

# Rapport de Projet

## Mise en œuvre d'une infrastructure cloud de supervision centralisée sous AWS

Réalisé par :

Bouhassoune Mohamed

Classe : 2ACI INFO GA



Encadré par :

Monsieur Azzedine Khat

But du projet :

Déployer une solution de supervision centralisée dans un environnement cloud AWS à l'aide de **Zabbix** (serveur déployé sous **Docker** avec une base de données **MariaDB**), afin de superviser des hôtes **Linux** et **Windows** via des **adresses IP privées** et d'assurer le suivi des métriques système (CPU, mémoire, disque et disponibilité).

Année universitaire : 2025–2026

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Architecture générale de la solution</b>	<b>4</b>
1.1 Vue d'ensemble . . . . .	4
1.2 Choix des IP privées (bonne pratique Cloud) . . . . .	4
<b>2 Mise en œuvre de l'infrastructure AWS</b>	<b>5</b>
2.1 Création du VPC . . . . .	5
2.2 Création des subnets et association . . . . .	6
2.3 Internet Gateway (accès Internet) . . . . .	7
2.4 Table de routage . . . . .	8
2.5 Groupes de sécurité (Security Groups) . . . . .	8
2.6 Création des instances EC2 . . . . .	9
<b>3 Déploiement de Zabbix sur le serveur (Docker)</b>	<b>11</b>
3.1 Installation et vérification de Docker / Docker Compose . . . . .	11
3.2 Conteneurs Zabbix en exécution . . . . .	11
3.3 Migration base de données et stabilisation . . . . .	12
3.4 Accès à l'interface Zabbix . . . . .	12
<b>4 Ajout et supervision des hôtes (Linux / Windows)</b>	<b>14</b>
4.1 Création du host Zabbix Server . . . . .	14
4.2 Configuration correcte (fichier agent) . . . . .	15
4.3 Création du client Linux . . . . .	16
4.3.1 Graphiques Linux . . . . .	16
4.4 Création du client Windows . . . . .	17
4.5 Création du groupe d'hôtes Windows . . . . .	18
4.6 Agent Windows (installation et fonctionnement) . . . . .	18
4.7 Client Windows supervisé . . . . .	19
4.7.1 Graphiques Windows . . . . .	20

<b>5</b>	<b>Résultats et preuves de supervision</b>	<b>21</b>
5.1	Disponibilité des hôtes (ZBX vert) . . . . .	21
5.2	Dashboard global . . . . .	22
<b>6</b>	<b>Problèmes rencontrés et solutions apportées</b>	<b>23</b>
6.1	Restrictions du Lab (droits AWS / SCP) . . . . .	23
6.2	Docker : permissions et exécution . . . . .	23
6.3	Compatibilité base de données (MySQL / MariaDB) . . . . .	23
6.4	Communication Agent & Docker (IP privée vs 127.0.0.1) . . . . .	23
6.5	Windows : absence de navigateur et installation MSI . . . . .	24
	<b>Conclusion</b>	<b>25</b>

# Introduction

La supervision est essentielle pour garantir la disponibilité et la performance d'une infrastructure, particulièrement dans un environnement cloud où les ressources sont distribuées. Ce projet consiste à mettre en place une **supervision centralisée** sur **AWS**, à l'aide de **Zabbix**.

L'objectif est de :

- déployer un serveur Zabbix dans une instance EC2 (Zabbix Server + Zabbix Web + base de données) via Docker ;
- ajouter et superviser un **client Linux** et un **client Windows** ;
- obtenir des preuves de fonctionnement : disponibilité des hôtes (ZBX vert), collecte des métriques et graphiques.

# Chapitre 1

## Architecture générale de la solution

### 1.1 Vue d'ensemble

L'architecture déployée se compose de :

- Un **VPC AWS** avec réseau privé (adressage 10.0.0.0/16 par exemple) ;
- Un serveur **Zabbix** (EC2 Linux) hébergeant :
  - Zabbix Server (conteneur),
  - Zabbix Web (conteneur),
  - MariaDB (conteneur).
- Un **client Linux** (EC2 Ubuntu) avec Zabbix Agent ;
- Un **client Windows** (EC2 Windows) avec Zabbix Agent.

### 1.2 Choix des IP privées (bonne pratique Cloud)

La communication entre le serveur Zabbix et les agents est réalisée via les **IP privées** au sein du VPC :

- meilleure sécurité (pas d'exposition inutile sur Internet),
- latence faible dans le réseau interne AWS,
- respect des bonnes pratiques d'architecture cloud.

# Chapitre 2

## Mise en œuvre de l'infrastructure AWS

### 2.1 Création du VPC

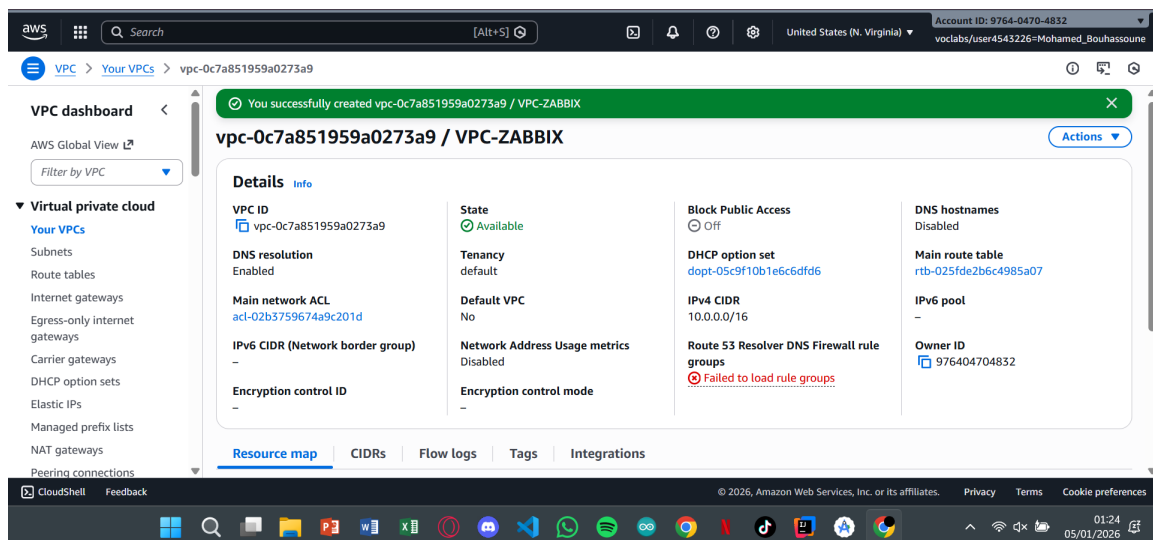


FIGURE 2.1 – Création du VPC AWS

**Explication :** un VPC est créé afin d'isoler l'infrastructure et de maîtriser l'adressage IP, les routes et la sécurité réseau.

## 2.2 Création des subnets et association

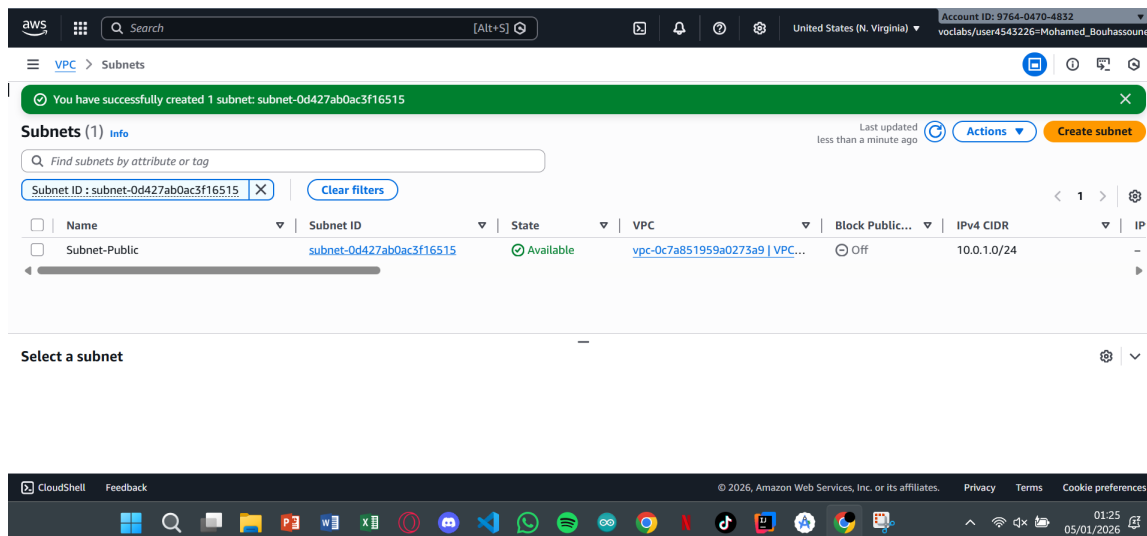


FIGURE 2.2 – Création du subnet

**Explication :** un subnet est créé pour héberger les instances EC2 (serveur Zabbix et clients). Il permet de segmenter le réseau.

