repo.zabbix.com)... 178.128.6.101, 2604:a880:2:d0::2062:d001
Connexion à repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 7144 (7,0K) [application/octet-stream]
Sauvegarde en : « zabbix-release_latest_7.4+debian13_all.deb »

zabbix-release_late 100%[=====] 6,98K --.-KB/s ds 0s

2026-04-16 17:32:01 (32,0 MB/s) – « zabbix-release_latest_7.4+debian13_all.deb »
sauvegardé [7144/7144]

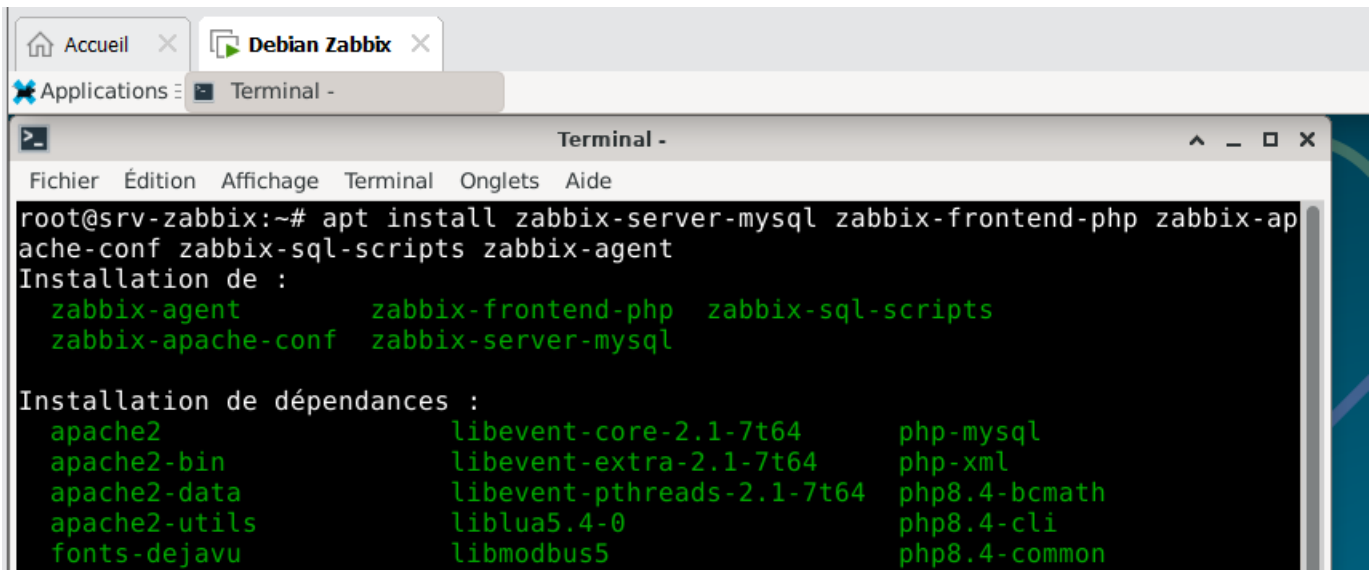
root@srv-zabbix:~# dpkg -i zabbix-release_latest_7.4+debian13_all.deb
Sélection du paquet zabbix-release précédemment désélectionné.
(Lecture de la base de données... 110776 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de zabbix-release_latest_7.4+debian13_all.deb ...
Dépaquetage de zabbix-release (1:7.4-1+debian13) ...
Paramétrage de zabbix-release (1:7.4-1+debian13) ...
root@srv-zabbix:~# apt update
Atteint : 1 http://deb.debian.org/debian trixie InRelease
Atteint : 2 http://security.debian.org/debian-security trixie-security InRelease
Atteint : 3 http://deb.debian.org/debian trixie-updates InRelease
Réception de : 4 https://repo.zabbix.com/zabbix/7.4/release/debian trixie InRelease [2 454 B]
Réception de : 5 https://repo.zabbix.com/zabbix-tools/debian-ubuntu trixie InRelease
```

Installation de Zabbix

Les services nécessaires au fonctionnement de Zabbix sont installés ainsi les dépendances :

- Serveur web (Apache), Base de données (MySQL), PHP

Image 4 : Installation du serveur Zabbix et de ses composants



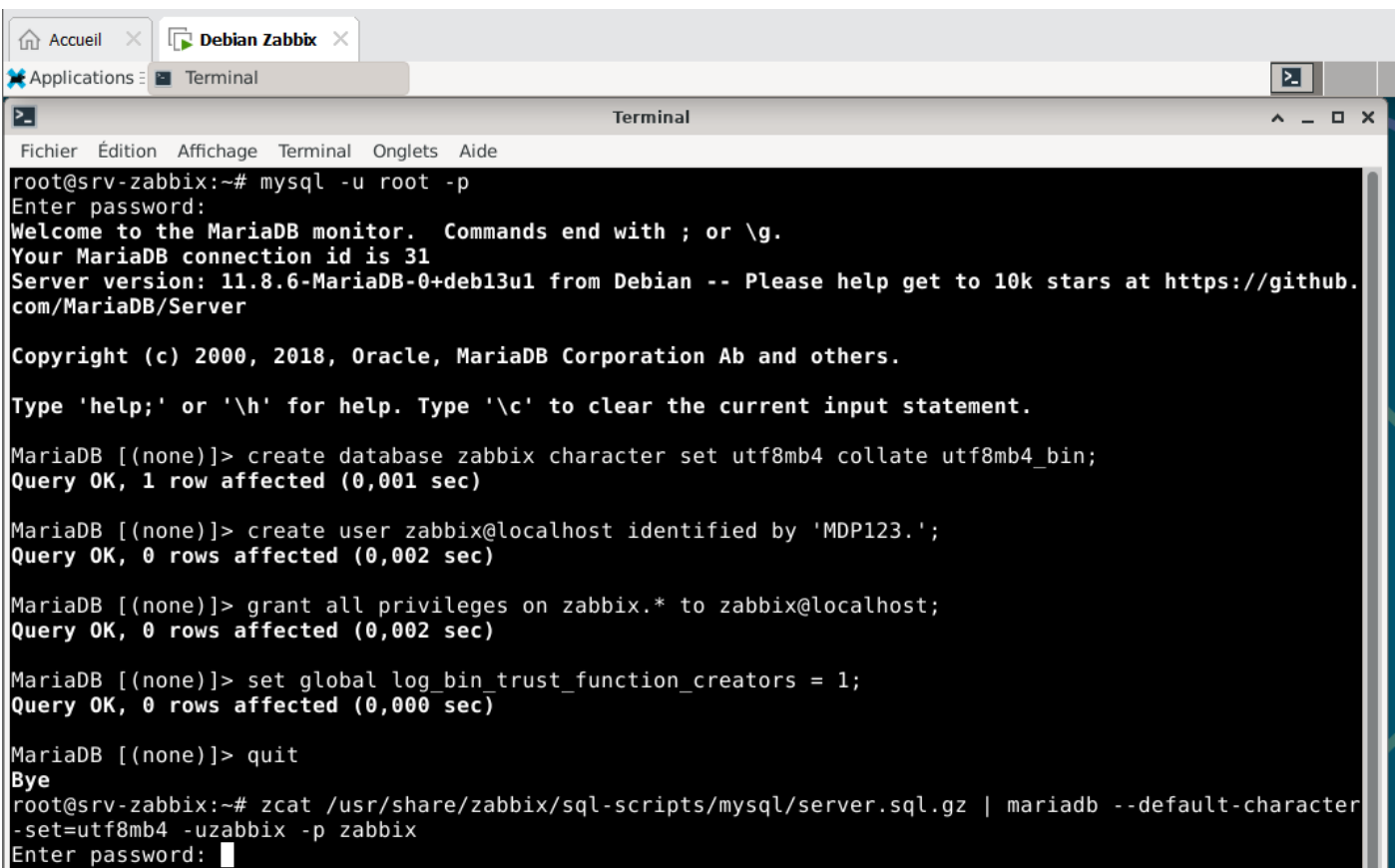
```
root@srv-zabbix:~# apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent
Installation de :
  zabbix-agent      zabbix-frontend-php  zabbix-sql-scripts
  zabbix-apache-conf zabbix-server-mysql

