



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

**Réalisé par :** Abdelhakim TAHA

**Encadré par :** Prof. Azeddine KHIAT

**Année universitaire :** 2025/2026

**Filière:** 2ACI INFO GB

## 1. Introduction

Dans un contexte où les infrastructures informatiques deviennent de plus en plus distribuées et critiques, la supervision constitue un élément essentiel pour garantir la disponibilité, la performance et la sécurité des systèmes. Ce projet vise à mettre en place une infrastructure de supervision centralisée dans le cloud AWS à l'aide de l'outil open source Zabbix.

L'objectif principal est de superviser des machines hétérogènes (Linux et Windows) à partir d'un serveur central déployé sur AWS, tout en assurant la collecte de métriques, la visualisation des données et la génération d'alertes.

## 2. Objectifs du projet

Les objectifs de ce projet sont les suivants :

- Concevoir une infrastructure de supervision centralisée dans le cloud.
- Déployer un serveur Zabbix sur AWS en utilisant Docker.
- Superviser deux clients distincts : un client Linux et un client Windows.
- Collecter des métriques système (CPU, mémoire, processus, disponibilité).
- Mettre en évidence les problèmes et alertes via le tableau de bord Zabbix.

## 3. Architecture de l'infrastructure

L'architecture repose sur des instances EC2 hébergées dans un même VPC AWS afin de garantir une communication sécurisée via des adresses IP privées.

Les composants sont les suivants :

- Serveur Zabbix : instance EC2 Ubuntu exécutant Zabbix via Docker.
- Client Linux : instance EC2 Ubuntu supervisée par Zabbix Agent.
- Client Windows : instance EC2 Windows Server supervisée par Zabbix Agent 2.

## 4. Mise en place de l'infrastructure AWS

### 4.1 Création des instances EC2

Trois instances EC2 ont été créées :

- Une instance dédiée au serveur Zabbix.
- Une instance client Linux.
- Une instance client Windows.

Chaque instance est déployée dans le même VPC et le même sous-réseau afin de faciliter la communication interne.

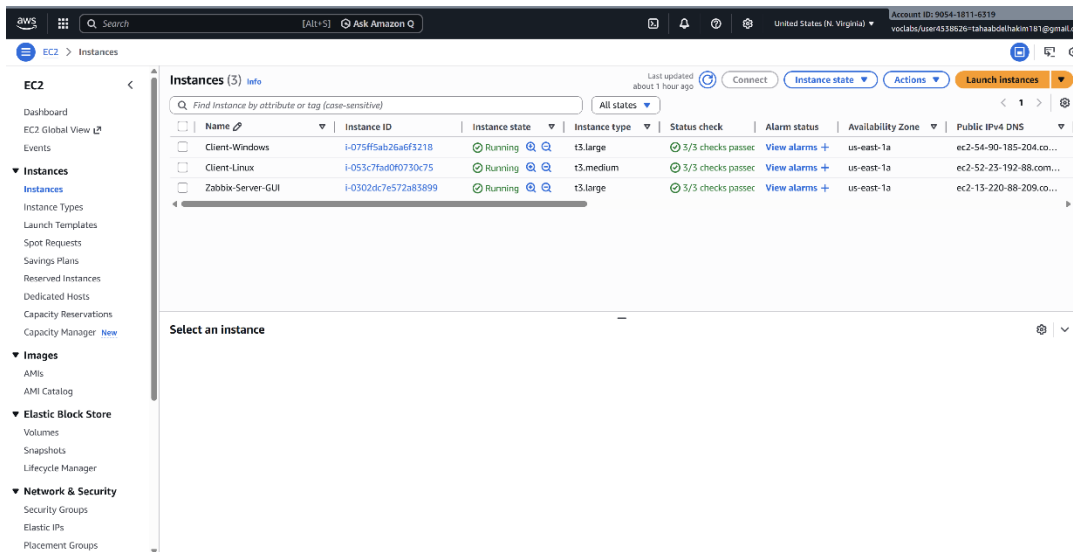


Figure 1: Liste des instances EC2 déployées sur AWS.

## 4.2 Configuration réseau et sécurité

Les règles de sécurité (Security Groups) ont été configurées comme suit :

- Port 22 (SSH) pour l'administration Linux.
- Port 3389 (RDP) pour l'accès à l'instance Windows.
- Port 80 pour l'accès à l'interface web Zabbix.
- Port 10050 pour la communication des agents Zabbix.
- Port 10051 pour le serveur Zabbix.

Les communications entre le serveur et les clients utilisent exclusivement les adresses IP privées AWS.