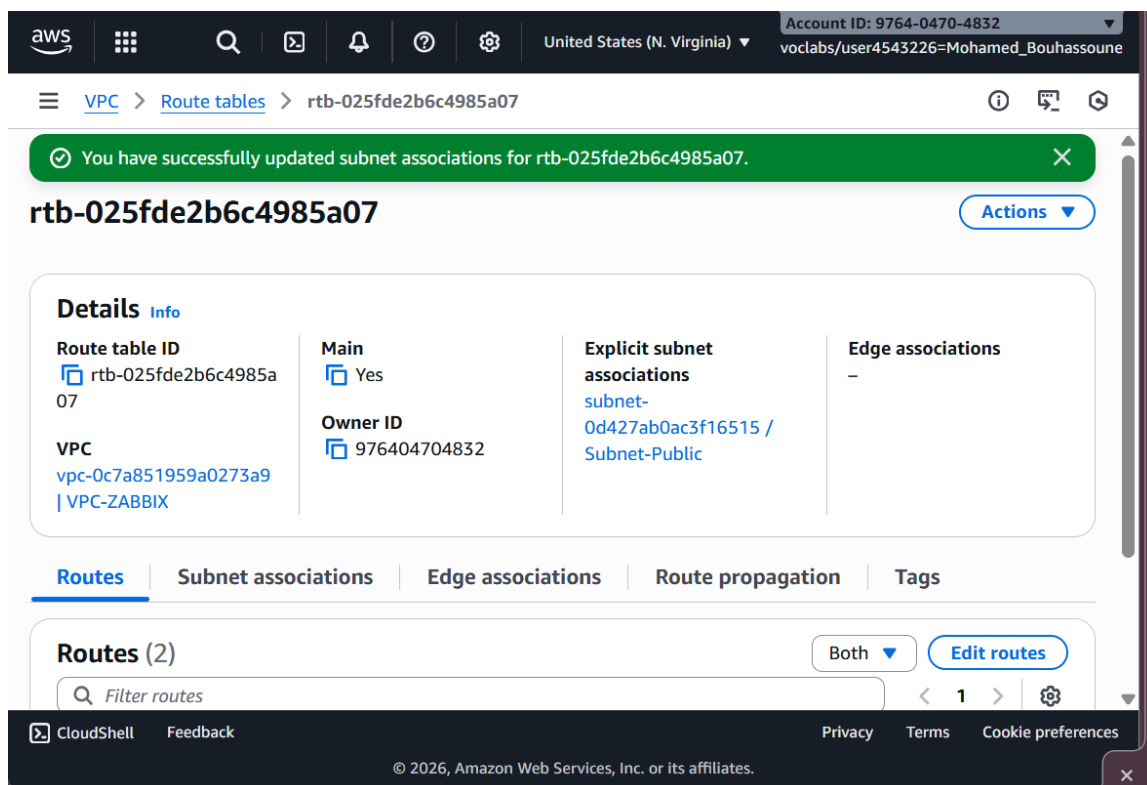


FIGURE 2.3 – Association / vérification du subnet

**Explication :** le subnet est bien rattaché au VPC, garantissant que les instances appartiennent au même réseau interne.

## 2.3 Internet Gateway (accès Internet)

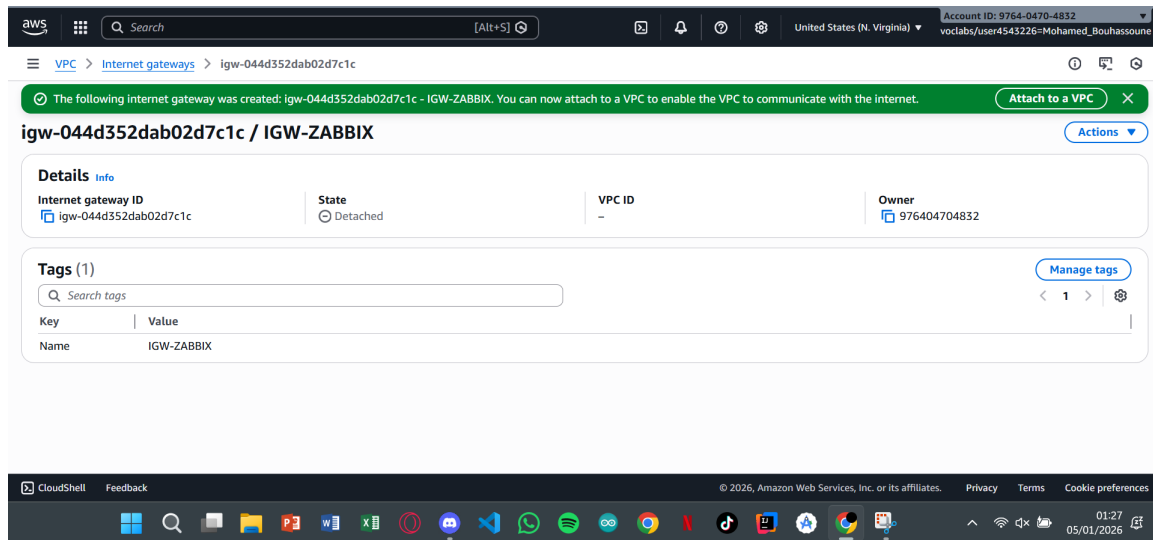


FIGURE 2.4 – Création de l'Internet Gateway

**Explication :** l'Internet Gateway permet l'accès Internet (mise à jour, téléchargement paquets, accès à l'interface web via IP publique).

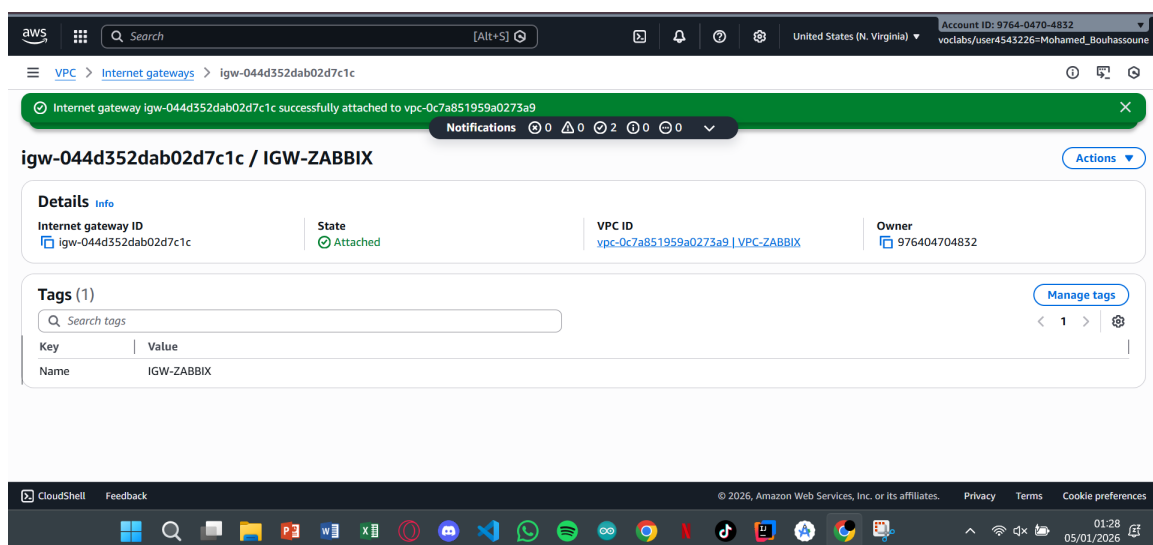


FIGURE 2.5 – Attachement de l'Internet Gateway au VPC

**Explication :** l'IGW est attachée au VPC afin d'activer la connectivité Internet.



## 2.4 Table de routage

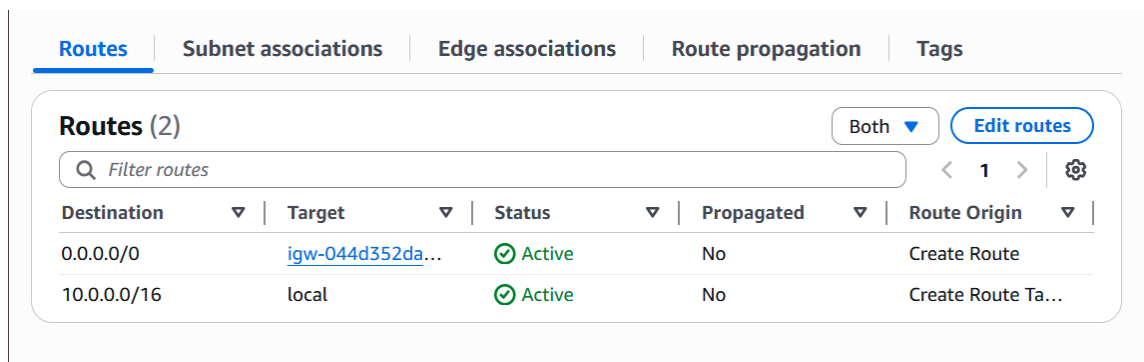


FIGURE 2.6 – Création de la route (accès Internet)

**Explication :** ajout d'une route (0.0.0.0/0) vers l'Internet Gateway pour permettre aux instances d'accéder à Internet.