Installation de dépendances :
  apache2          libevent-core-2.1-7t64  php-mysql
  apache2-bin      libevent-extra-2.1-7t64  php-xml
  apache2-data     libevent-pthreads-2.1-7t64  php8.4-bcmath
  apache2-utils    liblua5.4-0             php8.4-cli
  fonts-dejavu     libmodbus5              php8.4-common
```

Configuration de la base de données et importation

Maintenant je crée la base de données MySQL, je donne l'utilisateur et un mot de passe de connexion, en ce cas MDP123. Et je l'importe avec ce même mot de passe

Image 5: Création de la base de données Zabbix et importation avec mot de passe MDP123.



```
root@srv-zabbix:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 11.8.6-MariaDB-0+deb13u1 from Debian -- Please help get to 10k stars at https://github.com/MariaDB/Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database zabbix character set utf8mb4 collate utf8mb4_bin;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> create user zabbix@localhost identified by 'MDP123.';
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@localhost;
Query OK, 0 rows affected (0,002 sec)

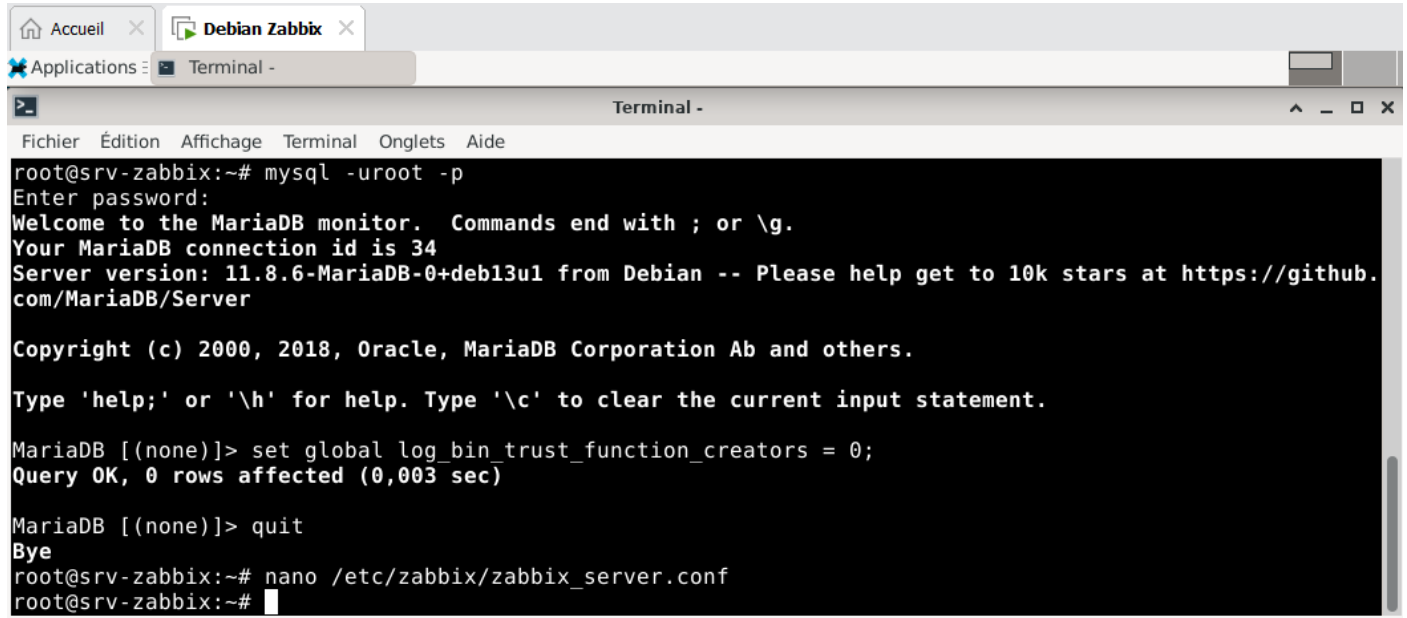
MariaDB [(none)]> set global log_bin_trust_function_creators = 1;
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]> quit
Bye
root@srv-zabbix:~# zcat /usr/share/zabbix/sql-scripts/mysql/server.sql.gz | mariadb --default-character-set=utf8mb4 -uzabbix -p zabbix
Enter password: █
```

Configuration du serveur Zabbix

Je désactive « `log_bin_trust_function_creators` » après l'importation du database, et j'ajoute le mot de passe choisi pour la base de donnée sur le fichiers configuration server de Zabbix.

Image 6: Configuration connexion base de données