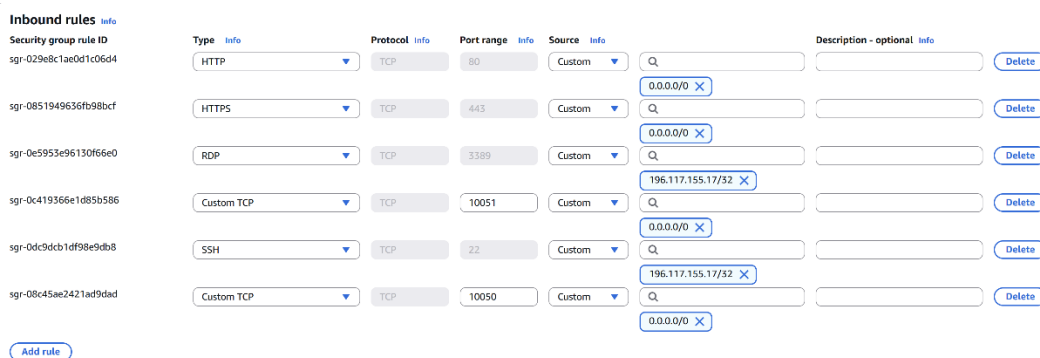


Figure 2 : Configuration des règles entrantes du groupe de sécurité AWS pour l'infrastructure Zabbix

## 5. Déploiement du serveur Zabbix

### 5.1 Installation de Docker

Le serveur Zabbix est déployé à l'aide de Docker afin de simplifier le déploiement et la gestion des services.

```
sudo apt update
```

```
sudo apt install -y docker.io docker-compose
```

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```

## 5.2 Déploiement avec Docker Compose

Un fichier docker-compose.yml a permis de déployer :

- Zabbix Server
- Zabbix Web (Nginx)
- Base de données MySQL

Après le déploiement, les conteneurs sont opérationnels.

L'interface web Zabbix est accessible via un navigateur.

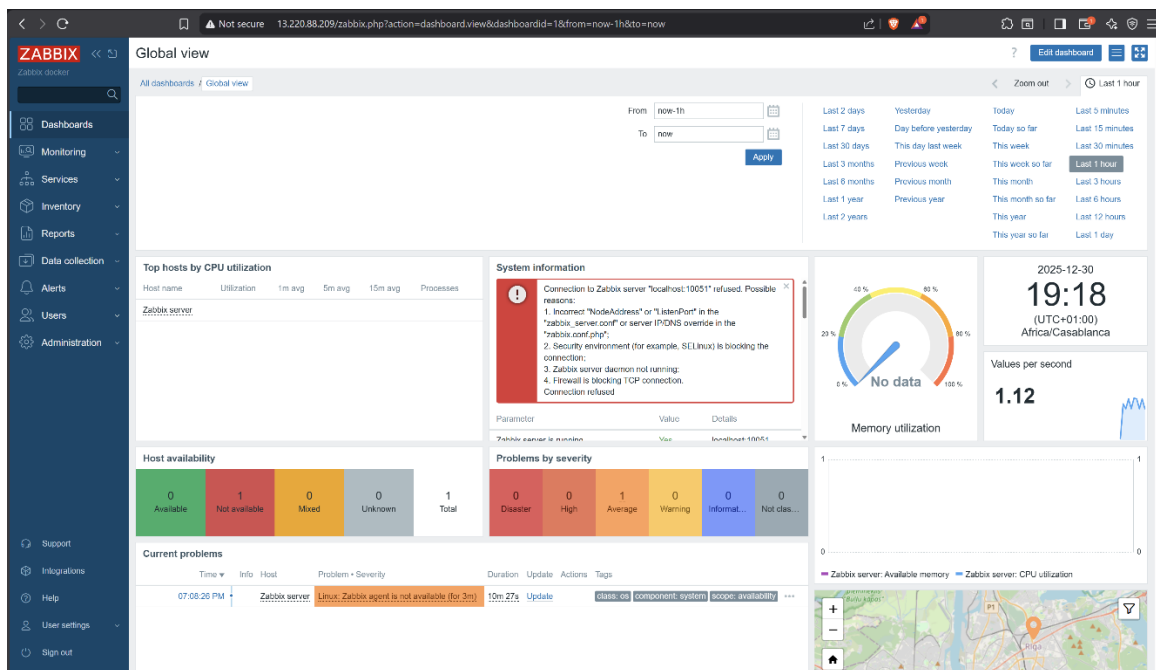


Figure 3 : Interface web Zabbix après déploiement.

## 6. Supervision du client Linux

### 6.1 Installation de l'agent Zabbix

Sur le client Linux, l'agent Zabbix a été installé afin de collecter les métriques système.

```
sudo apt update
```

```
sudo apt install -y zabbix-agent2
```