## 2.5 Groupes de sécurité (Security Groups)

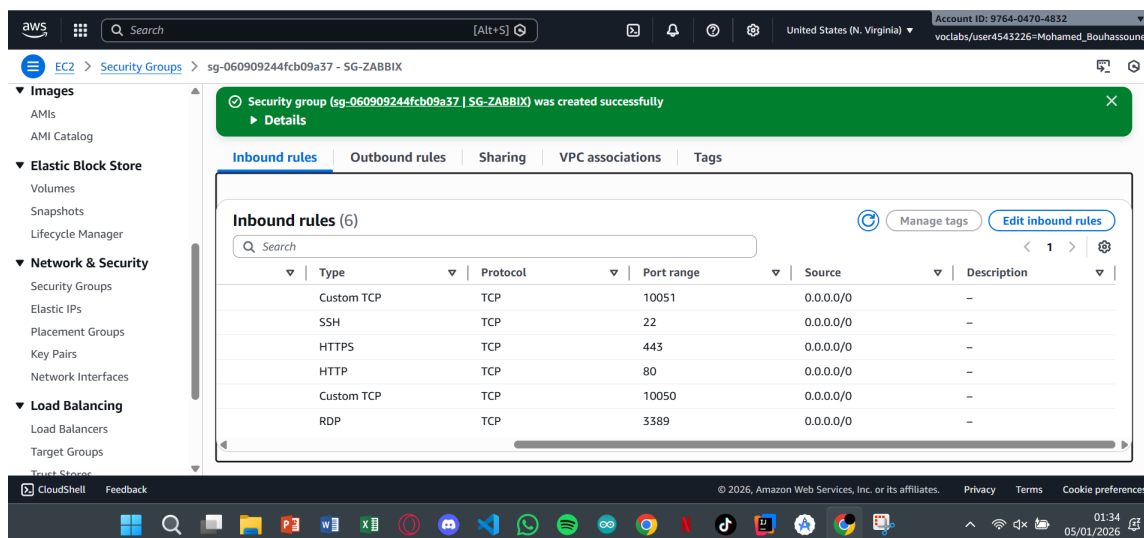


FIGURE 2.7 – Règles inbound du Security Group

**Explication :** ouverture des ports nécessaires uniquement :

- 22/TCP (SSH) pour l'administration,
- 80/TCP (HTTP) pour l'interface Zabbix Web,
- 10050/TCP (Zabbix Agent),
- 10051/TCP (Zabbix Server).

## 2.6 Création des instances EC2

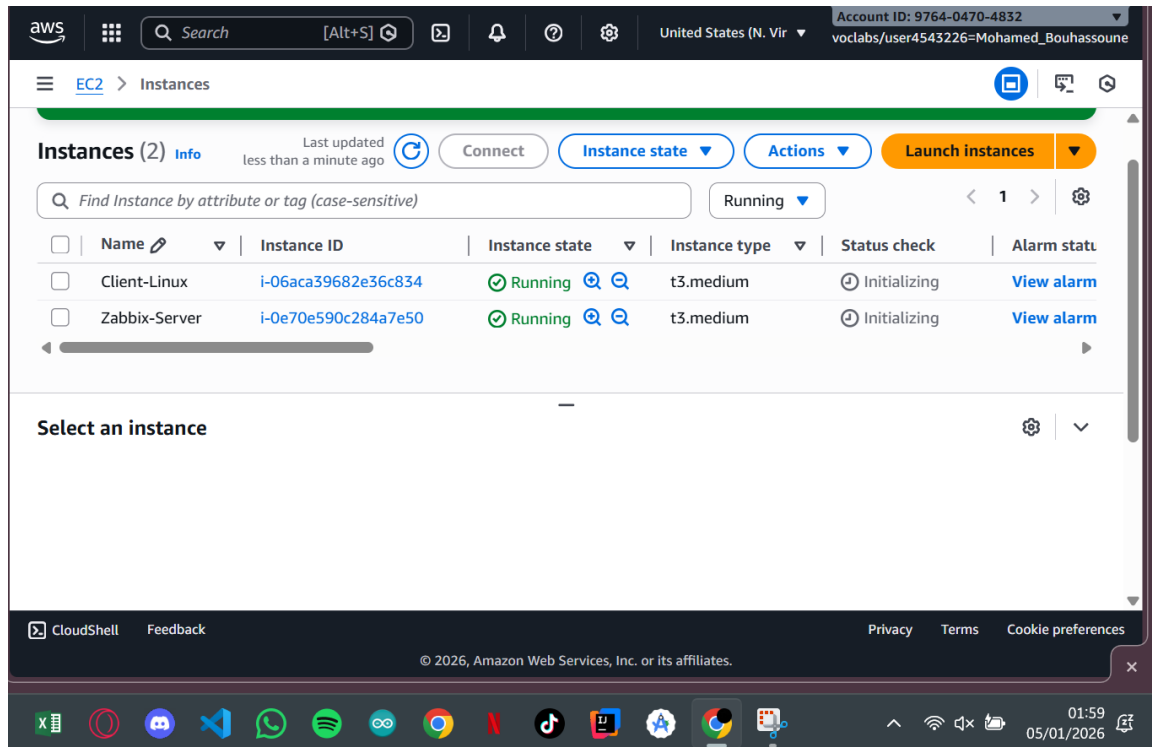


FIGURE 2.8 – Instance EC2 du serveur Zabbix créée

**Explication :** une instance EC2 Linux héberge la stack Zabbix (Docker). Elle doit avoir des ressources suffisantes (CPU/RAM) pour les conteneurs.

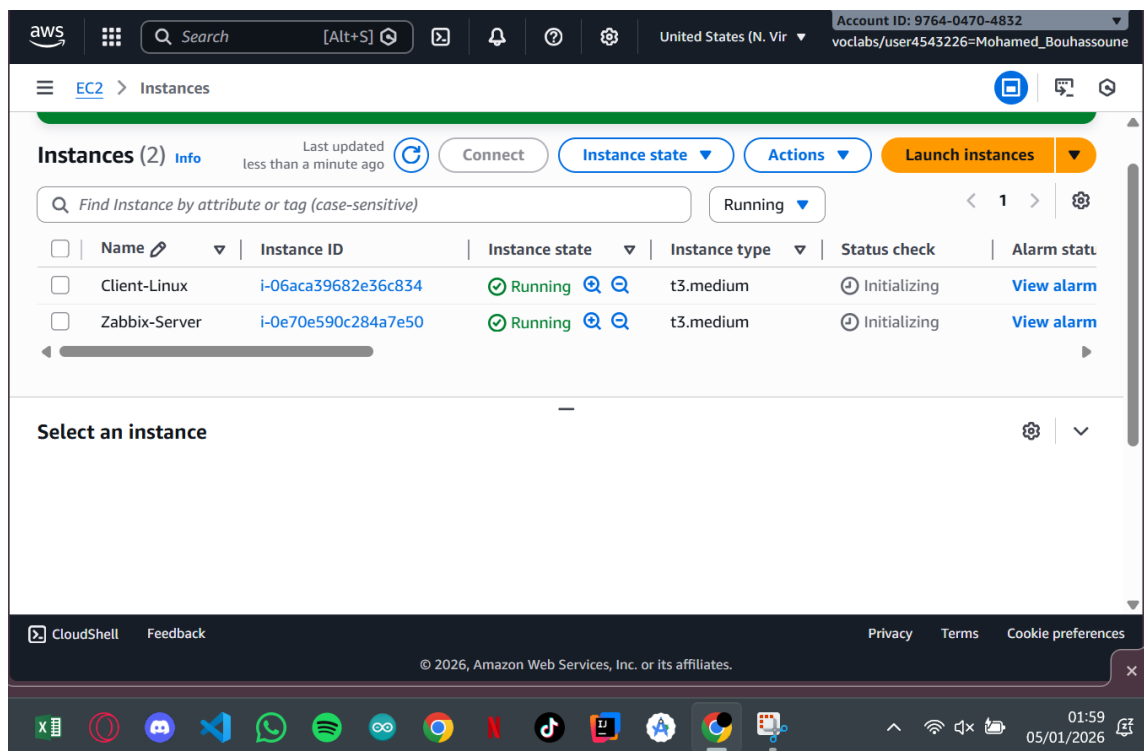


FIGURE 2.9 – Instance EC2 du client Linux créée

**Explication :** instance Linux utilisée comme machine supervisée (installation de Zabbix Agent).

# Chapitre 3

## Déploiement de Zabbix sur le serveur (Docker)

### 3.1 Installation et vérification de Docker / Docker Compose

```
ubuntu@ip-10-0-1-163:~$ docker --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
ubuntu@ip-10-0-1-163:~$ docker-compose --version
docker-compose version 1.29.2, build unknown
ubuntu@ip-10-0-1-163:~$
```

FIGURE 3.1 – Vérification des versions Docker et Docker Compose

**Explication :** Docker et Docker Compose sont nécessaires pour déployer Zabbix Server, Zabbix Web et la base de données sous forme de conteneurs.