```
root@srv-zabbix:~# mysql -uroot -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 34
Server version: 11.8.6-MariaDB-0+deb13u1 from Debian -- Please help get to 10k stars at https://github.com/MariaDB/Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

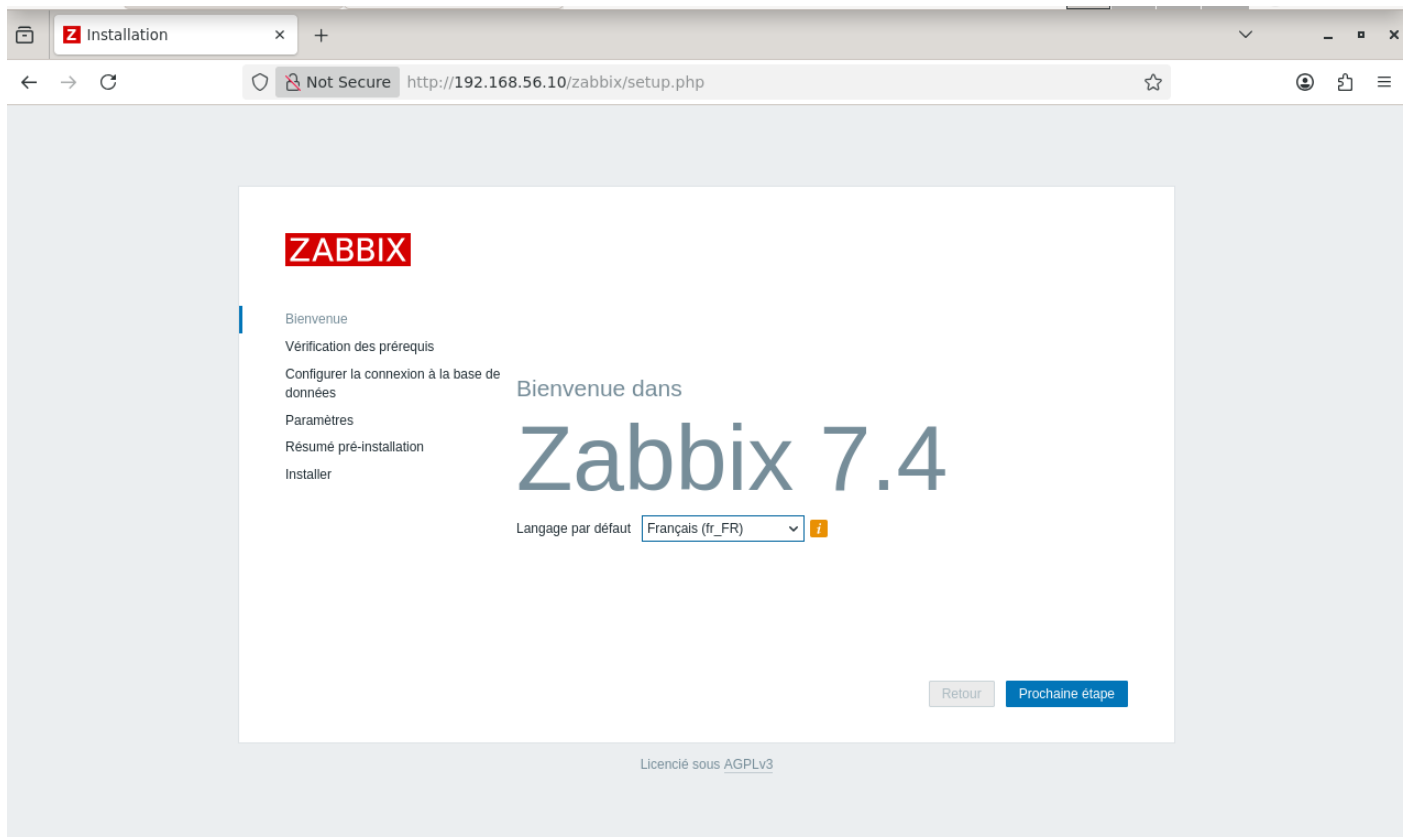
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> set global log_bin_trust_function_creators = 0;
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> quit
Bye
root@srv-zabbix:~# nano /etc/zabbix/zabbix_server.conf
root@srv-zabbix:~#
```

Enfin après le redémarrage des services j'ai l'accès à l'interface web pour terminer l'installation, renseignant les information pour accès Database et nom serveur

Image 7 : Accès à l'interface web de Zabbix



Déploiement des agents

Afin de superviser les machines du réseau, il est nécessaire d'installer un agent Zabbix sur chaque machine cliente. Les agents permettent de collecter les données système des machines et de les transmettre au serveur Zabbix.

Cet agent permet de collecter des informations telles que :

- l'utilisation du processeur, la mémoire, l'espace disque, l'état des services, etc..

Ces données sont ensuite envoyées au serveur de supervision.

Installation agent Linux

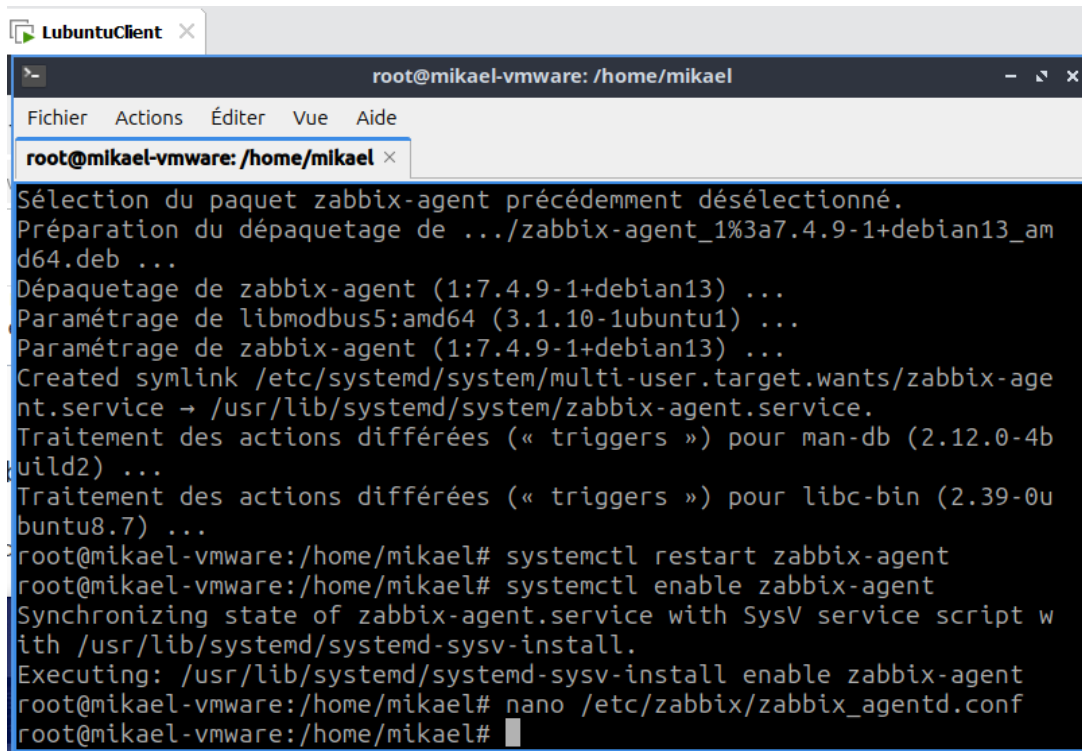
Chose importante que faut bien donner le lien du dépôt packages de Zabbix :