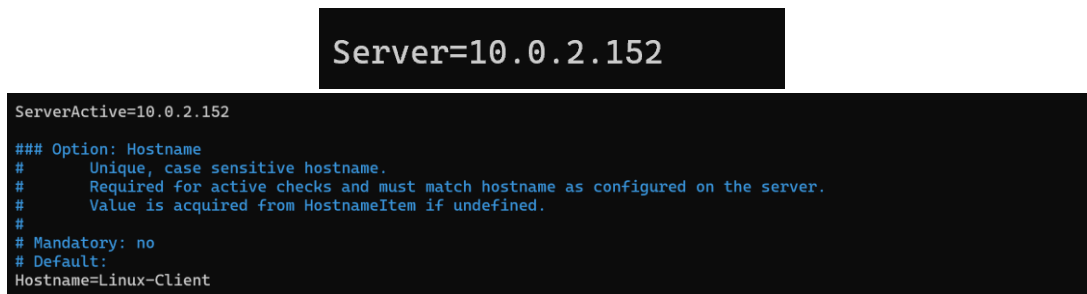
## 6.2 Configuration de l'agent

Le fichier de configuration /etc/zabbix/zabbix\_agent2.conf a été modifié pour définir le serveur Zabbix et le nom de l'hôte.

```
Server=<IP_privée_Zabbix>
```

```
ServerActive=<IP_privée_Zabbix>
```

```
Hostname=Linux-Client
```



```
Server=10.0.2.152

ServerActive=10.0.2.152

### Option: Hostname
# Unique, case sensitive hostname.
# Required for active checks and must match hostname as configured on the server.
# Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
Hostname=Linux-Client
```

Figure 4 : Configuration de l'agent Zabbix sur le client Linux.

## 7. Supervision du client Windows

Sur le client Windows, Zabbix Agent 2 a été installé via le package MSI officiel.

### Configuration de l'agent

```
New-NetFirewallRule `
-DisplayName "Zabbix Agent 2" `
-Direction Inbound `
-Protocol TCP `
-LocalPort 10050 `
-Action Allow
```

## 8. Tableaux de bord et alertes

Zabbix permet de visualiser l'état global de l'infrastructure à travers des tableaux de bord personnalisés.

Le tableau de bord global affiche :

- La disponibilité des hôtes.
- L'utilisation CPU des clients Linux et Windows.
- Les problèmes détectés par sévérité.

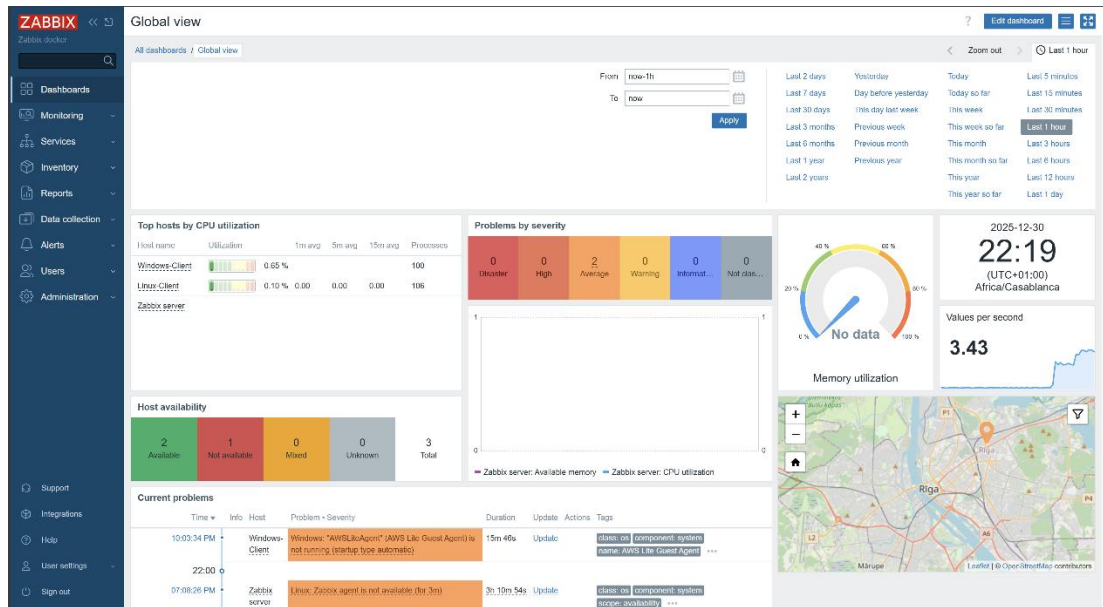


Figure 5 : Tableau de bord global de supervision.

## 9. Résultats obtenus

Les résultats obtenus montrent que :

- Les deux clients sont correctement supervisés.
- Les métriques système sont collectées en temps réel.
- Les alertes sont générées automatiquement en cas de problème.
- Le tableau de bord offre une vue claire et centralisée de l'infrastructure.

Top hosts by CPU utilization					
Host name	Utilization	1m avg	5m avg	15m avg	Processes
Windows-Client		0.65 %			100
Linux-Client		0.10 %	0.00	0.00	106
Zabbix server					

Figure 6 : Données de supervision CPU et mémoire.

## **10. Difficultés rencontrées**

Plusieurs difficultés ont été rencontrées durant la mise en œuvre :

- Configuration des règles de sécurité AWS.
- Problèmes de communication entre agents et serveur.
- Adaptation de Zabbix à un environnement Docker.

Ces problèmes ont été résolus grâce à une configuration correcte des ports, des IP privées et des templates Zabbix.

## **11. Conclusion**

Ce projet a permis de mettre en place une infrastructure de supervision centralisée performante en s'appuyant sur AWS, Docker et Zabbix. La solution déployée permet de superviser efficacement des systèmes Linux et Windows, de visualiser les métriques clés et de détecter rapidement les incidents.

Cette infrastructure constitue une base solide pouvant être étendue à des environnements plus complexes et professionnels.