### 3.2 Conteneurs Zabbix en exécution

```
Last login: Mon Jan 5 01:07:02 2026 from 196.64.12.74
ubuntu@ip-10-0-1-163:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND
CREATED       STATUS      NAMES      PORTS
f4cfd8b05a9f   zabbix/zabbix-web-nginx-mysql:latest "docker-entrypoint.sh"
 2 minutes ago Up 2 minutes (healthy)      8443/tcp, 0.0.0.0:80->8080/
tcp, [::]:80->8080/tcp   zabbix-web
edc52dbd1722   zabbix/zabbix-server-mysql:latest   "/usr/bin/docker-ent..."
 2 minutes ago Restarting (1) 24 seconds ago
                        zabbix-server
91578e76b36d   mysql:8.0                                "docker-entrypoint.s..."
 2 minutes ago Up 2 minutes
                        3306/tcp, 33060/tcp
                        zabbix-mysql
ubuntu@ip-10-0-1-163:~$
```

FIGURE 3.2 – Conteneurs Docker : Zabbix Server, Zabbix Web et base de données

**Explication :** la commande `docker ps` confirme que les services sont démarrés et exposés (web, serveur et base).

### 3.3 Migration base de données et stabilisation

```
ubuntu@ip-10-0-1-163:~/zabbix-docker$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
848483748162	zabbix/zabbix-web-nginx-mysql:6.4.0-ubuntu	"docker-entrypoint.sh"	6 seconds ago	Up 6 seconds	8443/tcp, 0.0.0.0:80->8080/tcp, [::]:80->8080/tcp
422a848cc5ef	zabbix/zabbix-server-mysql:6.4.0-ubuntu	"/usr/bin/tini -- /u..."	7 seconds ago	Up 6 seconds	0.0.0.0:10051->10051/tcp, [::]:10051->10051/tcp
64e08f43b2ca	mariadb:10.11	"docker-entrypoint.s..."	7 seconds ago	Up 7 seconds	3306/tcp

```
ubuntu@ip-10-0-1-163:~/zabbix-docker$
```

FIGURE 3.3 – Passage vers MariaDB pour compatibilité

**Explication :** lors du déploiement, des incompatibilités de versions MySQL/MariaDB ont empêché Zabbix Server de démarrer. Le choix de MariaDB et l'ajustement de la configuration ont permis la stabilisation.

```
ubuntu@ip-10-0-1-163:~/zabbix-docker$ nano docker-compose.yml
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
8a2153bc1bb5	zabbix/zabbix-web-nginx-mysql:6.4.0-ubuntu	"docker-entrypoint.sh"	2 minutes ago	Up 2 minutes	8443/tcp, 0.0.0.0:80->8080/tcp, [::]:80->8080/tcp
ec140f7e8c64	zabbix/zabbix-server-mysql:6.4.0-ubuntu	"/usr/bin/tini -- /u..."	2 minutes ago	Up 2 minutes	0.0.0.0:10051->10051/tcp, [::]:10051->10051/tcp
f5e6dfc62383	mariadb:10.11	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes	3306/tcp

```
ubuntu@ip-10-0-1-163:~/zabbix-docker$
```

FIGURE 3.4 – Stack Zabbix stabilisée après ajustements (DB + paramètre support)

**Explication :** après correction (base compatible et autorisation de version si nécessaire), le serveur Zabbix fonctionne correctement.

### 3.4 Accès à l'interface Zabbix

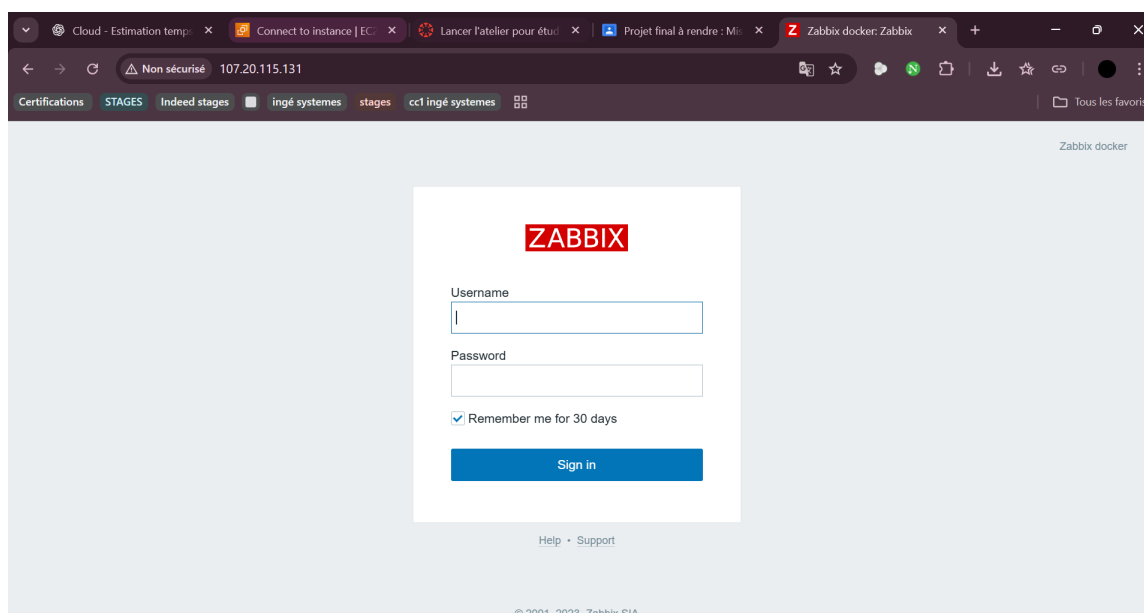


FIGURE 3.5 – Interface Zabbix Web accessible

**Explication :** l'interface web permet d'ajouter les hôtes, d'associer des templates et de visualiser les données (latest data, graphs, dashboard).