```
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
```

J'ai pu faire installation avec la commande simplifiée : `apt install zabbix-agent`

Ensuite j'ai du modifier le fichier de configuration de l'agent afin que puisse communiquer correctement avec le serveur Zabbix a l'adresse 192.168.56.10 sur le fichier `zabbix_agentd.conf`

Image 8 : Installation Agent sur machine Lubuntu et configuration

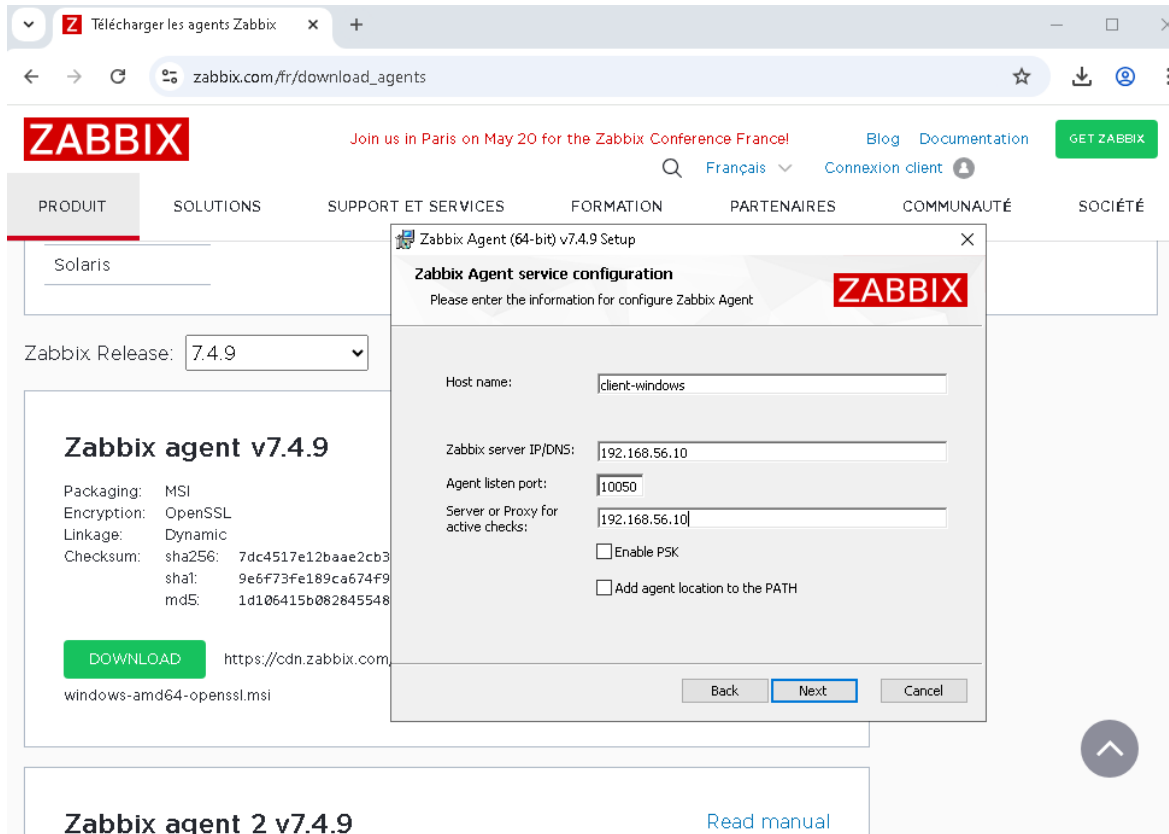