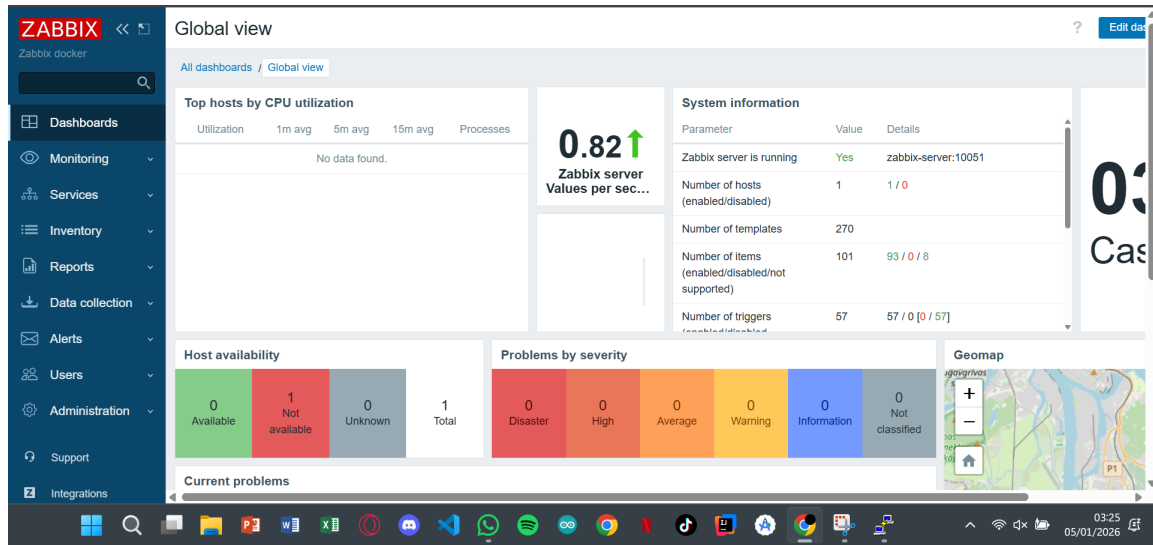


FIGURE 3.6 – Zabbix opérationnel

**Explication :** confirmation du bon fonctionnement global de la plateforme (web + serveur + base).

# Chapitre 4

## Ajout et supervision des hôtes (Linux / Windows)

### 4.1 Création du host Zabbix Server

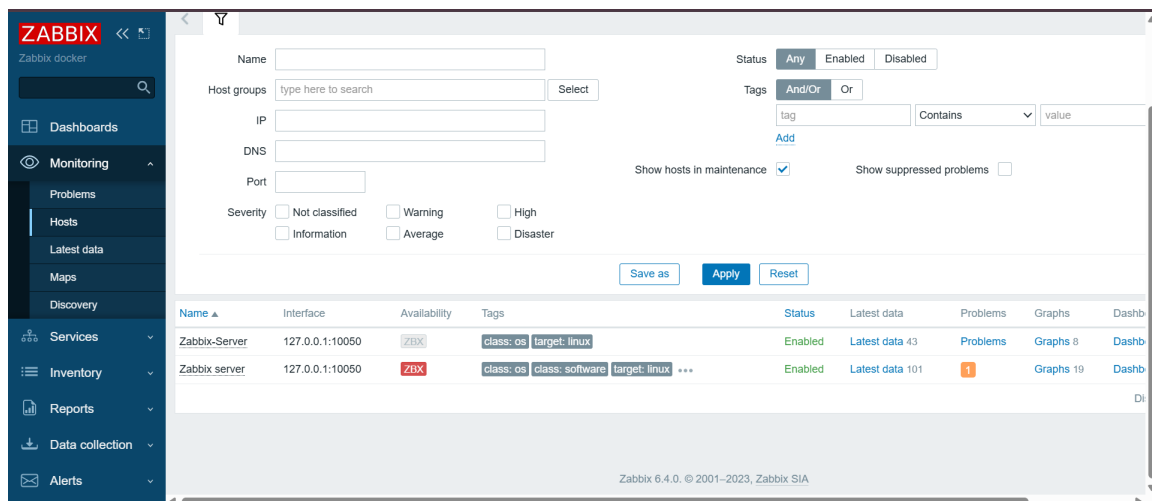
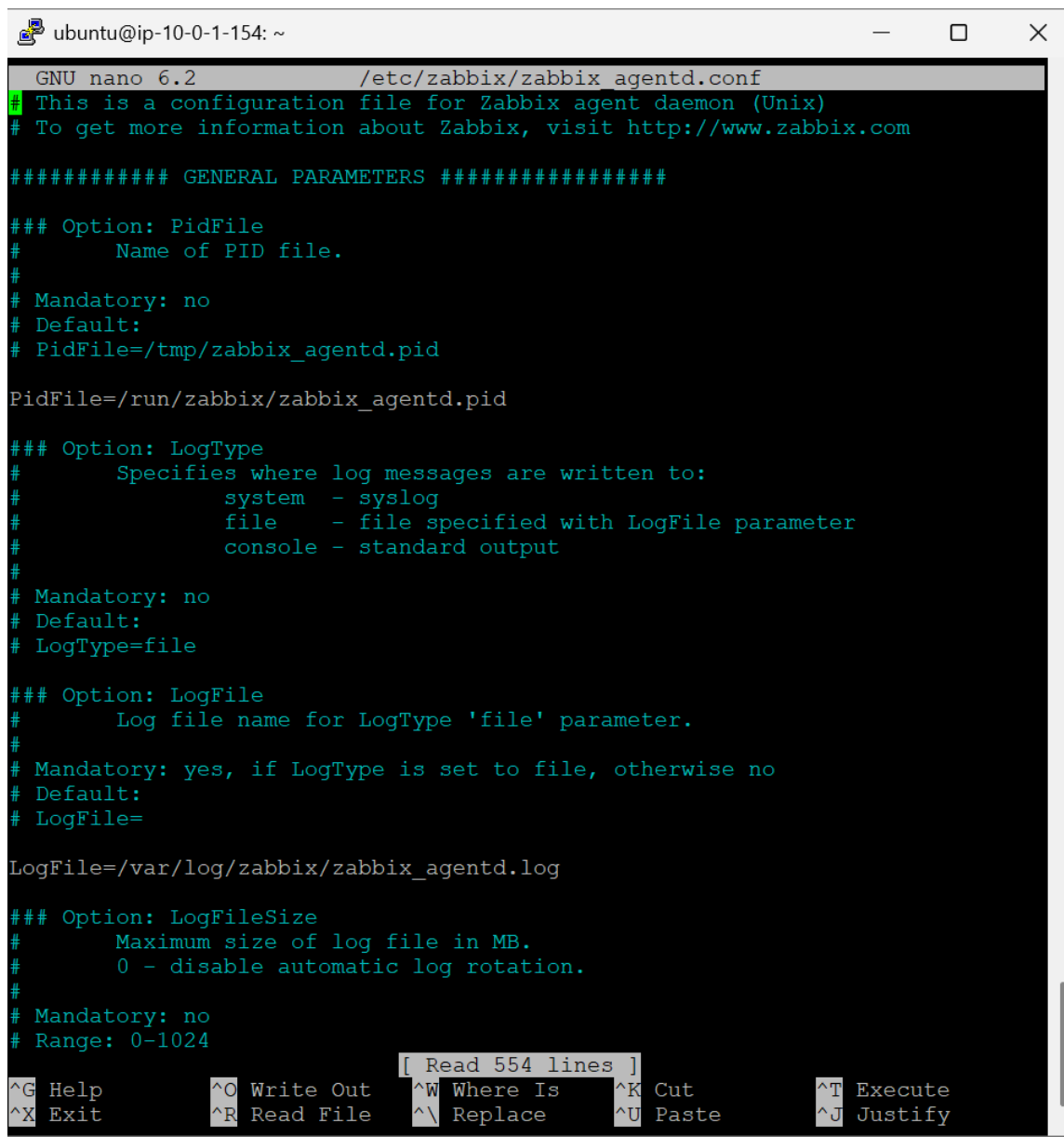


FIGURE 4.1 – Création de l'hôte Zabbix Server dans l'interface

**Explication :** l'hôte représentant le serveur est ajouté dans Zabbix afin de superviser la machine (CPU, mémoire, disque). La communication doit se faire via IP privée (important en environnement Docker).

## 4.2 Configuration correcte (fichier agent)



```
ubuntu@ip-10-0-1-154: ~  
GNU nano 6.2 /etc/zabbix/zabbix_agentd.conf  
# This is a configuration file for Zabbix agent daemon (Unix)  
# To get more information about Zabbix, visit http://www.zabbix.com  
  
##### GENERAL PARAMETERS #####  
  
### Option: PidFile  
# Name of PID file.  
#  
# Mandatory: no  
# Default:  
# PidFile=/tmp/zabbix_agentd.pid  
  
PidFile=/run/zabbix/zabbix_agentd.pid  
  
### Option: LogType  
# Specifies where log messages are written to:  
# system - syslog  
# file - file specified with LogFile parameter  
# console - standard output  
#  
# Mandatory: no  
# Default:  
# LogType=file  
  
### Option: LogFile  
# Log file name for LogType 'file' parameter.  
#  
# Mandatory: yes, if LogType is set to file, otherwise no  
# Default:  
# LogFile=  
  
LogFile=/var/log/zabbix/zabbix_agentd.log  
  
### Option: LogFileSize  
# Maximum size of log file in MB.  
# 0 - disable automatic log rotation.  
#  
# Mandatory: no  
# Range: 0-1024  
  
[ Read 554 lines ]  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

FIGURE 4.2 – Vérification de la configuration de l’agent (Hostname / Server)

**Explication :** le **Hostname** doit correspondre exactement au nom de l’hôte déclaré dans Zabbix (casse incluse). Les paramètres **Server** et **ServerActive** doivent permettre la communication réseau.



## 4.3 Création du client Linux

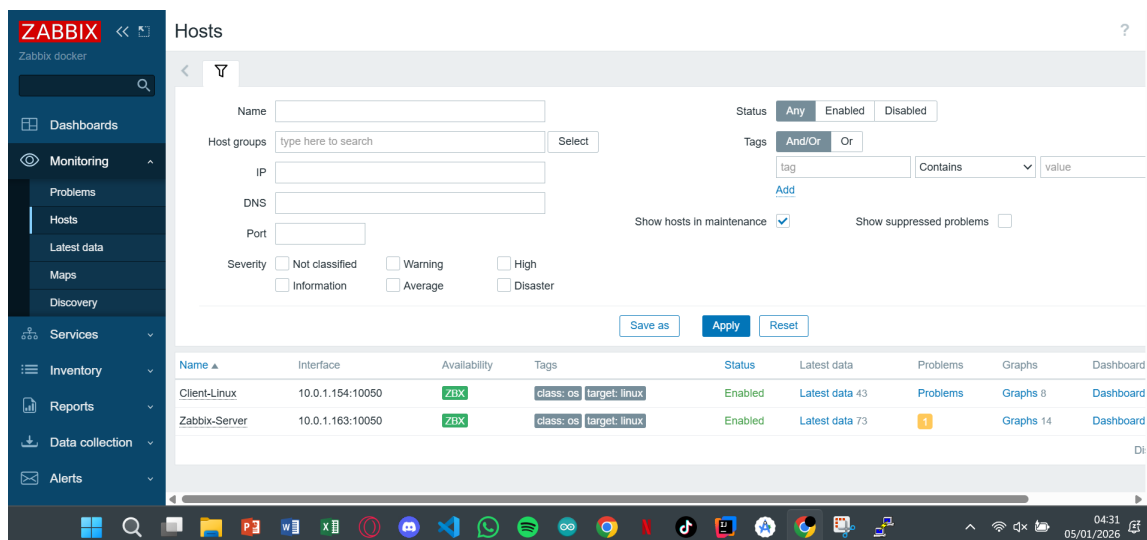


FIGURE 4.3 – Création de l'hôte Client-Linux dans Zabbix

**Explication :** le client Linux est ajouté avec son IP privée et le template *Linux by Zabbix agent*. L'agent installé sur la machine remonte les métriques.

### 4.3.1 Graphiques Linux

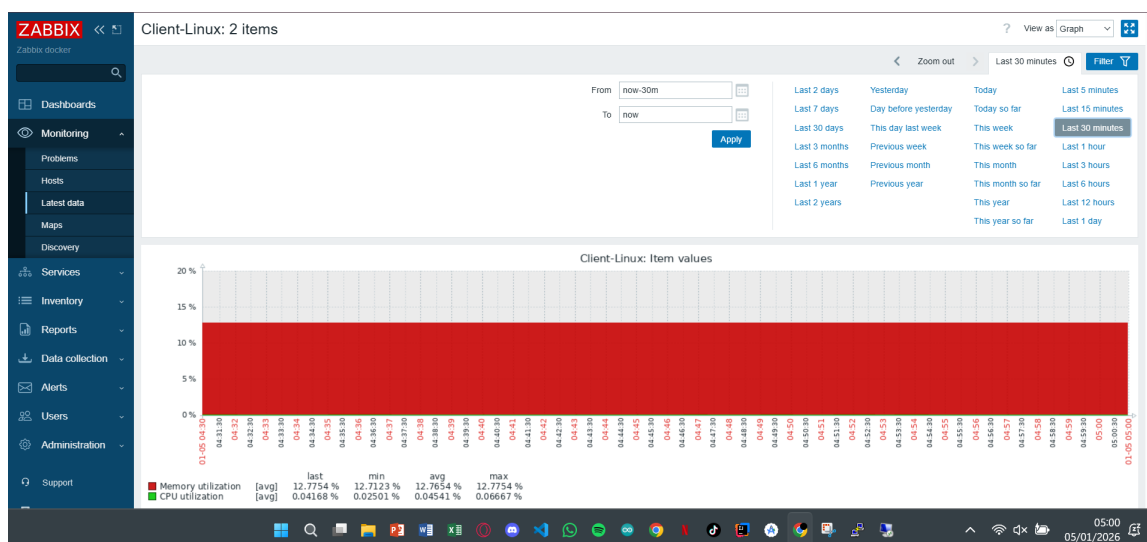


FIGURE 4.4 – Graphiques CPU / mémoire du client Linux

**Explication :** les graphes confirment la collecte en temps réel des métriques et permettent une visualisation rapide de la charge.

## 4.4 Création du client Windows

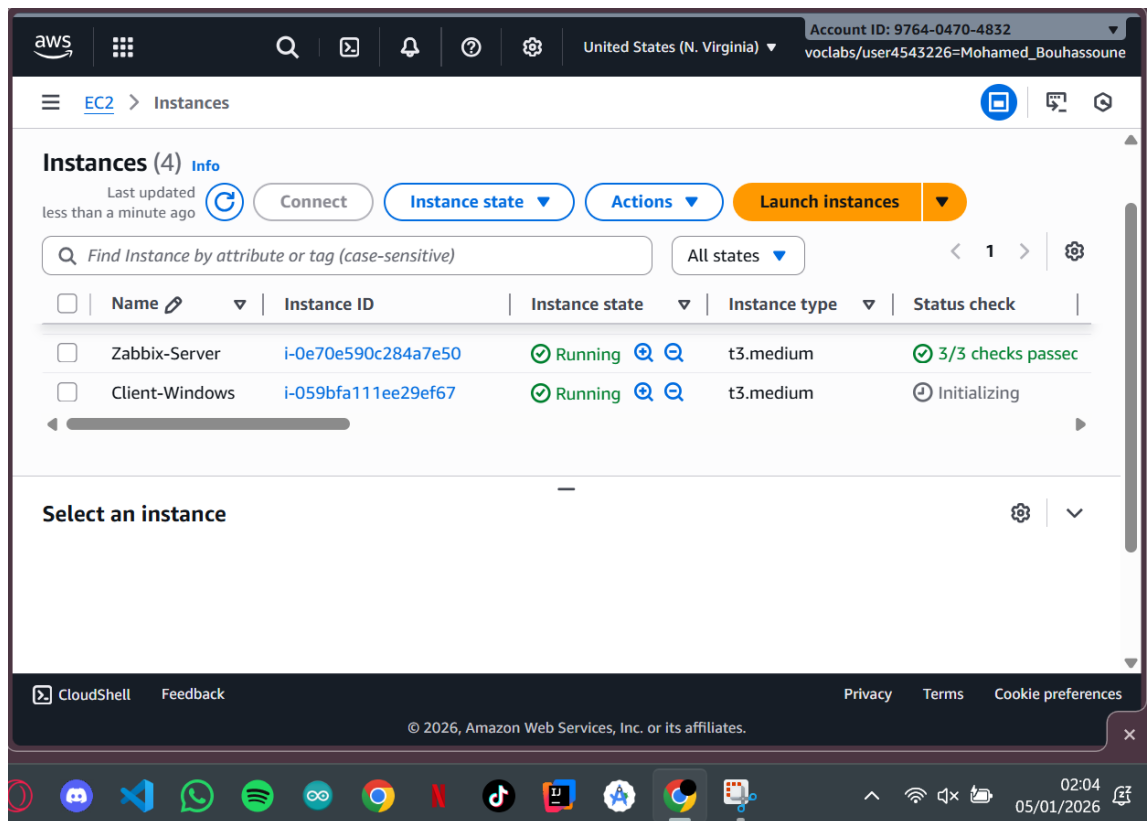


FIGURE 4.5 – Création de l'hôte Client-Windows dans Zabbix

**Explication :** le client Windows est ajouté avec IP privée et template *Windows by Zabbix agent*. L'agent doit être installé et le port 10050 autorisé.

## 4.5 Création du groupe d'hôtes Windows

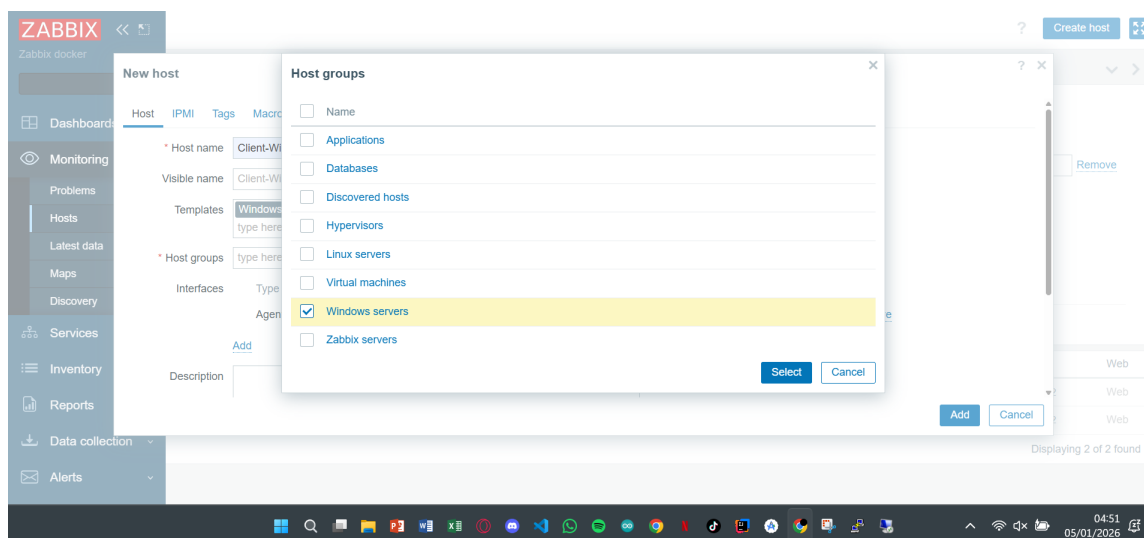