```
root@mikael-vmware: /home/mikael  
Fichier Actions Éditer Vue Aide  
root@mikael-vmware: /home/mikael  
Sélection du paquet zabbix-agent précédemment désélectionné.  
Préparation du dépaquetage de ../zabbix-agent_1%3a7.4.9-1+debian13_am  
d64.deb ...  
Dépaquetage de zabbix-agent (1:7.4.9-1+debian13) ...  
Paramétrage de libmodbus5:amd64 (3.1.10-1ubuntu1) ...  
Paramétrage de zabbix-agent (1:7.4.9-1+debian13) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-age  
nt.service → /usr/lib/systemd/system/zabbix-agent.service.  
Traitement des actions différées (« triggers ») pour man-db (2.12.0-4b  
uild2) ...  
Traitement des actions différées (« triggers ») pour libc-bin (2.39-0u  
buntu8.7) ...  
root@mikael-vmware:/home/mikael# systemctl restart zabbix-agent  
root@mikael-vmware:/home/mikael# systemctl enable zabbix-agent  
Synchronizing state of zabbix-agent.service with SysV service script w  
ith /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable zabbix-agent  
root@mikael-vmware:/home/mikael# nano /etc/zabbix/zabbix_agentd.conf  
root@mikael-vmware:/home/mikael#
```

Installation agent Windows

La mise en place d'un Agent Zabbix sur Windows est simplifiée avec l'installation d'un *.msi, après la configuration des adresse IP du serveur Zabbix j'ai pu terminer l'installation rapidement.

Image 9 : Configuration Agent Zabbix sur Windows



Vérification de la communication

Une fois les agents installés et configurés, la communication avec le serveur Zabbix a été vérifiée. Le bon fonctionnement est confirmé lorsque les machines apparaissent comme disponibles dans l'interface web de Zabbix. D'ailleurs un ping peut bien rendre l'idée que la connexion est bonne.

Image 10 : Ping des hosts client-linux et client-windows sur serveur Zabbix



Mise en supervision

Après le déploiement des agents, les machines ont été ajoutées dans l'interface Zabbix afin d'activer la supervision.

Ajout des hôtes

Chaque machine a été ajoutée manuellement dans Zabbix avec son nom d'hôte et son adresse IP.

Image 10 : Ajout des Hôtes

The screenshot shows the Zabbix web interface for adding a new host. The browser address bar indicates the URL: `http://192.168.56.10/zabbix/zabbix.php?name=&ip=&dns=&port=&status=-1&evaltype=0&tags[0][tag]=&to`. The page title is "Hôtes".

The "Nouvel hôte" form includes the following fields and options:

- Nom de l'hôte:** client-linux
- Nom visible:** client-linux
- Modèles:** taper ici pour rechercher (with a "Sélectionner" button)
- Groupes d'hôtes:** Virtual machines (with a "Sélectionner" button)
- Interfaces:** A table with columns: Type, adresse IP, Nom DNS, Connexion à, Port, and Défaut.

Type	adresse IP	Nom DNS	Connexion à	Port	Défaut
Agent	192.168.56.20		IP DNS	10050	<input checked="" type="radio"/> Supprimer
- Description:** A text area with the label "Ajouter".
- Surveillé par:** Server, Proxy, Groupe de proxy (radio buttons)
- Activé:**

Buttons at the bottom right: "Ajouter" and "Annuler".

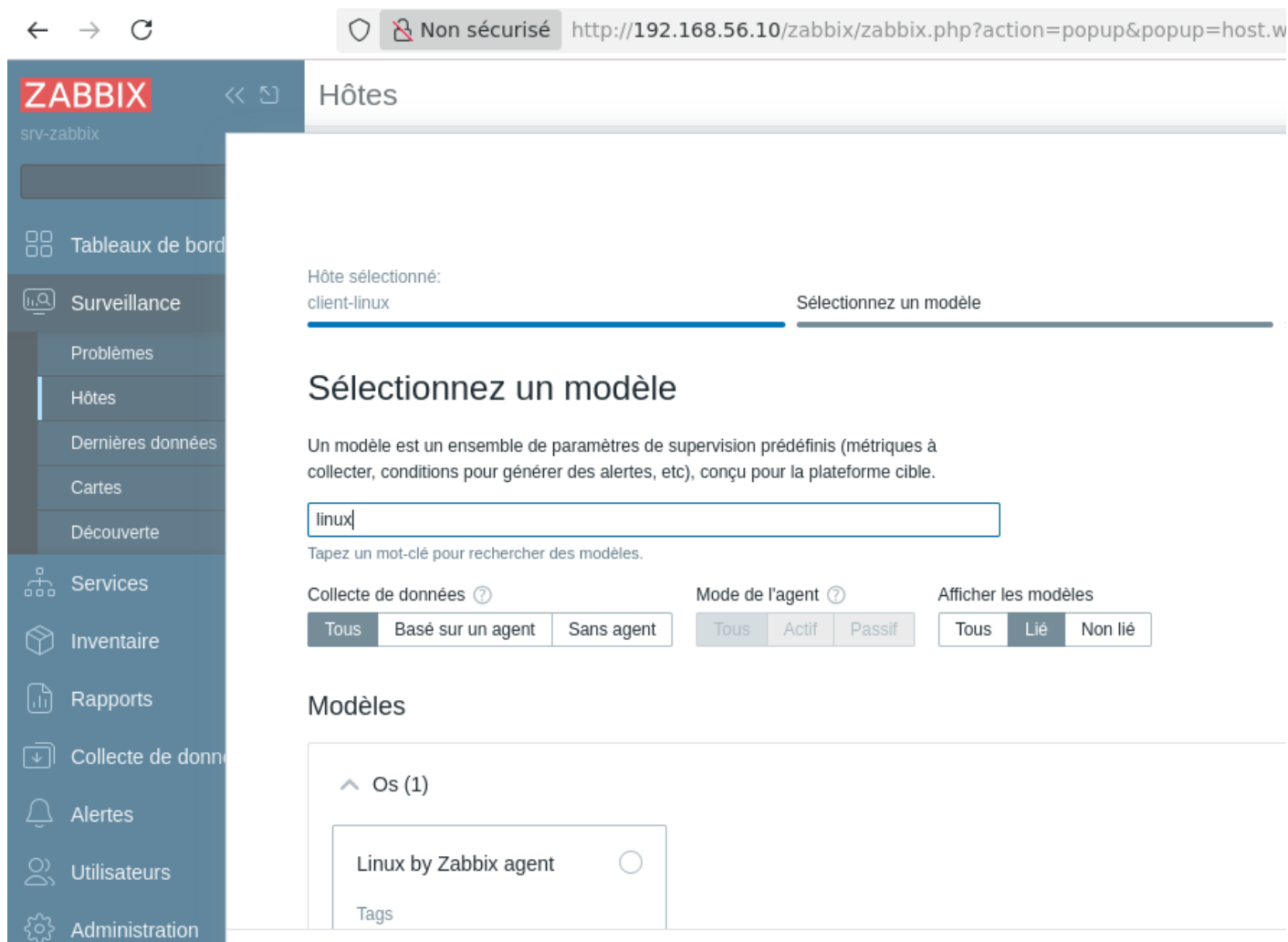
Association des templates

Des templates adaptés ont été associés à chaque machine :

Template Linux pour le client Ubuntu, Template Windows pour le client Windows.