FIGURE 4.6 – Ajout du groupe d'hôtes Windows

**Explication :** un groupe dédié *Windows servers* permet d'organiser les hôtes Windows et d'appliquer plus facilement les templates et politiques.

## 4.6 Agent Windows (installation et fonctionnement)

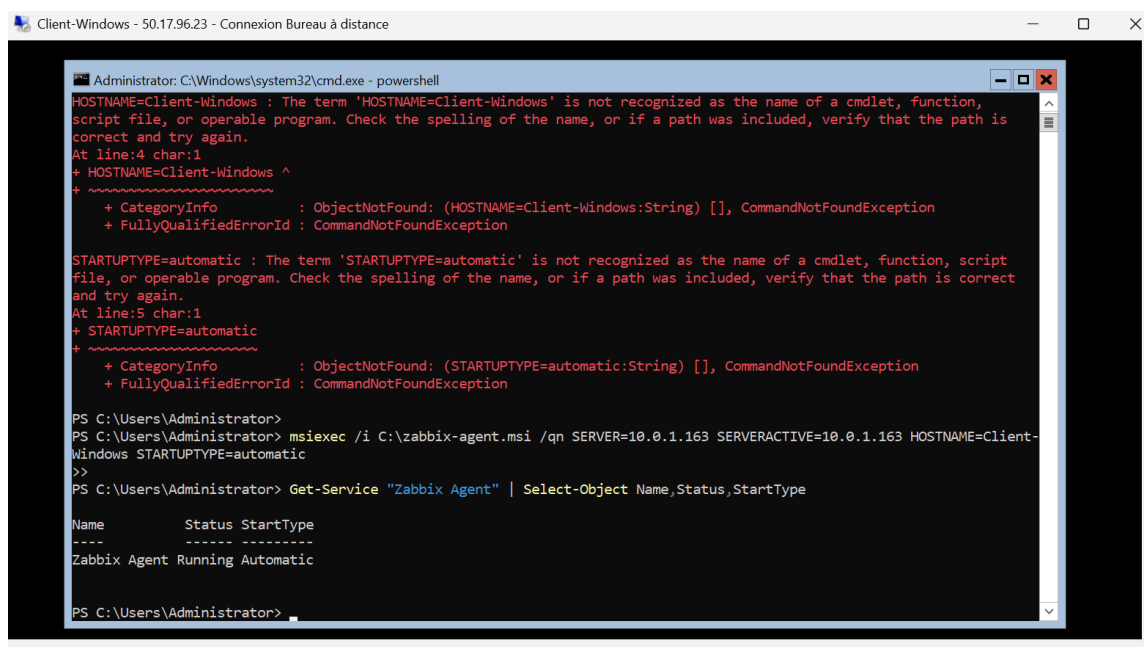


FIGURE 4.7 – Agent Zabbix sur Windows (fonctionnel)

**Explication :** validation que l'agent Windows est opérationnel (service en cours d'exécution + communication avec le serveur).

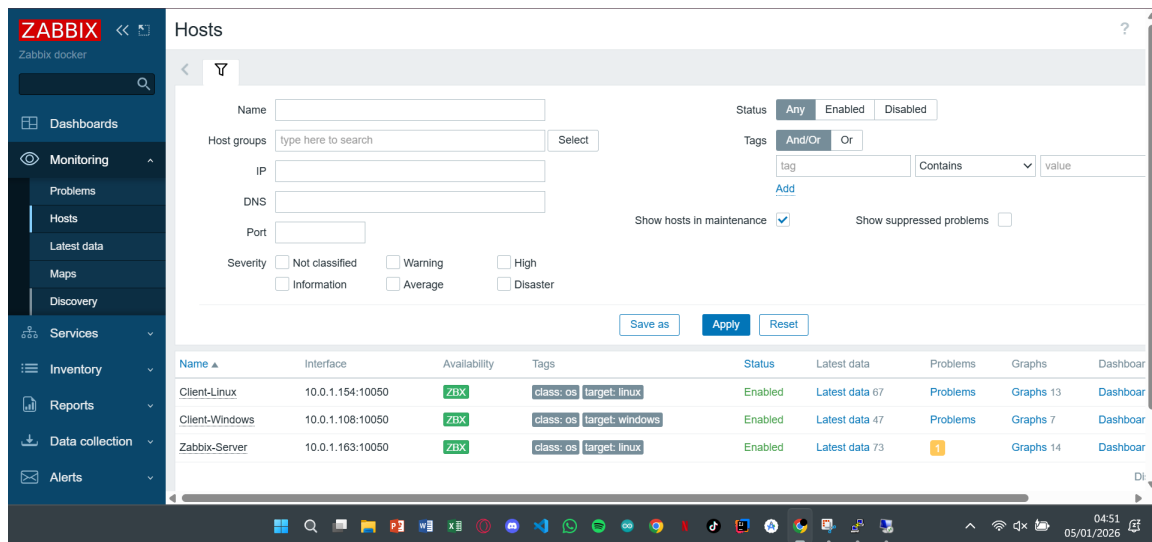


FIGURE 4.8 – Confirmation du bon fonctionnement de l'agent Windows

**Explication :** preuve supplémentaire utilisée pour le rapport : l'agent répond et Zabbix peut collecter les données.

## 4.7 Client Windows supervisé

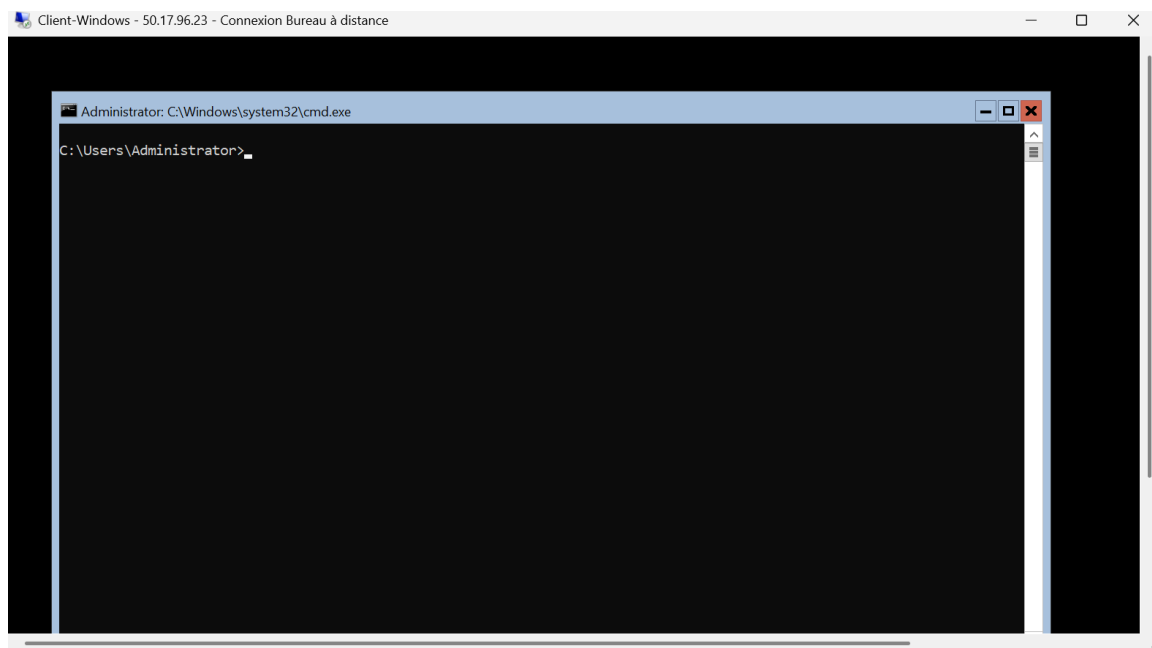


FIGURE 4.9 – Client Windows supervisé dans Zabbix

**Explication :** après association du template Windows et ouverture du port 10050, l'hôte devient disponible (ZBX vert).

## 4.7.1 Graphiques Windows

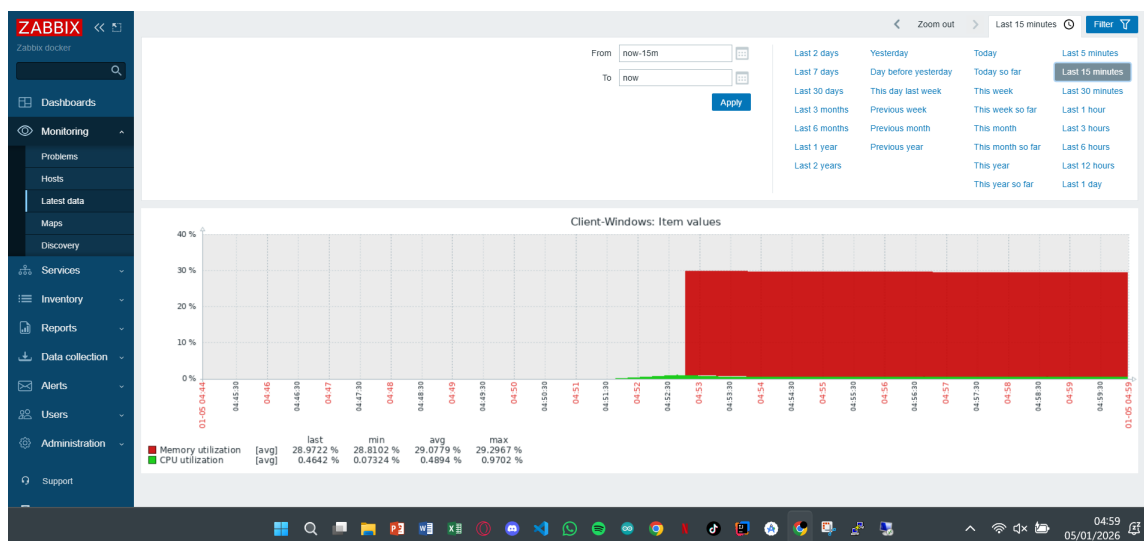


FIGURE 4.10 – Graphiques CPU / mémoire du client Windows

**Explication :** représentation de l'utilisation des ressources sur la machine Windows supervisée.

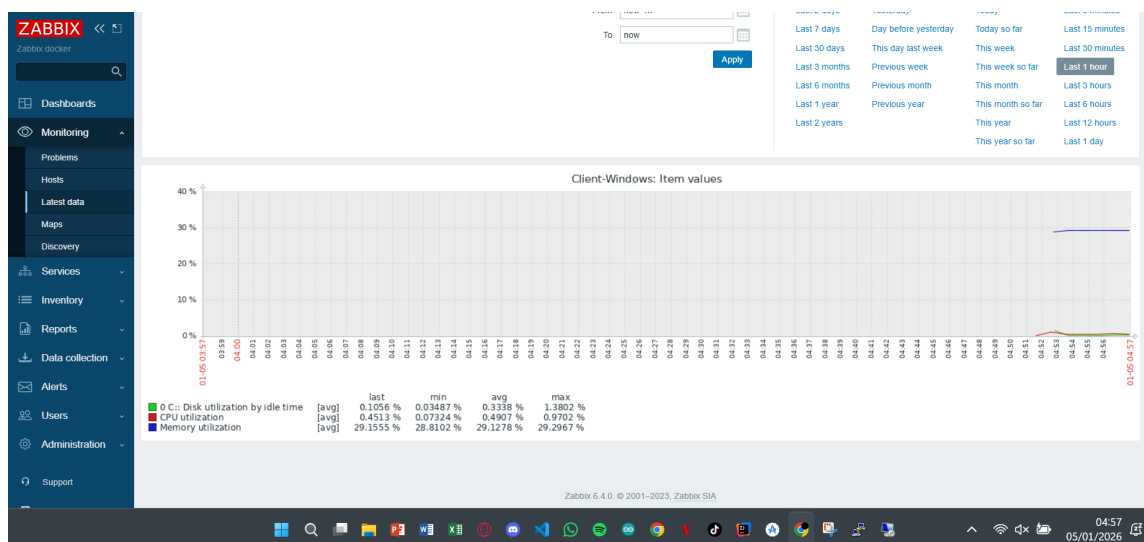


FIGURE 4.11 – Graphiques CPU / disque / mémoire (Windows)

**Explication :** visualisation plus complète des ressources (disque, mémoire, CPU) pour démontrer la richesse de la supervision.

# Chapitre 5

## Résultats et preuves de supervision

### 5.1 Disponibilité des hôtes (ZBX vert)

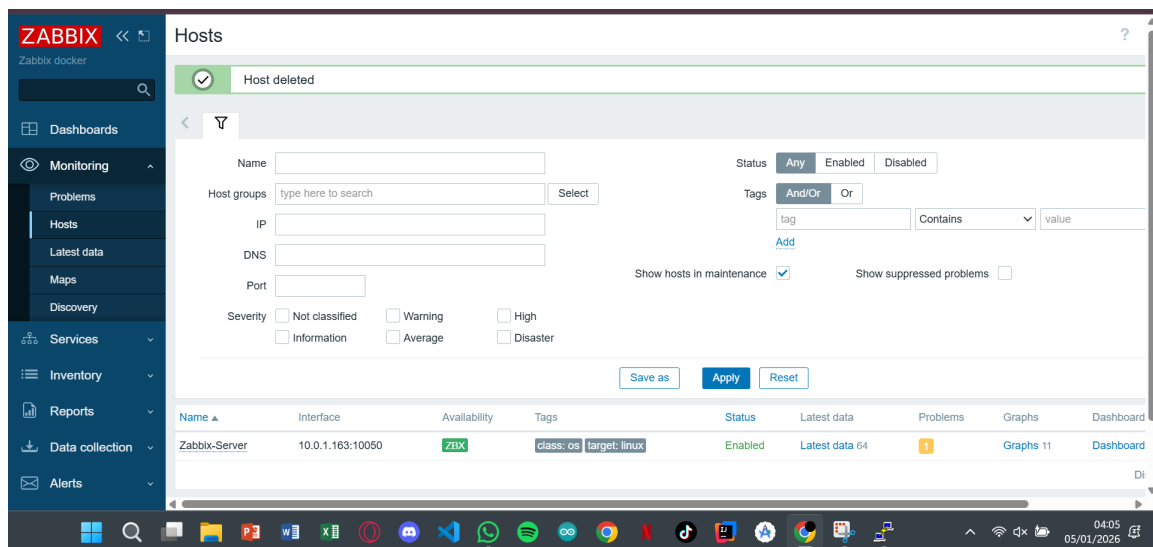


FIGURE 5.1 – Liste des hôtes supervisés avec disponibilité (ZBX vert)

**Explication :** ce tableau prouve que les trois hôtes (Zabbix-Server, Client-Linux, Client-Windows) communiquent avec le serveur de supervision.

## 5.2 Dashboard global

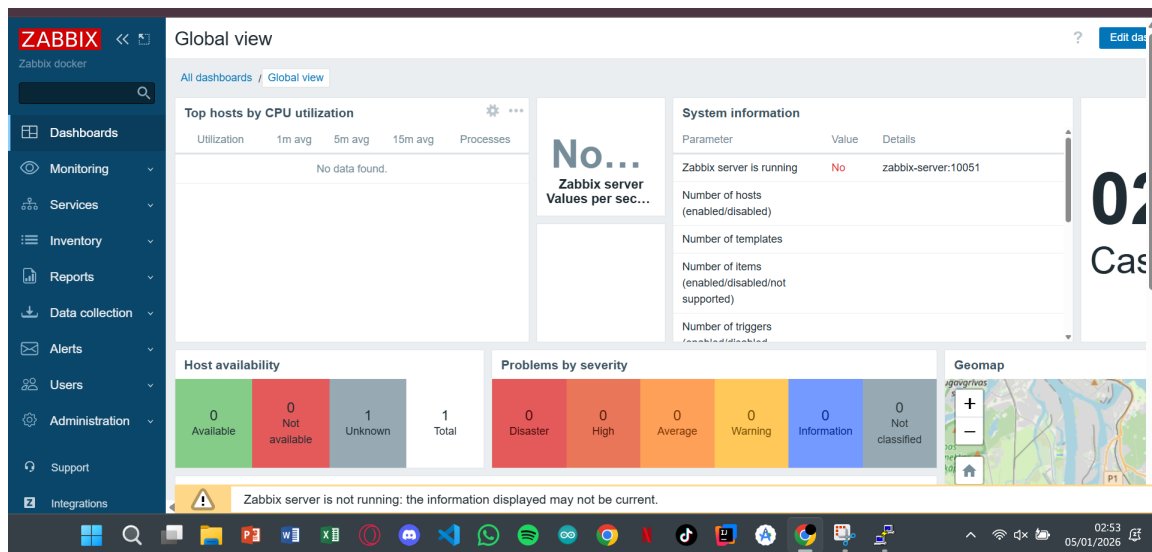


FIGURE 5.2 – Dashboard Zabbix (vue synthétique)

**Explication :** le dashboard regroupe les informations essentielles (disponibilité, métriques, widgets) pour une supervision centralisée rapide.

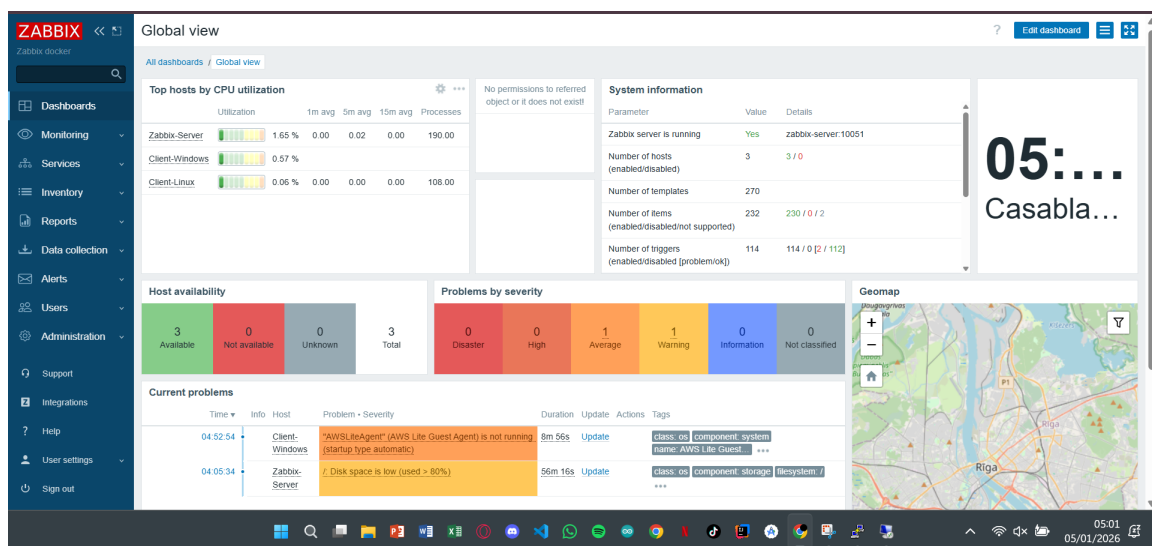


FIGURE 5.3 – Dashboard final avec données collectées