Ces templates permettent de superviser automatiquement les ressources système, plusieurs métriques sont automatiquement collectées comme : l'utilisation CPU, utilisation de la mémoire RAM, de l'espace disque, la charge du système et bien d'autre option peuvent être rajoute.

Image 11 : Ajout d'un template

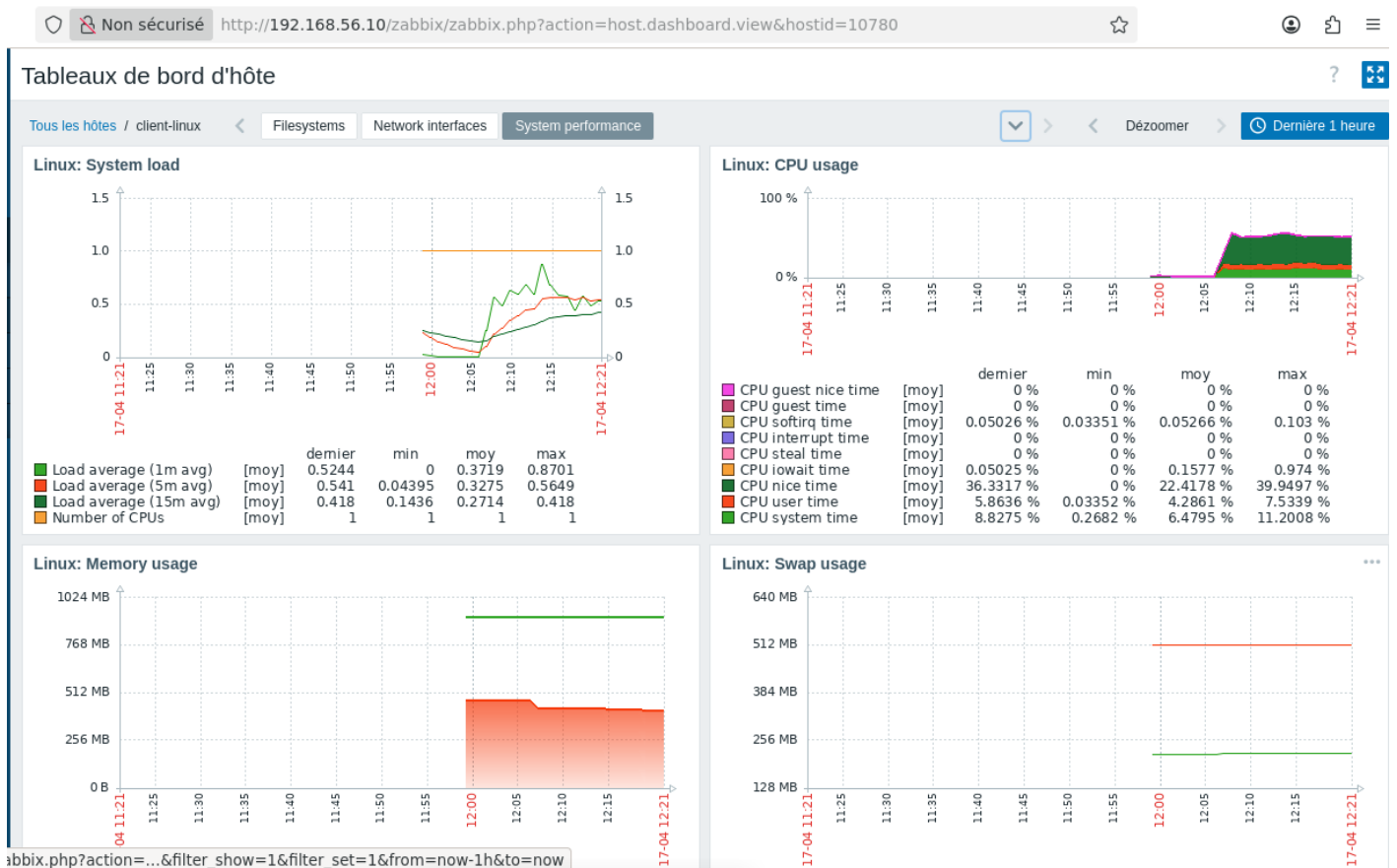


The screenshot shows the Zabbix web interface. The browser address bar indicates the URL: `http://192.168.56.10/zabbix/zabbix.php?action=popup&popup=host.w`. The page title is "Hôtes". The left sidebar contains navigation options: "Tableaux de bord", "Surveillance", "Problèmes", "Hôtes", "Dernières données", "Cartes", "Découverte", "Services", "Inventaire", "Rapports", "Collecte de données", "Alertes", "Utilisateurs", and "Administration". The main content area shows "Hôte sélectionné: client-linux" and "Sélectionnez un modèle". Below this, there is a search box containing "linux" and a prompt "Tapez un mot-clé pour rechercher des modèles.". There are three filter sections: "Collecte de données" with buttons "Tous", "Basé sur un agent", and "Sans agent"; "Mode de l'agent" with buttons "Tous", "Actif", and "Passif"; and "Afficher les modèles" with buttons "Tous", "Lié", and "Non lié". The "Modèles" section shows a list under "Os (1)" with one item: "Linux by Zabbix agent" with a radio button and "Tags" below it.

Surveillance des ressources

Une fois les hôtes configurés, Zabbix commence à collecter les données telles que : CPU, RAM, Disque dur, utilisation Réseau et beaucoup d'autre informations. Cette étape confirme le bon fonctionnement du système de supervision.

Image 12 : Graph CPU / RAM



Mise en place d'alertes

Dans une infrastructure réseau, la supervision ne se limite pas à la collection des données, elle doit permettre de détecter automatiquement les anomalies et d'alerter l'administrateur.

Cet outil permet de configurer des alertes à l'aide de triggers et de système de notifications.

Créations de triggers

Les alertes sont basées sur des déclencheurs (triggers) qui analysent les données collectées par les agents. Un trigger est une condition qui déclenche une alerte lorsqu'un seuil est dépassé, dans l'exemple d'une utilisation CPU élevée, de l'espace disque saturé ou encore de pas de mémoire ram disponible. Des triggers ont été configurés afin de surveiller les ressources critiques comme utilisation CPU élevée, espace disque insuffisante etc..

Image 13 : Création d'un trigger de surveillance CPU

The screenshot shows the Zabbix interface for creating a new trigger. The form is titled "Nouveau déclencheur" and contains the following fields and options:

- Nom:** CPU élevé
- Nom de l'événement:** CPU élevé
- Données opérationnelles:** (empty)
- Sévérité:** Non classé, Information, Avertissement, Moyen, Haut, Désastre (Haut is selected)
- * Expression:** abs(last(/client-linux/system.cpu.ut1))>90
- Génération d'événement OK:** Expression, Expression de récupération, Aucun (Expression is selected)
- Mode de génération des événements PROBLÈME:** Seul, Multiple (Seul is selected)
- Un événement OK ferme:** Tous les problèmes, Tous les problèmes si les valeurs de tag correspondent (Tous les problèmes is selected)
- Autoriser la fermeture manuelle:** (checkbox unchecked)
- Tags:** scope: security, scope: performance, scope: availability, scope: capacity

Détection des anomalies

Lorsque les seuils définis sont dépassés, Zabbix génère automatiquement une alerte visible dans l'interface.

Image 14 : problème détecté (en rouge)

The screenshot shows the Zabbix interface for viewing detected problems. The page is titled "Problèmes" and displays a table of detected issues:

Temps	Sévérité	Moment de la récupération	État	Info	Hôte	Problème	Durée	Actualiser	Actions	Tags
07:07:56	Haut		PROBLÈME		client-linux	CPU élevé	1m 10s	Actualiser		class: os component: cpu target: linux
07:00										
06:48:05	Avertissement		PROBLÈME		client-linux	Linux: System time is out of sync (diff with Zabbix server > 60s)	21m 1s	Actualiser		class: os component: system scope: availability

Le système permet également d'envoyer des notifications en cas de problème (email ou autre). Cette fonctionnalité permet d'alerter l'administrateur en temps réel.

Tests et validation

Afin de valider le bon fonctionnement de la solution de supervision mise en place avec Zabbix, plusieurs tests ont été réalisés sur les machines de l'infrastructure.

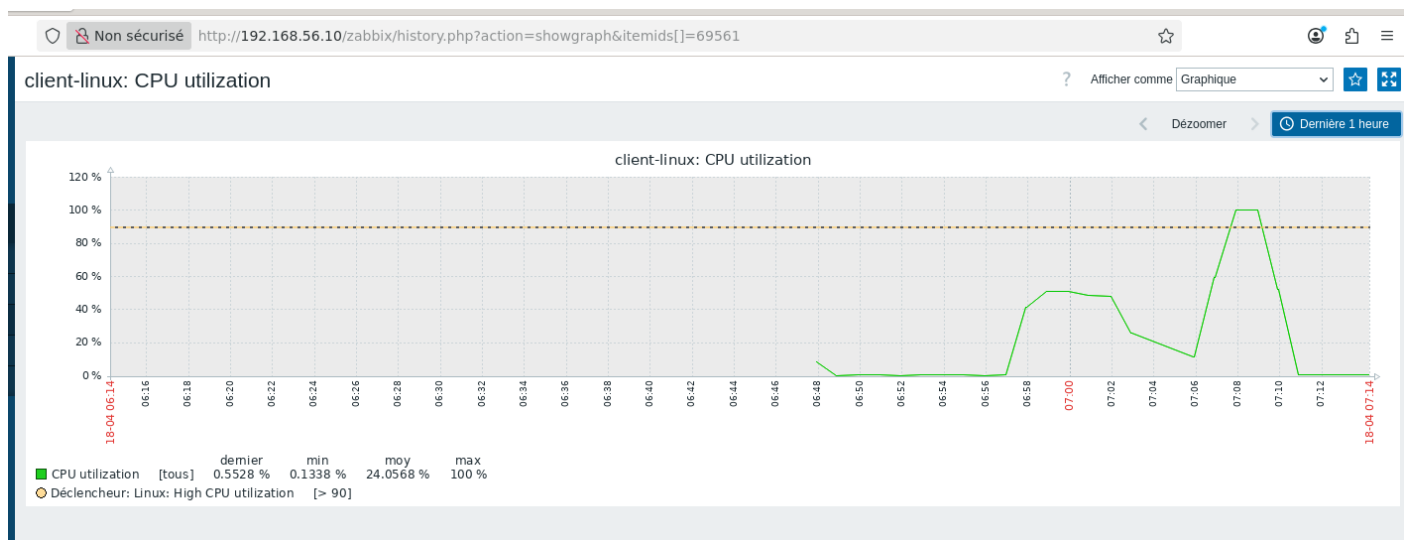
Ces tests ont pour objectif de simuler des situations réelles afin de vérifier la capacité du système à détecter les anomalies et à générer des alertes.

Simulation de surcharge CPU

Un test de charge a été réalisé sur la machine Linux afin de simuler une utilisation élevée du processeur. Cette opération a permis de vérifier le déclenchement automatique des alertes configurées.

Suite à cette simulation, une alerte a été générée dans l'interface Zabbix, confirmant le bon fonctionnement des triggers.

Image 15 : Graphique CPU élevé



Simulation d'arrêt de machine

Un second test a consisté à arrêter une machine cliente afin de simuler une panne. Cette action a entraîné la perte de communication avec le serveur Zabbix.

Le système a détecté cette anomalie et a généré une alerte indiquant que l'hôte n'est plus disponible.

Image 16 : Host en rouge (not available)

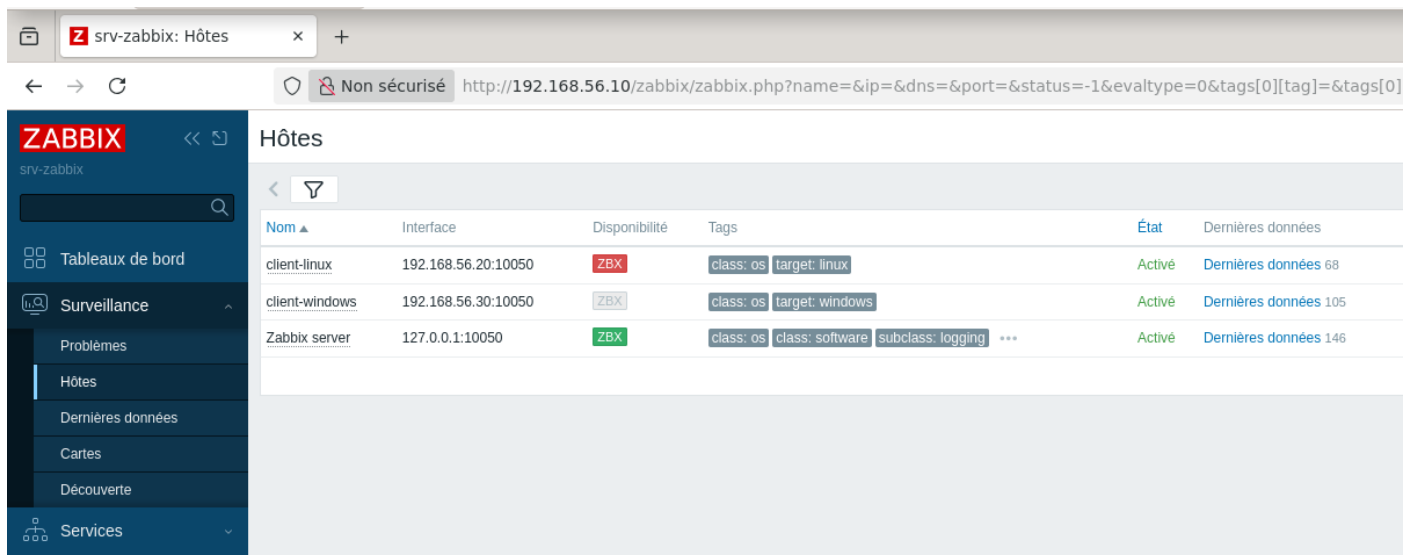


Image 17 : Details problèmes non available



Analyse des résultats

Les tests effectués démontrent que la solution de supervision est capable de :

- détecter une surcharge système
- identifier une perte de communication
- générer des alertes en temps réel

Ces résultats confirment que l'infrastructure mise en place est fonctionnelle et adaptée à un environnement de supervision.

Cette phase de validation permet de garantir la fiabilité du système avant une mise en production.

Sécurisation et limites

La sécurisation de l'infrastructure de supervision est une étape importante afin de garantir la confidentialité et l'intégrité des données échangées entre les machines et le serveur Zabbix.

Mise en place du firewall

Dans le cadre de ce projet réalisé en environnement de test sous VMware, le pare-feu n'a pas été activé sur le serveur afin de simplifier les communications entre les machines.

Cependant, dans un environnement de production, il est fortement recommandé de configurer un pare-feu afin de restreindre l'accès aux services essentiels et de sécuriser l'infrastructure.

Les ports à autoriser seraient notamment :

- 10050 pour les agents Zabbix
- 10051 pour le serveur Zabbix
- 80/443 pour l'interface web