**Explication :** confirmation que la collecte est effective (widgets alimentés par des données réelles).

# Chapitre 6

## Problèmes rencontrés et solutions apportées

### 6.1 Restrictions du Lab (droits AWS / SCP)

Lors de la création d'instances EC2, une erreur d'autorisation *explicit deny* peut apparaître (politique du lab). **Solution** : respecter les ressources autorisées par le lab (région, types d'instances, quotas).

### 6.2 Docker : permissions et exécution

Un problème *permission denied* lors de l'utilisation de Docker peut survenir si l'utilisateur n'a pas les droits. **Solution** : utiliser `sudo` ou ajouter l'utilisateur au groupe docker (selon la configuration autorisée).

### 6.3 Compatibilité base de données (MySQL / MariaDB)

Zabbix impose des contraintes strictes sur la version de la base et le charset/collation.

**Solutions appliquées :**

- migration vers MariaDB (version compatible) ;
- configuration `utf8mb4/utf8mb4_bin` ;
- ajout de l'option d'override lorsque nécessaire (`AllowUnsupportedDBVersions`).

### 6.4 Communication Agent & Docker (IP privée vs 127.0.0.1)

Le serveur Zabbix étant dans un conteneur, l'adresse 127.0.0.1 ne correspond pas à l'hôte. **Solution** : utiliser l'IP privée de la machine (ex. 10.0.1.163) dans la configuration des hôtes et des agents, et ajuster les permissions (`Server=`).



## 6.5 Windows : absence de navigateur et installation MSI

Sur la VM Windows, l'absence de navigateur a imposé une installation par commande.

**Solution :** téléchargement via PowerShell (`Invoke-WebRequest`) puis installation silencieuse avec `msiexec`.

# Conclusion

Ce projet a permis de déployer une solution de supervision centralisée complète sur AWS. Le serveur Zabbix déployé sous Docker, associé à MariaDB, supervise avec succès des hôtes Linux et Windows via IP privées. Les preuves (ZBX vert, graphiques, dashboard) démontrent la collecte et la visualisation des métriques en temps réel.

Des difficultés (compatibilité DB, réseau Docker/agent, contraintes du lab) ont été rencontrées puis résolues. La solution obtenue est robuste, extensible et conforme aux bonnes pratiques cloud.

Merci pour l'attention et la lecture de ce rapport

FIN