```
ufw allow 10050
```

```
ufw allow 10051
```

```
ufw allow 80
```

```
ufw enable
```

```
ufw status
```

Cette configuration permet de réduire les risques d'accès non autorisés.

Sécurisation des communications

Dans ce projet, la communication entre les agents et le serveur est réalisée en clair au sein d'un réseau local. Toutefois, Zabbix permet la mise en place de mécanismes de chiffrement (PSK ou certificats) afin de sécuriser les échanges dans un environnement de production.

Limites du projet

Malgré le bon fonctionnement du système, certaines limites ont été identifiées :

- absence de chiffrement des communications
- infrastructure limitée à un environnement de test (VMware)
- nombre restreint de machines supervisées
- absence de supervision avancée (SNMP, équipements réseau réels)

Conclusion

Ce projet a permis de mettre en place une solution complète de supervision basée sur Zabbix au sein d'un environnement virtualisé.

L'installation et la configuration du serveur de supervision ont été réalisées avec succès, suivies du déploiement des agents sur différentes machines Linux et Windows. La supervision des ressources système a été mise en œuvre grâce à l'utilisation de templates, permettant de collecter et d'analyser les données en temps réel.

La mise en place de déclencheurs a permis de détecter automatiquement les anomalies, tandis que les tests réalisés ont validé le bon fonctionnement du système, notamment lors de simulations de surcharge et de panne.

Ce projet démontre l'intérêt d'un outil de supervision dans la gestion d'une infrastructure informatique, en offrant une meilleure visibilité et une réactivité accrue face aux incidents.

Des améliorations peuvent être envisagées, telles que l'ajout de nouvelles machines, la sécurisation des communications ou encore l'intégration d'équipements réseau.